2019

Three essays on applications of machine learning in problems with high dimensional data

Shaobai Jiang
Iowa State University

Three essays on applications of machine learning in problems with high dimensional data

by

Shaobai Jiang

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Economics

Program of Study Committee:
Brent E Kreider, Co-major Professor
Zhengrui Jiang, Co-major Professor
Otávio Bartalotti
Wallace E Huffman
Cindy L Yu

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation. The Graduate College will ensure this dissertation is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2019

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

ACKNOWLEDGMENTS

I would like to thank my committee co-chair, Dr. Brent Kreider and Dr. Zhengrui Jiang, and my other committee members Dr. Wally Huffman, Dr. Otávio Bartalotti, and Dr. Cindy Yu, for their guidance and support throughout the course of this research.

In addition, I would also like to thank my friends, colleagues, the department faculty and staff for making my time at Iowa State University a wonderful experience. I want to also offer my appreciation to those who were willing to participate in my surveys and observations, without whom, this dissertation would not have been possible.

ABSTRACT

The amount of data businesses collecting from the internet is massive. Researchers and analysts can now track various data features generated from log files, such as customers' behavior history, product descriptions and aggregate level data. etc. In an ideal scenario, such data could be represented in a spreadsheet, with columns representing each dimension. In practice, the number of data dimensions can be staggering, making data processing difficult. With high dimensional data, the number of features can be more than the number of observations, and it can be very challenging for traditional econometric method to handle this scenario. My dissertation addresses this data issue by applying machine learning techniques, including LASSO (least absolute shrinkage and selection operator), decision trees, and neural networks, to help decision makers perform descriptive-predictive, and prescriptive analytics based on high dimensional data.

My dissertation comprises three essays. The first essay applies tree based machine learning models (random forest and gradient boosting decision tree) and free text information to predict house prices and understand how certain factors could affect the prices. In the second essay, I propose a LASSO method in high dimensional data and use daily prices of hotels to understand hotel's competition pattern in a certain area. In the third essay, a word embedding and neural network model is applied to real estate data to more efficiently extract free text information, which leads to more accurate of house prices.

In these essays, I apply and extend a variety of analytic tools including supervised learning, unsupervised learning, statistics, and econometric methods. These essays contribute to the applied econometric and business analytics literature and can help researchers and analysts

appreciate both traditional econometrics and predictive analytics tools, and make data-driven business decisions.

CHAPTER 1. GENERAL INTRODUCTION

High dimensional data source is becoming a new popular topic for researchers to solve. Different to the conventional data source, the number of features in high dimensional space can be more than the number of observations, and the coefficients are un-identified in this case. My dissertation addresses this data issue by applying a variety of machine learning techniques to handle model selection and variables selection in high dimensional data problem.

The first essay focuses on real estate pricing analysis and prediction. Traditional real estate researches only use no more than 20 different features for variable explanation and price prediction. Very few papers include text information of the house in their house modeling since it is difficult to handle high dimensional and unstructured words vectors. In the first essay, I will introduce bag-of-words to convert words into word vectors that can be used in econometric models. Then I propose a decision tree based machine learning model to analyze factors that impact price and make price prediction including free text information. The result analyzes and ranks the importance of factors that highly related to house prices. It also shows the decision tree based model can automatically detect non-linear relationship in data which leads to better prediction in prices.

The second essay attempts to illustrate the competitive pattern in New York hotel market. It is interesting to find how decision makers in the market identify its competitors and make according revenue strategy with competitors. In this paper, I use online customer's search data to construct the problem, and reveal how frequently each hotel has been exposed and requested on daily basis. Price competition in the hotel industry is simultaneous system equations since it is a joint determination of all the market competitors. The challenge is to estimate the coefficients in the systems of equations in a higher dimensional space since there are hundreds of hotels in each market. I propose a data-driven linear regression approach like adaptive LASSO and group LASSO to identify the competition pattern. I find hotel revenue managers weight hotels with the same star rating more than hotels in the same region as their major competitors. The LASSO algorithm can be also applied into some other high dimensional business problem to identify competition pattern, and provide a recommendation to related product/service producers to make decisions.

The third essay discusses how to efficiently utilize free text information and predict house prices with this information. The most popular and efficient approach to understand the unstructured text in natural language processing area is Word2Vec. I propose to apply neural network structure to predict the house prices with pre-trained text information using Word2Vec approach. I compare six groups of results with different hyperparameter and show the proposed model outperforms the traditional linear model and machine learning model in out-of-sample prediction. This paper clearly shows that neural network model has a significant advantage in utilizing text information to predict the house price and shed light on other economic research paper to apply neural network technique on price prediction topic.

CHAPTER 2. ANALYZING REAL ESTATE USING MACHINE LEARNING AND TEXT MINING

2.1 Introduction

Since 2004, American traditional real estate industry has undergone a fundamental shift, from local agency brokerage dominant business to online search, mobile internet database and user generated content business. With web-based real estate platform companies like Zillow.com, Redfin.com, Trulia.com founded, consumer can acquire and compare real estate property information with significantly less time. With search engine and web platform upgraded, customers are capable of searching house listings, acquiring market information and contacting real estate agents all by online platform website. Moreover, consumers are not only the information receivers anymore, but also generate information or online reviews as well. This shift has reduced information asymmetry issue in real estate industry between buyers and sellers, and makes house transaction opaquer to both sides of business participants. From statistics, 67% of house buyers would view the house webpage first before they make the actual house visit. From transactions, 51% of all buyers choose to purchase the house they have been searched and clicked on the internet before, compared to 34% of buyers asked traditional real estate agency channel to complete the transaction.

The very first group to build and benefit from this shift is online real estate platform companies, like Zillow.com and Trulia.com, etc. Mainly, they provide consumers with complete information chains and comfortable search experiences. Among all these online real estate platforms or online agencies, Zillow is currently the most prominent information website in the U.S. Up to March 2016, Zillow already has 166 Million Monthly Active Users (MAU), with 115 million homes information covered in their website.

With these large amounts of customers and properties information, valuation models for real estate are crucial to understand the industry and plan the corresponding strategy to make a profit. Hedonic model is widely used to estimate implicit prices for attributes of heterogeneous goods. Hedonic model dissects the price of homes into different attributes with linear relationship and values each one of them. The model has been widely discussed since it can analyze price and demand for not only single source like appliances and amenities but also public goods or aesthetic values. One of the primary concern in this model is omitted variable bias. Linear hedonic model is prone to suffer variables unobserved to researchers, and these unobserved factors are correlated to included, observable attributes. Econometricians often rely on econometric method such as instrument variables, fixed effect to obtain inference of the target variable. These methods usually require a strong underlying assumption that is difficult to justify.

However, increasing richness of data from different public resources, omitted bias concern can be alleviated since more previous unobserved noise data can be observed or controlled. In this paper, we propose to apply text mining technique from online resource Zillow.com, to increase availability and dimensions of data from the website. This new approach can increase the explanatory power of the dependent variable, also reduce the coefficient bias brought by previous data limitation. We add these texts based approach to the traditional hedonic model framework to assess house attributes value and interpret some interesting findings.

Including a new source of data from text information also transforms the problem from the low dimensions to high dimensions. Moreover, this transformation makes the hedonic price framework closer to real life home buying decision. Buying a home is undoubted a high dimensional problem with many variables included in the buyer decision process: quality of house structure, appliances house included, cooling and heat system, neighborhood demographics, local schools quality, outside view or aesthetic, etc. However, traditional hedonic literature can only model the real estate price in low dimensional way with limited variables. From hedonic related literature's statistic summary (Xiao 2017), most previous economic literature use around 10- 20 attributes to construct their model, which simplify and underestimate the complexity of house purchase decisions. In this paper, we use a much richer attribute set with around 1000 variables that extracted from Zillow.com, including house listed information and user generated content. Online information helps us build a better explanatory and predictive model to describe home buyers' attribute preference and decision. This type of data has rarely been used in previous research owing to lack of a method to quantify and include in the linear regression model. However, it is relevant to the purchase price and should not be unobserved noise like previous work.

To show how these new sources of text data included would make different valuation result compared to the original hedonic model, we will use two listed house in Zillow dataset to illustrate. House A and B sit in the same zip code area in New Jersey, both have four bedrooms and two bathrooms. Both houses have around 1370 square footage and built in the age between 1975 and 1979. Two houses also share the same school districts, same neighborhood demographic statistics. Most previous hedonic pricing literature, which only uses these house basic attributes and demographic information, would conclude house A and B have similar prices. However, the reality is much different. House A sold in July 2016 at a price 450,000. House B sold in May 2016 at a price 665,000. A model only includes listed house observable attributes cannot explain or predict nearly 50% sold price bias in house A and B. But taking an in-depth exploration on text information of house listed attributes and owner self-descriptions can explain a lot of the difference. Detail are listed in Table 2.1, Table 2.2, and Figure 2.1. Table 2.1 contains text form house A and house B listed facilities, Table 2.2 is house owner/ agent generated house description for respective houses. Figure 2.1 is a photo of their house.[1]

Table 2.1. House Listed Facts

<table>
<tr><th>Price</th><th>Listed house facts</th></tr>
<tr><td>House A
Sold $450,000</td><td>Lot: 4,500 sqft| Single Family| Built in 1975| All time views: 2,386| Last sold: Jun 2016 for $450,000| Last sale price/sqft: $329| Exterior material: Wood| Stories: 2| Unit count: 1| Floor size: 1,368 sqft| Lot depth: 75 ft| Lot width: 60 ft|</td></tr>
<tr><td>House B
Sold $665,000</td><td>Lot: 4,030 sqft| Single Family| Built in 1979| All time views: 1,394| Cooling: Wall| Heating: Baseboard| Last sold: May 2016 for $665,000| Last sale price/sqft: $478| Ceiling Fan/Deck| Flooring: Carpet, Hardwood, Laminate| Parking: Off street| Patio| Skylight| Vaulted Ceiling| View: Water| Exterior Features: Deck| Doors - Sliding Glass| Outdoor Shower| Windows - Double Hung| Windows - Screen(s) --- Interior Features: Balcony| Bedroom on 1st Floor| Breakfast Bar| Open Floor Plan| Window Treatments |Dishwasher| Dryer| Microwave| Range / Oven| Refrigerator| Washer| Breakfast nook| Dining room| Master bath| Exterior material: Vinyl| Roof type: Other| Stories: 2| Structure type: Cape cod| Unit count: 1| Floor size: 1,392 sqft| Heating: Electric| Laundry: In Unit| Lot depth: 62 ft| Lot width: 65 ft|</td></tr>
</table>

[1] Data, text and photos all come from Zillow.com.

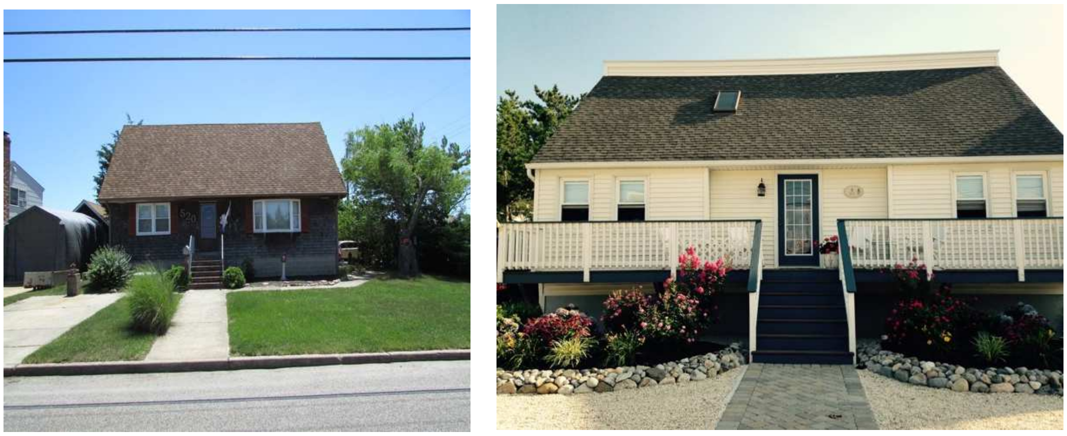


Figure 2.1. House A (left) and House B (right) Comparison.

Table 2.2. House Listed Description

	House description
House A sold for $450,000	Very well maintained cape close to causeway for quick on and off the island. This home was built as a personal home of a builder with over sized beams and insulation. The home has newer vinyl windows. There is a nice porch on the back of the home to allow you to feel like you are outside but yet stay dry and bug free. This lot is treed to give you some great places to sit out of the sun as well as fenced back yard that has plenty of places to sit in the sun.
House B sold for $665,000	Contemporary cape located 5th from the beach in the very desirable Southport area of Holgate. This home features 4 bedrooms, 2 full baths, master bedroom with a private deck. Open floor plan with cathedral ceiling in the living room, kitchen with breakfast bar and dining area. The lower deck off the kitchen is wonderful for entertaining and outdoor dining. Recent upgrades include second floor carpet, first floor wood flooring and ductless air conditioning. Professionally landscaped garden area

Compared to house A, house B has much more detail, including facilities and amenities that lots of home buyers are very interested. The facts and descriptions in house B give house viewers impression that although house B is smaller than house A, it is very likely more refined decorated. Moreover, house B mentioned that house has an ocean view and private spot to enjoy the scene. In fact, this aesthetic value also affects house prices significantly. Our model shows that houses with water view would raise $44,913 of the price compared to a similar house without a view. Without transforming this house attributes lists and unstructured text information into observable and quantifiable attributes, the traditional hedonic model cannot explain the price difference between two houses, or accurately estimate the outlook value for house B.

These contexts are categorized as text mining data in information retrieval and machine learning area. One approach can be applied in here is to parse the words, stem the word origins, and transform unstructured text as a matrix of word frequencies. Finally, extract informative words matrix and build a high dimensional data frame for further research. These series of approach involved natural language processing area techniques like bag-of-words, stemming, and term frequency inversed document frequency (TFIDF). We will give more detail about this approach in section 3. A word frequency matrix with thousands of columns increases the data availability, but also brings concerns to choose predictive variables from the big set correctly. Majority of words used in the text are not predictive for house pricing, and the traditional linear regression model will suffer from high dimensional variable selection problem. This issue can be handled with modern machine learning method which automatic execute model selection and rank the variable importance with a matrix.

Machine learning models not only can deal with high dimensional variables selection, but perform well in model prediction. From all the machine learning models, regression tree models are the most common choice for statisticians and computer scientists. Tree models rank similarities in the observations and group similar observations together to minimize the model prediction error. Recently, econometricians start to apply tree models in causal inference and covariate impact explanation. Based upon their previous findings, we will show that machine learning models have better predictive performance in house prices, and they also yield better performance to evaluate attributes of the house using the similar approach.

This paper has three main contributions. First, we show how unstructured data in the form of free text can be transformed into a structured data frame and included in the traditional regression model. We illustrate clear steps on how to use information retrieval tools to get quantifiable attributes which can be applied in other forms of text. This added source of text information increases model explanatory power adjusted R-square from 48% to 70%. These methods are originally developed and widely used in the computer science area; however, few economic research papers have considered using these methods in economics area problem, especially rarely used in real estate economics.

Second, we show how the regression tree learning model can handle and recover the non-linear explanatory variables impact to response variable where linear regression model is challenging to do. In previous literature, it is rare to see economists use non-linear form variables or higher order variables to analyze the real estate prices. Even in a few exceptions, researchers did include the non-linear form variables but variables selection was based on their in-depth domain knowledge or adopting related professional advisory. Moreover, this ad-hoc way to build models only works for a few well-known variables and low-dimensional data. It is nearly impossible for domain experts to understand all non-linear form impact to real estate price for each variable in high dimension data. It is also not plausible to try every single possible choice for this issue. We will show machine learning models have a better algorithm to recover non-linear impact in this dataset. Some covariates coefficients in the hedonic linear model are not significant, but it has significant non-linear correlations on real estate price, which can be obtained by tree based model.

Third, we show two tree based models: random forest and gradient boosting regression tree model both have better performance in sample fit and out of sample fit (measured by mean squared error) compared to the hedonic linear model and LASSO regression model. In high dimensional space with non-linear impact from covariates, Varian (2014) suggested that model with highly predictive ability can be used to analyze covariates treatment effect. Tree based model, especially gradient boosting tree model with lower out of sample mean squared error, is a good predictor for house prices. Moreover, the traditional hedonic linear model suffers from the missing correlation between omitted variables and included regressors. The degree of bias is proportional to the degree of correlation from omitted variables. With including more available data from Zillow.com website, we sample 10,223 real estate house data from all over the US area. This paper shows that houses listed information with an outside view has sold significant higher prices than houses without views. Also, different outlook views like mountain, city skyline, water, also have different values boosting in the model.

These findings just shed light on how this online text information can be applied in analyzing home buyers behaviors and can help real estate agents and online real estate platforms better utilize this information to compute suggested price, reduce the information asymmetric and ultimately improve transaction efficiency for real estate market. From the researcher perspective, this paper is a new effort to apply methods and tools from other areas like natural language processing, computer science, statistics to analyze the hedonic house pricing models.

We hope this study can draw further researches on this topic and methods, which brings more broader applications in different economic and business topics.

The rest of paper is organized as follows. In section 2.2, we review the related literature. In section 2.3, we introduce the methodology used in this paper. In section 2.4, we summarize the Zillow.com house data. Section 2.5 shows the prediction result and the approximate estimate of attributes value. Section 2.6 concludes the paper with discussions on contributions and further area can be applied.

2.2 Related Literature

Our literature review discloses previous formal research on the real estate industry, text mining methods, high dimensional model selection, and tree based machine learning model study. The first area of study is from real estate literature. Rosen (1974) is first to study real estate price with hedonic model, and he indicated that total price of a house can be considered as the sum of prices of each homogeneous attribute, and each attribute has a specific implicit price. After that, many research literatures discuss various possible factors that influence the housing market. Most popular attributes researchers interested in are basic house structure like square footage and house ages. Sirmans et al. (2005) summarized top 20 attributes that have been used to specify hedonic pricing models. He listed the total number of times each attribute has been used and statistically significant in previous papers. Age, square footage, bathrooms, bedrooms, garage are listed the most frequently used attributes. However, there is no concluded answer for any of the above attributes is consistently significant (positive or negative) from these papers. Kain and Quigley (1970) investigated house condition attributes like exterior structure, floor type and windows that have a significant effect on the price of housing. The second category of attributes prevalent in housing study is neighborhood characteristics. Hughes and McCormick

(1994) identified the neighborhood region's unemployment and incomes have significant impacts on house price. Dubin and Goodman (1982) estimated the impact of school quality on 1,765 house prices in Baltimore in 1978. They concluded school quality had a significant effect on house price, but they excluded school distance factor in their study. Third part attributes bring many discussions are environment and aesthetic attribute. McLeod (1984) discovered that river views were particularly essential and had a greater impact than park views on house prices. Weigher and Zerbst (1973) found five parks in Columbus, Ohio brought down nearby house prices for $1150 compared to house sat in one block away.

Though still not widespread, the second area is papers using natural language tools in economics. As mentioned before, the unstructured text contains some crucial features that traditional numerical attributes cannot adequately represent. Gentzkow et al. (2015) used machine learning tools to analyze political affiliations from speeches. Kang et al. (2013) used Yelp review to estimate restaurant hygiene quality. The most related to our work is Nowak and Smith (2017) used MLS listing houses' text remarks to predict house prices, their paper concluded similar result that text information from MLS comments reduced house price errors by over 20%. However, they still used linear regression models and did not analyze any specific attribute added value to house prices. Our paper shows that the regression tree model can reduce house price errors even lower than the linear model.

The third area of paper is linear regression models in high dimensional data. With data dimension increases at an exponential level in the modern era, many researchers choose the regularization method to select their model variables. The most widely used is Lasso (least absolute shrinkage and selection operator) which induces to shrink some model coefficients to zero (Tibshirani (1996)). Knight and Fu (2000) studied asymptotic results for LASSO coefficients. Belloni et al. (2014) proposed to use post-double-selection Lasso to infer treatment effect in high dimension data and show an empirical example of estimating the effect of abortion on the crime rate. In this paper, we will use LASSO regression model as model performance comparison with regression tree based models in predictive ability.

The fourth area of paper is tree based model applied in research. Breiman et al. (1984) first stated CART (classification and regression tree) model with a recursive partition of attributes in data. This model is very prevalent in area like statistics and computer science, but suffers over-fit data problem. Breiman (2001) introduced another version of tree model-- random forest method which includes bootstrap and reduces overfitting problem in the algorithm. Freidman (2001) investigated a new method called gradient boosting decision tree which combines the ideas of both CART and boosting. Both algorithms are currently among the most popular tools in model prediction in many academic research areas. Bajari et al.(2015) is the first economist that applied random forest model in estimate product demand in IRI Marketing Research dataset. In recent years, economists start to investigate on regression tree model on heterogeneous treatment effect. Wager and Athey (2017), Athey et al. (2016) respectively applied random forests and gradient forests to get inference on heterogeneous treatment effect.

Our research on online real estate platform was rarely researched in the previous study. Lee and Sasaki (2014) studied the sensitivity of house price on Zillow's proprietary algorithm suggested price. Hu et al. (2017) investigate Zillow house reviewed times and house saved by home buyers ratio has a positive impact on house sold price. However, both of papers are more focusing on comparison with Zillow's proprietary house price algorithm—Zestimate, which is black box algorithm and is hard to define its correlation and impact on other independent variable and house prices. In this paper, we will only use attributes that can be observed in Zillow.com website and do not use Zestimate as an index, since its algorithm is unclear to compare.

2.3 Methodology

In this section, we will introduce the methods that have been used for this paper, including high-dimensional linear regression model (Post-Lasso), regression trees and advanced machine learning models: random forest and gradient boosting tree. Text mining technique bag of words and term frequency inversed document frequency are also listed in this section. Currently, there are other models also have a solid performance on prediction, like support vector machine (SVM) or neural network model, but these models lack explanations of how variables are selected and used in the model. All models that have used in this paper have a metric to quantify or illustrate how variables involve in these models.

2.3.1 Hedonic Model

Before introducing other models, we will start with the traditional baseline model: hedonic pricing model. Hedonic price model (Rosen (1974)) interprets house prices can be regarded as the sum of different house attributes implicit price. The gradient of hedonic price function is the implicit price for each attribute included in the model. In most literature, the semilog form is favorite to use: the dependent variable is log form, and the explanatory variable is linear like below:

$$\ln p_{ikt} = x_{ikt}\beta + z_{kt}\gamma + \epsilon_{ik}$$

p_{ikt} is the price of house i in neighborhood k at times t. x_{ikt} represents house i in neighborhood k at time t. z_{kt} is the neighbor area demographic vector, ϵ_{ikt} is an idiosyncratic error. In this framework, β measures buyers most willing to pay for an incremental change in this equilibrium. For unbiased and consistent estimator, assumption $\mathrm{E}[\epsilon_{ik}|x_{ikt}, z_{kt}] = 0$ is needed. However, for most research paper, there are only limited variables observed. Omitted variables, which correlate with observable variables, would lead to $\mathrm{E}[\epsilon_{ik}|x_{ikt}, z_{kt}] \neq 0$ and estimator biased and inconsistent. With increasing data availability, concerns of estimator bias and inconsistency are reduced since more previous omitted variables can be observed.

2.3.2 High Dimensions Linear Regression Method

In high dimensional data, previous research literatures are difficult to compare directly. There are many variables in our dataset that rarely been used in past literature, and these variables' importance and correlation are both unknown. Furthermore, including words in context as variables is likely to face observations N is near or smaller than the number of variables K. For the case like $\mathrm{N} \leq \mathrm{K}$, linear equations cannot be identified and the whole model fails to estimate any covariates. Of course, use domain knowledge to drop variables that unlikely to have an impact is also a feasible choice. However, dropping possible irrelevant variables is subjective and difficult to justify. Therefore, it desires to have an automated algorithm to finish the task.

In this case, the variable selection algorithm in LASSO is possible to finish this task. Especially in the sparse model with many zeros in data point, the computing speed for LASSO is also quicker than the typical linear model. Moreover, LASSO algorithm can also prevent high dimensional multicollinearity problem and potential overfitting problem with cross-validation.

2.3.2.1 Post LASSO

Tibshirani (1996) is first to claim to add regularization term in executing the model selection. Lasso algorithm regularizes linear model with high dimensional variables by shrinking part of model coefficients to zero. Lasso uses l-1 norm as the penalty function in the least square optimization formula, form as[2]:

$$\min_{\beta} \frac{1}{N} \sum_i (y_i - x_i\beta)^2 \; + \; \lambda||\beta\,||_1 \qquad \text{(2-1)}$$

Equation (2-1) is the sum of squared errors plus penalty term of all coefficients excluding intercept. The penalty term is proportional to the sum of absolute values of coefficients in the first term least square problem. Parameter λ is a tuning parameter or called hyperparameter, which controls the weight of the penalty term and affects equation (2-1) result. If $\lambda = 0$, equation (2-1) equals to ordinary OLS equation. For case $\lambda > 0$, the estimator is an optimization problem with Lagrangian form. The problem can be solved with Kuhn-Tucker condition and has unique solution when K < N. When $\lambda > 0$, coefficients are biased toward zero. λ is given and not an optimized parameter in the equation, but the cross-validation method can be used to find optimal λ that performs best. Cross-validation divides data into m partitions, where the model is trained on m -1 subsamples and the left untouched subsample is used as test data to determine model's out-of-sample performance. This process repeats m times and average of m times out-of-sample mean squared error is regarded as the model's out-of-sample performance. The model hyperparameter λ value is determined by λ_{CV} with the best out-of-sample performance. LASSO model reduces the prediction variance and simplifies the model with parsimonious form. It shrinks variables with some coefficient to zero. Also, parsimonious form suffers less from multicollinearity, and model prevents overfitting data using cross-validation.

[2] Lasso form use glmnet (Friedman (2017)) package formula.

Since original tuning parameter λ is still prone to include too many variables and coefficients are biased toward zero, another version of LASSO has been studied to prevent this problem. Belloni and Chernozhukov (2013) claim to run OLS using only regressors with nonzero coefficients from LASSO estimate to resolve the issue. More specific, the first stage is to estimate equation (2-1) use LASSO model and select a list of variables that coefficient is not zero. In the second stage, run OLS regression only on the list of variables in the first step and get unbiased coefficient. Another approach is Zou (2006) claims to use two-step adaptive LASSO where adaptive weights are used for the l-1 penalty, which gives different penalty weight to parameters. The choice of method depends on the specific problem. In this real estate pricing case, the house purchase decision is affected by many factors, and this paper focuses on variables that have not been observed and founded in previous literature. We choose to use post-LASSO method proposed by Belloni and Chernozhukov (2013) in this paper.

Both of the above alternative versions of λ are still data-driven hyper-parameter. Large λ will reduce variance but increase bias. Small λ is vice versa. The selected λ makes the balance between bias and variance. Post LASSO method takes advantage of variable selection from LASSO and still can have OLS unbiased coefficient standard error.

2.3.3 Regression Tree

Regression tree based model is another prevalent approach in statistics and machine learning. The significant difference to LASSO method is tree based model uses the nonparametric idea to split the input data into small regions. Recursive partitions lead to nonlinear splits, which are a better fit for variables have nonlinearity and term interaction issue. At each step, the decision is made to split the inner space X, which leads to smaller RSS (residual sum of squares). Through recursive partitions, similar observations are grouped and researcher can use the conditional expectation of the group to represent group feature.

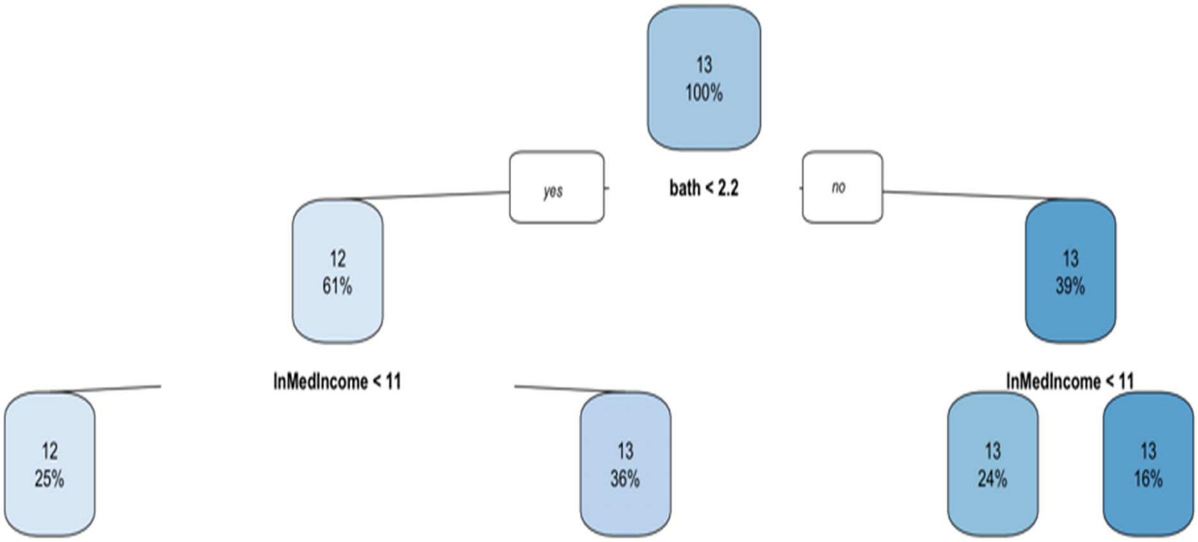


Figure 2.2. Tree model to predict house prices with log form.

Figure 2.2 has hierarchical split which separates data space into 2 in each node. “lnMedIncome” represents log form’s house median income.

We will use Figure 2.2 tree visualization as an example to explain how the tree model works. In here, the model uses a log of house price, as the dependent variable. Other explanatory variables in the tree model are decision splits that predict house price. This tree has a depth of 2 and leaf nodes of 4. Each node represents one group of observations condition on specific criteria. Boxplot of each node has two number, the first line is predicted value, in here indicates predicted log price for this group observations. The second line is the proportion of total observations in this node. Specifically, for trees in Figure 2.2, model first splits observations into two groups by bathroom number: smaller than 2.2 goes to the left group, and larger than 2.2 goes to the right group. Left group node has proportions 61% of total observations, and predicted house log price in this group is 12. Right group node has 39% of total observations and predicted house log price in this group is 13. On condition of the first split, tree model splits the data into small groups by another attribute, log form of household median income. The numbers in the plot can be interpreted in a similar way as above. Only show top 2 levels of the tree are shown as a simple illustration to explain how tree model works. With recursive partitions, observations space is sliced into small subspace and the split becomes nonlinear.

For the first split variable x, we select threshold point $x^{(1)}$ (threshold can be more than one) and approximation prediction is estimated by conditional expectation, with the form:

$$\widehat{c_{\iota}(\mathrm{x})} = \begin{cases} c_1^{(1)}, & \text{for } x_i \; < \; x^{(1)} \\ c_2^{(2)}, & for\ x_i \; \geq x^{(1)} \end{cases}$$

For minimized least square problem, loss function can be written as

$$L^{(1)}\left(x^{(1)}, c^{(1)}\right) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{c_{\iota}(\mathrm{x})}\right)^2$$

The loss function is minimized with respect to $x^{(1)}$, suppose $\hat{\mathrm{x}}^{(1)}$ is the minimizer of loss function,

$$c_1^{(1)} = \frac{\sum_{i=1}^{N} y_i 1\{x_i \leq \hat{\mathrm{x}}^{(1)}\}}{\sum_{i=1}^{N} 1\{x_i \leq \hat{\mathrm{x}}^{(1)}\}}$$

$$c_2^{(1)} = \frac{\sum_{i=1}^{N} y_i 1\{x_i \geq \hat{\mathrm{x}}^{(1)}\}}{\sum_{i=1}^{N} 1\{x_i \geq \hat{\mathrm{x}}^{(1)}\}}$$

Then the loss function can be written as form:

$$L^{(1)}\left(x^{(1)}, c^{(1)}\right) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{c_1^{(1)}})^2 1\left\{x_i \leq \hat{x}_1^{(1)}\right\} + \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{c_2^{(1)}})^2 1\{x_i \geq \hat{x}_1^{(1)}\}$$

The above equations are loss function expressions after deciding the first split. All splits are sequentially chosen on each separated subspace. In each split, the algorithm will search all possible variables and thresholds that minimize RSS when create two new regions. The node in the bottom of the tree's branch is called the leaf node of the tree. The number of layers from top to the leaf node is called depth of the tree. In practice, depth of the tree and the minimum number of observations in leaf node are two important hyperparameters that prevent tree model from overfitting. In general, the deeper trees and fewer observations left in leaf nodes have lower bias and better performance in training data, but also sometimes suffer overfitting and bad performance in the testing set.

The sequential partition algorithm in the regression tree model is a good choice to handle high dimensional variable selection issue, and understand better the nonlinear interactions across variables than the linear model. Regression tree is also immune to multicollinearity concern since splits are sequentially chosen with variables lower the RSS most picked first. If two variables are highly correlated, only one of the variables will be chosen since including both will not help lower the RSS.

2.3.3.1 Random Forest

In practice, the regression tree algorithm has a drawback that a single decision tree is prone to overfit the training data and perform poorly in the test set. Random forest applies bootstrap aggregation (bagging) to prevent overfitting problem. The algorithm is straightforward:

For b = 1,2,3,4, ..B

1. Sample with replacement, n training example from X, y; denote as X_b, y_b as the subsample training set

2. Sample without replacement, choose m variables from K, denote as V_b as subsample variable set

3. Run regression tree on training set X_b, $y_b$with variable set V_b, get tree function $f_b(x)$

4. After training, predictions for test sample X_{test} with average weighting B trees function $f_1(x), \dots f_b(x)$, get:

$$\hat{f} = \frac{1}{B}(\sum_{b=1}^{B} f_b(x)),$$

Random Forest is not prone to overfit since each sub model $f_b(x)$ only uses a portion of data and variables, each of the individual trees in random forest should do reasonably well at predicting the target values in the training set but should also be constructed differently in some way from the other trees in the forest. The concluded ensemble model $\hat{f}$ averages different trees and will have better performance than one single regression tree. A clearer explanation about bagging, ensemble, and boosting will be given in Appendix A.1 and A.2.

2.3.3.2 Gradient Boosting

Friedman (2001) proposes gradient boosting regression tree is a generalization of boosting model using gradient descent idea that origins from optimization literature. Like random forest, gradient boosting is an ensemble method that combines many different tree results, and it also has smaller MSE compared to the single regression tree model. The major difference between gradient boosting and random forest is that gradient boosting gives each subtree different weights based on its impact on reducing RSS error. On the other hand, random forest imposes equal weight to each subtree. Gradient boosting tree adds new trees based on prediction result from the previous built tree, which make it a sequential tree model, not like random forest which is parallel tree model and can concurrently build all subtrees.

Gradient boosting model can be considered as an additive training model. Start from constant predictions, and each round adds a new function tree which optimizes loss function:

$$\min_{f_k} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 + \sum_{k} \Omega(f_k)$$

At each subsequent step, gradient boosting chooses subtree that along $\hat{y}_i$ gradient descent direction, which means reduce RSS error most effectively. Start from constant predictions, add a new function each time for t iterations, the expression will be:

$$\hat{\mathrm{y}}_{\mathrm{i}}^{0} = 0$$

$$\hat{\mathrm{y}}_{\mathrm{i}}^{1} = f_1(x_i) = \hat{\mathrm{y}}_{\mathrm{i}}^{0} + f_1(x_i)$$

$$\hat{\mathrm{y}}_{\mathrm{i}}^{2} = f_1(x_i) + f_2(x_i) = \hat{\mathrm{y}}_{\mathrm{i}}^{1} + f_2(x_i)$$

$$\dots$$

$$\hat{\mathrm{y}}_{\mathrm{i}}^{t} = \sum_{k=1}^{t} f_k(x_i) = \hat{\mathrm{y}}_{\mathrm{i}}^{t-1} + f_t(x_i)$$

To prevent gradient boosting overfitting data, in practice, the model adds a step length tuning parameter γ to control fitness speed to real data. Thus, the gradient boosting final form is like:

$$\hat{y}_i^t = \sum_{k=1}^{t} \gamma f_k(x_i)$$

More detail on how gradient direction is computed and optimized will be shown in Appendix A.2. After several steps of math deduction, minimizing below function will be:

$$\hat{y}_i^t = \hat{y}_i^{t-1} + arg\min_{f_t,\gamma} \sum_{i=1}^{t} \left(y_i - \widehat{(y}_i^{t-1} + \gamma f_t)\right)^2,$$

In gradient boosting, iterative adding subtree model is to fit the residual term between $y_i - \hat{y}_i$ and model selects variables that reduce error the most, which make gradient boosting a highly effective algorithm. Not like random forest, this method suffers from overfitting when iteratively fit the residual term, which needs γ to control its learning speed and prevent from overfitting.

Both random forests and gradient boosting methods are nonlinear functions and have advantages in learning higher orders and interactions across variables. Tree based model is invariant to variable scaling, which makes these models also not suffer from unstandardized coefficient scale problem. Moreover, gradient boost can handle missing values well since algorithm splits to subspace only on full information observations. For missing value, gradient boost uses tree based surrogate variable to split. Surrogate variable is also good for model fit in high dimensional space and no need for arbitrary imputation missing data or discard missing data.

One of distinct between regression tree and the linear regression model is regression tree does not have coefficients can directly explain the impact. Thus, analyzing the impact of the explanatory variable to the dependent variable can be challenging work. Partial dependence plot is a good alternative to visualize the relationship between two variables. It describes how each explanatory variable affects model predictions. Assuming the targeted explanatory variable is $\mathrm{x_s}$, and other explanatory variables are as complementary set $\mathrm{x_{-s}}$. Then partial dependence of response on $\mathrm{x_s}$ is as

$$\mathrm{f_s}(x_s) = \mathrm{E}\left[\hat{\mathrm{f}}(\mathrm{x_s}, \mathrm{x_{-s}})\right] = \int \hat{\mathrm{f}}(\mathrm{x_s}, \mathrm{x_{-s}})\mathrm{p}(\mathrm{x_{-s}})\mathrm{dx_{-s}}$$

$\mathrm{p}(\mathrm{x_{-s}})$ is marginal probability density function of $\mathrm{x_{-s}}$, then function $\mathrm{f_s}(x_s)$ can be estimated from Monto Carlo simulation:

$$\bar{\mathrm{f}}_{\mathrm{s}}(x_s) = \frac{1}{\mathrm{n}}\sum_{i=1}^{n} \hat{\mathrm{f}}(\mathrm{x_s}, \mathrm{x_{i,-s}})$$

and $\mathrm{x_{i,-s}}$ (i = 1,2,..n) are all observations that in training set, thus $\bar{\mathrm{f}}_{\mathrm{s}}(x_s)$ partials out all other variables effect on response. The plot describes predicted value $\bar{\mathrm{f}}_{\mathrm{s}}(x_s)$ and x_s relationship. In section 6, I will use partial dependent plot to illustrate how explanatory variables impact house price.

2.3.4 Unstructured Text Processing

The standard approach processing text is to parse words into a big counted set, then use a binary vector to projects words into high dimensional space. This approach is called bag-of-words (BOW). Each distinct word is an integer variable that its value represents the frequency of the words. BOW method effectively grasps single word meaning but is suffering to catch semantic meaning between words like the phrase. For example, Under Armour is a brand, but in BOW model, it is just two words “under”, “armour” without semantic connection. Lewis (1992) proposed to use the word pair to replace the single word. Single word corp is called uni-gram token. Two words nearby each other is a bi-gram token, and n-gram is for n words nearby group, etc. N-gram method should better understand semantic text meaning than BOW model in theory, but it also brings about huge computation burden. In this Zillow dataset, even only count the frequency of the words larger than 100, it still forms a word vector with length over 1000 in BOW. For bi-gram, the dimension explodes to over 200,000 and most of the bi-gram token has frequency no more than 2. We use bi-gram LASSO model to analyze house price, and find bi-gram variables overfit the training data and have worse out of sample MSE. Thus we choose uni-gram as word representation for the free text.

Besides transforming unstructured text into text, the meaningless stop words are removed from house descriptions. In this essay, we use natural language processing tools NLTK stop words set. Moreover, English words have many morphological variants. (e.g., take, took, taken). From semantical understanding, these morphological variants have the same meaning. We apply PorterStemmer algorithm to transform word to its root form called stemming.

After getting the word representative matrix of whole text dataset, meaningful words are still needed to extract from matrix since most of the words are adjective and verb, which have trivial impact on house price. The most common method to calculate relevancy in documents vectors is TFIDF (term frequency inversed document frequency). TF (term frequency) is expressed as

$$\mathrm{tf_{ij}} = \frac{n_{ij}}{\sum_k n_{kj}}$$

where n_{ij} represents times of term i been used in document j, and $\mathrm{tf_{ij}}$ is the proportion of term i been used to total terms in document j. IDF (inversed document frequency) is expressed:

$$\mathrm{idf_i} = \log(\frac{|D|}{|1 + \{j: t_i \in d_{j\}}|})$$

$|\mathrm{D}|$ is the total number of documents in dataset, $|\{j: t_i \in d_j\}|$ is the total frequency of documents containing term i. Tf-idf index represents by:

$$\mathrm{tf - idf(i)} = \mathrm{tf_{ij}} * idf_i$$

Using this index, it can easily filter words having a high frequency in specific documents, but a low frequency overall, which is widely used to find topics in documents. In this research paper, I apply the tf-idf method to find high-frequency words that describe appliances and facility in property facts and property description.

2.4 Data

Houses data are sampled in 11 different states in the U.S. from Zillow.com. To obtain the representative sample of entire U.S. housing market, data selected largest, median and smallest Metropolitan Statistics Areas (MSA) from each of the four regions of U.S. (Northeastern, Midwest, South, West), and then find all zip codes in selected MSAs, resulting in totally 1894 zip codes. Afterward, from each selected zip code, five houses listed in Zillow.com are randomly picked to form the entire house dataset. In this paper, we only use single-family house with last sold dates from January 2016 to September 2016. The reason to choose this short period is property facts and property description was collected in September 2016. Property owners may remodel the house or purchase appliances after move in. For the last sold date earlier than 2016, sold price is likely not consistent with the property information when collected.

Variables commonly used in previous house price literatures are called basic house information, and it includes number of rooms, number of bathrooms, square footages. Each property was mapped to its closest elementary, middle, high school, and distance to each school. Data of school quality is from GreatSchools.org, where each school is rated on a scale of 1-10 with 10 being the highest standard, based on state standardized test performance compared to other schools in the same state. Based on the listed zip code, we add neighborhood demographic characteristic for each house. Demographic variables include area household median income, area population size and area unemployment rate. Table 2.3 below summarizes basic house variables and their summary statistics.

House structure attributes are extracted from property facts in each house. The tf-idf score in range [0.2, 0.5] contains most property structures category, and score higher than 0.5 contains specific texture, appliance name. We compare the result to Zillow.com website and Ho (2016) paper and summarizes 13 house structure categories and 116 specific variables in total. Detail of category names and variable names are listed in Table 2.4. Each Type in the second column in Table 2.4 is a dummy variable as structured attribute in model. If an attribute type is listed in property facts, the value of that variable in matrix is 1, otherwise it is 0.

Table 2.3. Basic Variables and Summary Statistics

Basic Variables and Summary Statistics (N = 10223)

Variables	**mean**	**Std Dev.**	**Min**	**Max**	**Description**
lnprice	12.75	0.82	9.84	17.69	log(property sold price)
lnMedIncome	11.19	0.38	7.32	17.68	log(house median income in neighbor)
lot_size	6374.92	2,315.71	700	10876	lot size of house
all_time_views	2914.55	2,562.75	60	4002	house all time been viewed
age_at_sale	53.93	27.88	0	274	age of house by sale time
beds	3.28	1.05	0	10	number of bedroom in property
bath	2.25	0.91	0.25	22	number of bathroom in property
sqft	1908.49	823.78	700	331404	square footage
school1_rating	6.57	2.45	1	10	elementary school rating
school2_rating	6.48	2.55	1	10	mid school rating
school3_rating	6.75	2.38	1	10	high school rating
Unemp. Rate	0.07	0.03	0	0.3	unemployment rate in neighbor
n_photos	11.37	10	0	55	number of photos listed in Zillow
logDesc	3.98	0.61	1	4.92	log(number of words in property description)
logFact	4.68	0.63	3.43	6.52	log(number of words in property facts)
Pop	24991.7	19432.8	88	113916	population by zip code

Table 2.4. Summary of Property Structure Facts

Property Category	**Type**
Appliances	dishwasher, dryer, freezer, garbage disposal, microwave, range/ oven, refrigerator, washer
Architecture type	bungalow, cape cod, colonial, contemporary, craftsman, french, georgian, loft, modern, ranch, spanish, split level, tudor, victorian
Basement type	finished, partial, unfinished
Cooling source	central, evaporative, geothermal, refrigeration, wall
External material	brick, cement/concrete, composition, metal, shingle, stone, stucco, vinyl, wood, wood products
Features	attic, barbecue, basketball court, cable ready, ceiling fan, deck, disability access, dock, double pane storm windows, elevator, fenced yard, furnished, garden, gated entry, greenhouse, high-speed internet, hot tub, intercom, jetted tub, lawn, mother-in-law, patio, pond, pool, porch, RV parking, sauna, security system, skylight, sports court, sprinkler system, vaulted ceiling, waterfront, wet bar, wired
Floor covering	carpet, concrete, hardwood, laminate, linoleum/vinyl, slate, softwood, tile
Heating type	baseboard, forced air, heat pump, radiant, stove, wall
Heating source	coal, electric, gas, oil, propane butane, solar, wood pellet
Parking type	carport, garage, garage-attached, garage-detached, off-street, on-street
Roof type	asphalt, built up, composition, metal, shake shingle, slate, tile
Rooms	breakfast nook, dining room, family room, laundry room, library, master bath, mud room, office, pantry, recreation room, sun room, walk-in closet, workshop
View	city, mountain, park, territorial, water
Other resource	school bus service, commuter train, historical district

Besides property structure attributes listed by Zillow, unstructured property description written by owner and agent is also a source of data could uncover impactful house attributes that rarely been noticed before. As mentioned in section 2.3.4, we apply uni-gram model to transform descriptions into the dummy variable. Moreover, stop words set is applied to eliminate meaningless words and signs, apply Porter Stemmer algorithm to group morphological variants in the text. Here uses a sentence in Table 2.2 property description as the example.

"Contemporary cape located 5th from the beach in the very desirable Southport area of Holgat."

Compared to property facts, this sentence has subjective words, such as "contemporary", "desirable", which can be good marketing words that possibly impact house price. More importantly, this sentence uncovers important attribute that previously unobservable without text mining involved: outside view attributes "beach", and its location information which is likely associated with property structure attribute and impacts house price. In general, not every word in context includes unobservable house attribute information, past studies (Bajari et al. (2017)) proposes only partial of words has a significant correlation with the dependent variable, and these variables are correlated to other regressors in equations too. Figure 2.3 shows the top 15 words frequency in all house description contexts.

One fundamental assumption in house description data is the owners deliver correct property information and will not deliberately miss house attributes in their description. All the property information is gathered from Zillow.com, and it is difficult to track down attributes listed as missing but existed in the house. This type of missing information potentially affects actual house sold prices. It is one limitation of this dataset and our model build on this assumption.

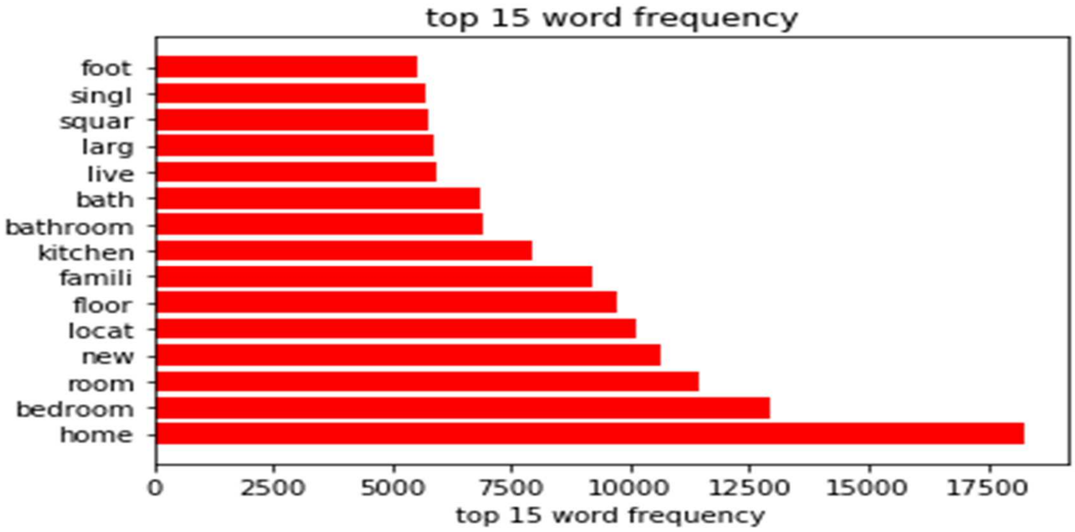


Figure 2.3. Property Description Word Frequency Summary

2.5 Result

2.5.1 Model Setting

Machine learning model tends to overfit the training data and have higher adjusted R-square in-sample fit. Therefore, using mean square error (MSE) of out-of-sample test set to compare prediction accuracy across models is a more reasonable approach. A model that has good out-of-sample accuracy on house price understand the impact on house price and interactions across variables. Training on 75% of observations in the dataset, we propose to compare three different models: post-LASSO, gradient boosting and random forest. Belloni and Chernozhukov (2009) had claimed that, in LASSO model, the correctly specified model should be determined by a small number of variables, which is not fit in this problem. Home buyers consider many attributes that might be unobservable but important factors, and the true model in here should not only include a few variables if it is correctly specified. Based on this reason, we choose to use post-LASSO rather than adaptive LASSO since adaptive LASSO tends to give a much sparser solution.

We run the regression model in 3 rounds with different variables included in each round. In each round, we respectively run Post-LASSO, gradient boosting and random forest model. The first run uses variables that commonly selected in previous related literature. It includes bedrooms, bathrooms, square footage, house age when sold, neighborhood demographic information like population, unemployment rate, household median income, and education information which includes rating and distance to nearby primary school, middle school and high school. In the second run, we add all house structured features extracted from property facts, which are listed in Table 2.4. Besides that, variables only generated in online platform Zillow are also included, like all-time views of the house, number of photos listed, length of property facts and length of property description. In the third run, all unstructured text descriptions are included in the model, 891 informative words in total. After training, respectively fit left 25% test set with model parameters decided by training set and get MSE (mean squared error) of each model. Bajari et al.(2015) have shown that estimators in regression tree models (random forest and gradient boosting) are asymptotic normal and converge to real response value with lower converge rate $\mathrm{O_p}(\frac{1}{\sqrt[3]{n}})$.

Unlike linear regression, all three models need tune hyper-parameter to find model with best out of sample predictability. As mention in section 2.2.1, LASSO needs tune penalty parameter λ with cross validation. After grid search hyperparameter λ, we select λ = 0.015 as hyper- parameter.[3] Random forest randomly selects subset of variables to build tree model. In practice, 1/3 of entire variables is the most popular proportion for regression. For other hyper-parameter, number of trees is set in 2000. In random forest model, setting large number of trees lower the weight of each tree and is unlikely to overfit the training data. In gradient boosting regression tree, number of trees set to be 1000 with minimum observations in node is 75. With a smaller value of minimum observations in leaf, the depth of the tree will be deeper and model will be more robust to nonlinear relationship. Gradient boosting model sequentially builds new trees based on residuals of the previous tree, which makes it easier to overfit than random forest. Thus, a larger value of minimum observations in node is necessary. Learning rate (γ) is another hyperparameter to control overfitting problem in this model, which set to 0.05. The detailed reason why this model is easy to overfit and formula will be discussed in Appendix A.2.

2.5.2 Model Result and Nonlinearity

Table 2.5 shows out of sample prediction accuracy of the three models in three different variables set. Algorithms behind LASSO and gradient boosting model have the procedure of model selection. The number of variable selected in the model shows the scale of relevant variables and is included in column. In the three different rounds, adjusted R-square increases with more variables included in the data. Effect of including house structure attributes is significant by increases 18% of adjusted R-square, and unstructured text increase 4% of additional adjusted R-square. Post-LASSO has the lowest accuracy out of all models, mainly due to its sparsity assumption added on the model. The true model is likely to depend on more variables. Lowering the penalty hyper-parameter λ can obtain a less sparse model. However, this move still cannot detect interactions across variables and nonlinear relationship. Regression tree

[3] Different software has slight difference in computing. In this paper, I use R's glmnet package to compute LASSO result.

model is a better method to handle this issue and shows better result in Table 2.5. Among all three models, gradient boosting has the better out of sample prediction accuracy than random forest.

Figure 2.4 shows the top 20 import features in predicting house price when run gradient boosting model. Relative importance ranks attributes by their contributed importance to the model. It shows that “house age” is No.4 important features in the model. However, the result is inconsistent with findings in linear model regression. Table 2.6 first column shows in OLS model, house age coefficient is insignificant and is not selected if run the post-LASSO model. The discrepancy of two models is more apparent when we draw the partial dependence plots of house age at price in Figure 5. “House age” has a nonlinear correlation pattern on price with ages increases. Linear model has a limitation that the plot fits data only with straight lines and cannot truly reflect the complex pattern.

The partial plot pattern of house age in Figure 5 is like a piecewise function. We choose to split the data into 3 groups: house age smaller than 40 years, house age between 40 - 100 years and house age larger than 100 years. The result shows in Table 2.6. Age coefficient is significantly negative in the group of houses age less than 40 years, and significantly positive in the group of 40 – 100 years. For the case houses age over 100 years, the sign is not significant, but the number of observations is too small to confirm the sign of house ages. Previous literature discusses ages effect should depend on the regions and age of the city. House age has a positive correlation with prices in the large, historical cities which is consistent with finding in this paper.

The advantage of tree based regression is to find nonlinear relationship and interactions across variables without addressing extra assumption or structure. Without using these models, researchers need the domain knowledge or awareness to manually detect and create potential piecewise function in the model, which is unlikely in high dimensional data case. Regression tree model algorithm is a natural fit in high dimensional space and is capable to capture the features that are non-linear correlated with the response variable y, in this case, house price.

Table 2.5. Model Accuracy

Model	**Adjusted R-Squared**	**# Variable Selected**	**Out of Sample MSE**
Post-LASSO			
basic	0.48	19	0.243
+structured	0.66	106	0.235
+unstructured	0.70	390	0.217
Gradient Boost			
basic		19	0.237
+structured		102	0.193
+unstructured		316	0.178
Random Forest			
basic		19	0.226
+structured		141	0.195
+unstructured		936	0.192

Table 2.6. House Sold Age Coefficient in Linear Model

Variable	Coefficient	Observations
age at sale	-0.0002 (0.0002)	7765
age at sale < 40 years	-0.0038*** (0.001)	2454
40 < age at sale < 100 years	0.002*** (0.001)	4876
age at sale > 100 years	-0.0072 (0.005)	450

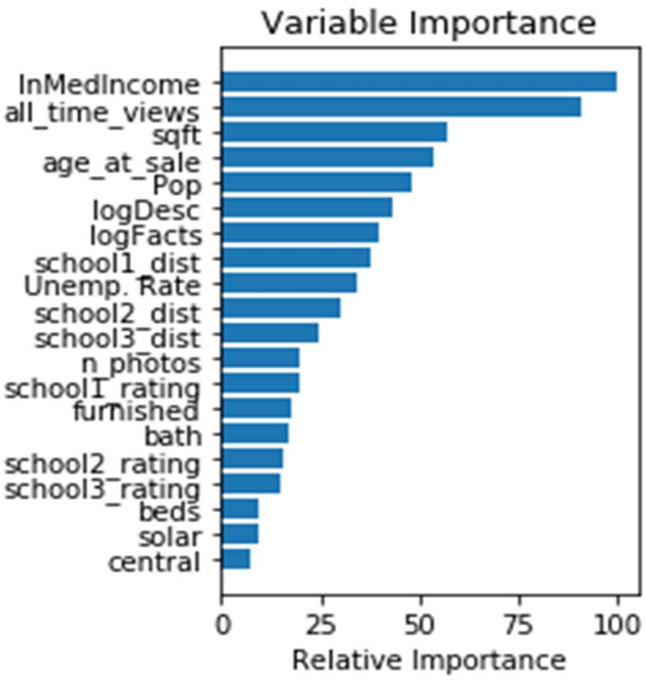


Figure 2.4. Gradient Boosting Top 20 Important Features in Price

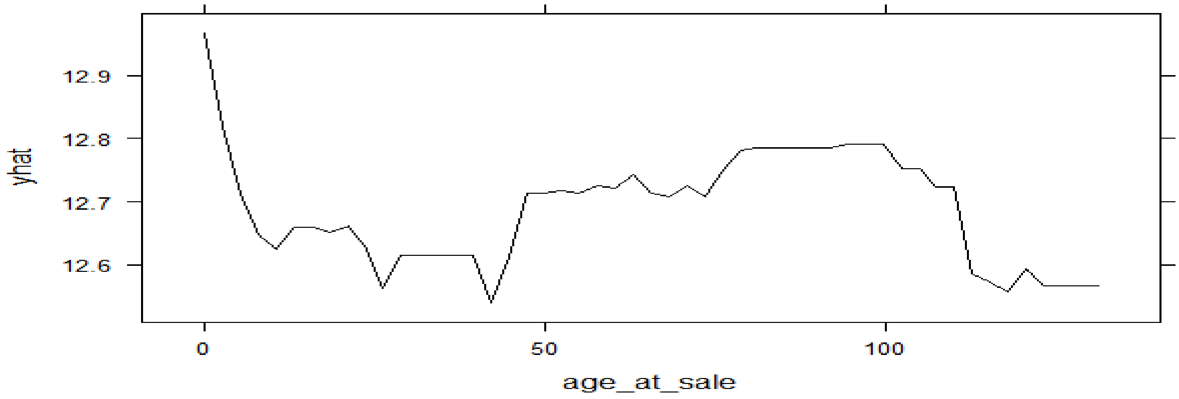


Figure 2.5. House Age Partial Dependence Plot with log(House Price)

2.5.3 Variable Impact

2.5.3.1 Define Variable Impact and Condition Checked

It is a tricky problem to quantify the magnitude of explanatory variable impact on the dependent variable in a tree based model since tree based model does not have model coefficients. Varian (2014) proposes that a good predictive model can be better than a randomly assigned control group in estimating treatment effect and causality. In high dimension data, corrected specified model is difficult to justify, more researcher focus on using machine learning model selection to investigate average treatment effect and heterogeneous effect. Athey and Imbens (2016), Athey et al.(2016), Wager and Athey (2017) had proposed the feasibility of using gradient boosting, random forest model to estimate heterogeneous treatment effect.

If a selected model is predictive in the entire dataset, it can also predict a model in a counterfactual case, where the treatment group is not treated. Moreover, the difference between the real observed value and predicted value in the counterfactual case can be interpreted as variable impact on house price. In this paper, the difference between real case and the counterfactual case can be regarded as the implicit price of interested attributes in housing model. In detail, we train and evaluate the best regression tree model in data with mix treated and untreated observations. Then we feed the test set with the best-performed model, and get the predictions as counterfactual result. Therefore, the ideal average factor effect will be similar as ATE in treatment:

$$\mathrm{ATE} = \frac{1}{N_{treat}} \sum_{1}^{N^{treat}} (Y_{treat} - \hat{\mathrm{f}}_{\mathrm{counterfctual}}(x)),$$

$\hat{\mathrm{f}}_{\mathrm{counterfctual}}(x)$ is predicted value of treatment group on the control group, mean difference in treatment group with number N_{treat} is the average treatment effect ATE.

In this paper, we are interested in the outside view impact on the house price. To be specific, compare and analyze the price impact between property facts have view listed and not listed. The implicit price of outside views is difficult to define and rarely discussed in previous literature. Lansford and Jones (1995) estimated the recreational and aesthetic value of water, but their paper focused on environmental perspective, and views like mountain, city skyline, park are not included in their paper. Athey et al.(2016) have theoretically proved tree-based estimator of treatment effect is consistent and asymptotic normal if the treatment is unconfounded and the tree's construction follows the honesty assumption. Though not the same scenario as their paper, it is good theoretical support to claim that we can evaluate a single factor impact on house prices.

This work evaluates the impact of outside views on house prices, and it is known that outside views belong to natural scenes and rarely is decided by other house variables, like squared feet, number of bedrooms, number of bathrooms. If we construct the tree following the rule of honesty tree assumption in Athey et al.(2016), the estimator is asymptotic converge with rate $\mathrm{O_p}(\frac{1}{\sqrt{\mathrm{n}}})$. The detail of honesty assumption will be put in Appendix A.3.

There are 5 different types of views listed in Zillow’s dataset: city, mountain, park, territorial, water. Based on the way constructing variable for house attribute, each different view is one dummy variable with binary value 0 or 1. Thus houses with at least one kind of view listed can be considered as treated observations, and houses without any view are control group observations. Then it is feasible to predict the implicit view price by taking the difference between treated group and its counterfactual predicted price i.e. houses without a view.

We compute the implicit view price by using gradient boosting regression tree on entire house data with structure variables but excluding unstructured text. The reason behind this decision is description sometimes includes address or view information, which might highly correlate with the interested variable. For a group of highly correlated variables, gradient boosting tree model chooses only one variable in that group for reducing RSS, it is possible that view is never selected in the group and cause potential bias. Based on the above way to construct GBDT model, we observe that houses with views have a significant increase in price.

One of concern about using tree model to evaluate implicit price is that tree models cannot fully explain causality between response and explanatory variable. A good binary feature should control selection bias since the distribution between treatment and control is not randomly designed. It is crucial to check if potential two group’s bias variables are included in the training model. We run the model with the view as classifier and all other variables as predictors excluding unstructured text in a gradient boosting decision tree model in Figure 2.6. Loss function is the only difference between gradient boost decision tree (classifier) and regression tree, detail about decision tree can see Friedman(2001).

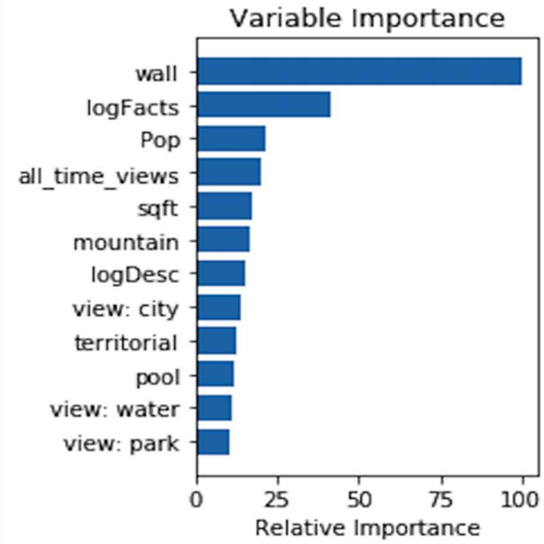


Figure 2.6. Top Important Variable to Binary Feature View

Table 2.7. Group Statistics Summary

Variable	Control: No view	Factor: with view	Welch t-value
Price	312,200 (232,452)	556,933 (221,087)	28***
Age at sale	54.23 (28.2)	52.49 (26.8)	1.90*
Square footage	1884.6 (808.3)	2020.2 (884.5)	5.2***
Bath	2.2 (0.90)	2.4 (0.98)	1.91*
Beds	3.26 (1.10)	3.36 (0.95)	3**
Log Median income	11.186 (0.385)	11.204 (0.356)	1.8
Observation	6395	1366	

Observational data have limitations that treated observations and non-treated are not randomly assigned. In observational data, assuming two groups of observations have identical distribution is not realistic. Table 2.7 shows, for basic house variables, under Welch's unequal variance t-test, two groups of observations have nearly equal means in most of them except square footage. Even in case that two groups are not identical, control groups have slightly broader bandwidth in most columns. Control groups have enough overlap to approximately match most of the treated observations and to compute the counterfactual effect for each treated.

2.5.3.2 Implicit Price of View

In this section, we try to predict how outside views affect house price. Figure 2.7 and Figure 2.8 show the comparison between top 20 important features in gradient boosting and random forest model that affect the price. Pretreatment correlated variable is important to price in the model and is controlled in the model. In gradient boosting, relative importance is measured and ranked by magnitude of a cutoff point on a variable that reduces the mean square error. For random forest, the criterion is similar: variable importance is ranked based on how much MSE is increasing if drop this variable from the model. Compared to top variables in both models, there are 18 variables overlapped in both sides, indicating most top features are essential to decide house price.

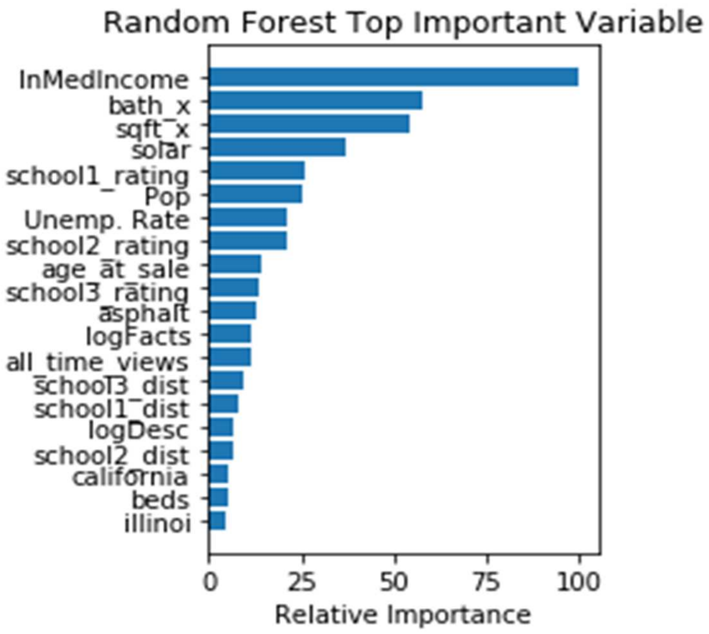


Figure 2.7. Random Forest Top 20 Features

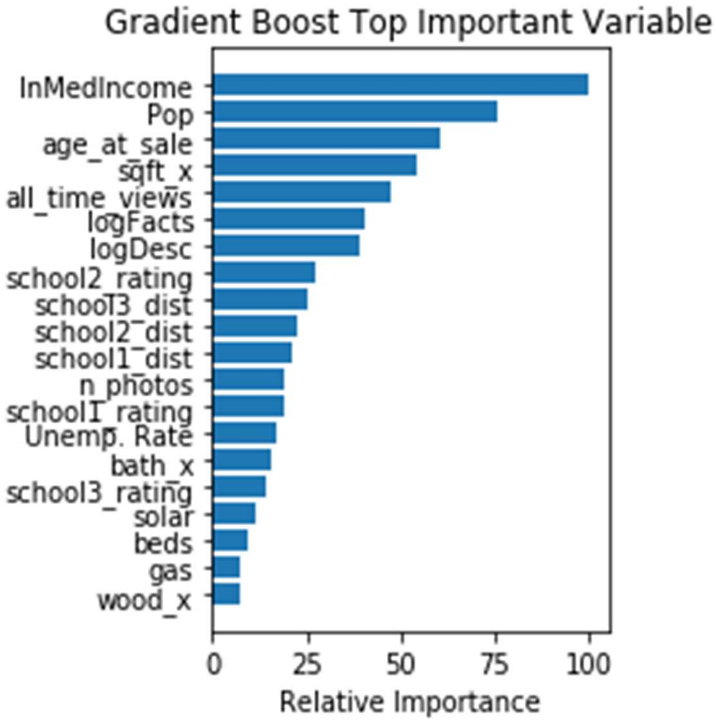


Figure 2.8. Gradient Boosting Top 20 Features

One possible explanation that random forest model performs worse than gradient boosting is random forest uses subset to construct trees. Majority of variables in both models are words and corresponding word matrix is full of zeros. Random Forest randomly subsets 1/3 of variables to build tree in each iteration, which likely has cases that most selected variables are words and has limited predicting power. Average prediction accuracy would be lower when this case happens many times in iterations. That also explains why random forest out of sample MSE does not drop as much as gradient boosting when unstructured text variables are added.

2.5.3.2.1 Predicted Effect

Predicted average factor effect (i.e. ATE) is listed in Table 2.8. Table 2.8 shows the change in factor effect with more variables added in the model. Most of the view attributes have a positive impact on price which is consistent with our analysis. The only exception is the view of the park has a negative impact on house price. Explanation from previous literature suggested that park not only means space for recreation, sometimes means space for more crime or illegal events, which brings more concern to home buyers. Our result partially collaborates to this paper.

Since gradient boosting is the best predictable model, we show the change in implicit price as more variables added in gradient boosting model. The result shows that adding unstructured text variables correct the predicted implicit value of price since view impact is lower when text information is added in the model. Similarly, compared to the model with only basic information included, adding property structure variable reduces the price difference between view and no view houses. The possible reason is omitted noises have a positive correlation with views e.g. city, mountain, park, water et al. Positive correlation causes the upward bias. The omitted noises are likely observable when model includes more text mining variables and outside views influence on house price are corrected downward.

A good example is the houses A and B example illustrated in section 2.1(Table 2.1, Table 2.2, Figure 2.1). With water view attribute included, house B's sold price is much higher than house A's. House B's property description includes its location information, a private deck, and it has remodeled recently. Unstructured text information like location and extra amenities have a positive correlation with water view and positive impact on price, but are rarely included and are treated as noises in previous research. Predicted impact of water view is upward bias if model runs only on the structured variables.

Table 2.8. Model Average Predicted Effect by View

Model	View	City	Mountain	Park	Water	Territorial
Basic Variable	0.3209					
Gradient Boost						
Add Structure	0.0891	0.1152	0.1173	-0.0637	0.1188	0.0739
Add Unstructure		0.0793	0.1033	-0.0663	0.07396	0.0653
Random Forest		0.3072	0.2378	-0.0263	0.1435	0.0801
Post-LASSO		-0.07	0.0614	-0.064	0.252	0.0198
Sample		303	421	168	308	211

Table 2.8 also computes the factor impact on LASSO and random forest model for comparison. The comparison results show at the bottom of Table 2.8. Both models use full variables including unstructured text variables. In general, both comparison models have the same sign of factor impact as gradient boosting in each column, just at different magnitude. From the table, random forest overestimates the factor impact in views. The reason behind is

random forest randomly selects subset (1/3) variables to train the model, and most of the variables are the descriptive words. For example, when subset only includes water view but excludes correlated structure variable like swimming pool, or words like “private deck”, model would assume water view explain all the price variations and overestimate its impact. In general, random forest has decent prediction accuracy, but its algorithm has limitations in explaining factor impact and treatment effect if existed. Post-Lasso leads to a not very sparse model, still has fewer variables selected than gradient boost, resulted in low prediction accuracy and underestimate the impact from view.

Table 2.9. Regression on Average Impact of View

Variable	**Water**	**Mountain**	**City**	**Park**	**Territorial**
age_at_sale	-0.006 (0.003)	0.001 (0.002)	-0.0017 (0.002)	0.001 (0.007)	0.001 (0.003)
bathroom	-0.059 (0.138)	0.031 (0.055)	-0.1032 (0.078)	-0.011 (0.352)	0.124 (0.126)
beds	0.180 (0.123)	-0.048 (0.063)	-0.0728 (0.088)	-0.081 (0.179)	0.193** (0.087)
sqft	-0.0001 (0.000)	5.084e-05 (6.59e-05)	0.0001 (0.000)	0.0002 (0.000)	0.0001 (0.000)
Pop	-8.574e-06 (5.18e-06)	2.636e-06 (2.12e-06)	5.392e-08 (4.1e-06)	9.098e-06 (1.09e-05)	0.0001 (0.000)
Unemployment	-3.946 (3.286)	-2.212 (1.920))	6.0413 (3.829)	-4.045 (5.30)	-6.57e-06 (4.2e-06)
MedIncome	-0.641 (0.424)	-0.3151* (0.167)	0.2188 (0.239)	0.098 (0.648)	-0.212 (0.375)
Adjust R-Square	0.007	0.007	0.20	0.002	0.24

In Table 2.9, we plan to examine whether heterogeneous treatment effect exists in the model. We regress the price variation, which is predicted treatment effect, on all of basic house attributes. If there are basic attributes that have a nonlinear correlation with specific views attribute, then their coefficients should be significant in this price variation regression. Results in Table 2.9 shows that all coefficients in views are insignificant at 95% in each row except one.

Basic house attributes are not strongly correlated with treatment, implies outside views implicit price is unlikely changed with basic house attributes.

2.5.3.2.2 Binary Attribute Impact

Table 2.8 shows the view impact on house price is trending downward when more structure house attribute and property description are included in model training. Table 2.9 indicates that basic house attributes do not strongly correlate with the treatment effect, which means structure and unstructured variables are likely correlated with views. However, it is challenging to disentangle all these features since most of the variables are binary. Moreover, in high dimensional space, it is even more unlikely to identify the effect of each attribute separately. The tree based structure has no direct coefficient making the disentanglement even more difficult. With these restrictions, we use Table 2.10 to show the approximate directional impact of these binary features on house price, which hopefully enlighten other researchers heuristically. The calculated price increase does not mean its correct implicit price, but it shows the approximated lift with having the features when average all other correlated or uncorrelated variables.

Table 2.10 selects several important binary attributes or binary words for each kind of view and plots their partial dependence on price**.** Table 2.10 shows that view has a higher correlation with words that convey similar information, like water view associates with words like “beach”, “pool”, “river”. For words description, it is unlikely to identify the level of lift is words’ implicit price, especially for marketing words. The words used here shows that its direction of impact on price when it associates with correlated structure attributes. Words description with location information like “California” raise the price significantly. However, marketing words have a distinct impact on price. General sentimental words like “best”, “big”, "spacious" have a negative sign, whereas words like "best", "beautiful" shows a positive correlation. One thing that needs to remind is unstructured texts have lifts of price do not equal to its implicit prices. Text variables are rarely included in previous house pricing literatures, and these variables are strongly associated with price and help us better understand home buyers' decision.

Table 2.10. Important Binary Feature Price

Free Text Feature	w/o Feature log(price)	w/ feature log(price)	price raise ($)
lake	12.75	12.68	-4,200
beach	12.75	12.89	23,400
pool	12.74	12.79	15,800
hardwood	12.74	12.78	14,000
hill	12.75	12.79	10,900
California	12.74	12.88	13,700
security system	12.75	12.77	7,300
garbage disposal	12.75	12.76	2,700
best	12.75	12.74	-4,200
beauti	12.75	12.76	2,500

2.6 Conclusion

Previous hedonic topic literatures assume the price of real estate has a linear relationship with all its observed attributes and unobserved attributes. Previous literatures only have a subset of variable available and study the topic in low-dimension approach. In this case, hedonic models are prone to omitted variable bias if key attributes of house are not included in the data. In this paper, we find property structure attributes and unstructured text listed in online real estate platforms are relevant in house purchase. This source of data is rarely studied since it lacks tools to extract unstructured data into the appropriate form. We show after appropriate preprocessing with text mining approach, including this source of information increases our model explainable ability.

Moreover, very few previous literatures discuss how to specify models in high dimensional data since previous linear model selection approaches are prone to overfitting and fail to perform well in out of sample prediction. Machine learning models provide ideas of using the data-driven model to solve the economic and business problem. We show these models are suitable for model selection in high dimensions and automatically explore the nonlinearity and interactions across variables. With the help of partial dependence plot, nonlinearity across variables can be visualized and better to be analyzed for further research. Results also show regression tree models have better performance on out of sample predicting prices. This paper offers a new pipeline from data extraction to model selection and model prediction which can be widely applied to many different areas, like house rental, automobile transaction and electronic commerce such as Amazon, eBay.

Another highlight of this paper is to predict the attribute's implicit price in high dimensional data. Borrowing treatment effect idea, implicit price can be computed as average treatment effect by the treated group. This approach also has practical implications for real estate market professionals to evaluate house price better. In this paper, we show the predicted implicit price of all kinds of views. This idea can be applied to evaluate the implicit price of other house amenities and helps real estate agencies to give recommended price to both home buyers and sellers.

References

Athey, S., & Imbens, G. (2016). Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, *113*(27), 7353-7360.

Athey, S., Tibshirani, J., & Wager, S. (2016). Solving heterogeneous estimating equations with gradient forests. *arXiv preprint arXiv:1610.01271*.

Ba, S., & Yang, X. (2016). Zillow—Online Media Tycoon in US Real Estate Brokerage Industry. *In "Internet Plus" Pathways to the Transformation of China's Property Sector (pp. 67-84). Springer, Singapore.*

Bajari, P., Nekipelov, D., Ryan, S. P., & Yang, M. (2015). Demand estimation with machine learning and model combination (No. w20955*). National Bureau of Economic Research*.

Belloni, A., & Chernozhukov, V. (2013). Least squares after model selection in high-dimensional sparse models. *Bernoulli*, *19*(2), 521-547.

Belloni, A., Chernozhukov, V., & Hansen, C. (2014). Inference on treatment effects after selection among high-dimensional controls. *The Review of Economic Studies*, *81*(2), 608-650.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-123

Dubin, R. A., & Goodman, A. C. (1982). Valuation of education and crime neighborhood characteristics through hedonic housing prices. *Population and environment*, *5*(3), 166-181.

Gentzkow, M., Shapiro, J. M., & Taddy, M. (2016). Measuring polarization in high-dimensional data: Method and application to congressional speech (No. w22423). *National Bureau of Economic Research*.

Hughes, G., & McCormick, B. (1994). Did migration in the 1980s narrow the North-South divide?. *Economica*, 509-527.

Kain, J. F., & Quigley, J. M. (1970). Measuring the value of housing quality. *Journal of the American statistical association*, *65*(330), 532-548.

Kang, J. S., Kuznetsova, P., Luca, M., & Choi, Y. (2013). Where not to eat? Improving public policy by predicting hygiene inspections using online reviews. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 1443-1448).

Knight, K., & Fu, W. (2000). Asymptotics for lasso-type estimators. *Annals of statistics*, 1356-1378.

McLeod, P. B. (1984). The demand for local amenity: a hedonic price analysis. *Environment and Planning A*, *16*(3), 389-400.

Lee, Y. S., & Sasaki, Y. (2014). How Sensitive are Sales Prices to Online Price Estimates in the Real Estate Market?

Lewis, D. D. (1992, June). An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 37-50). ACM.

Nowak, A., & Smith, P. (2017). Textual analysis in real estate. *Journal of Applied Econometrics*, *32*(4), 896-918.

Rosen, S. (1974). Hedonic prices and implicit markets: product differentiation in pure competition. *Journal of political economy*, *82*(1), 34-55.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, *24*(5), 513-523.

Sirmans, S., Macpherson, D., & Zietz, E. (2005). The composition of hedonic pricing models. *Journal of real estate literature*, *13*(1), 1-44.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267-288.

Varian, H. R. (2014). Big data: New tricks for econometrics. *Journal of Economic Perspectives*, *28*(2), 3-28.

Wager, S., & Athey, S. (2017). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*.

Weigher, J. C., & Zerbst, R. H. (1973). The externalities of neighborhood parks: an empirical investigation. *Land economics*, *49*(1), 99-105.

Xiao, Y. (2017). Hedonic Housing Price Theory Review. In *Urban Morphology and Housing Market* (pp. 11-40). Springer, Singapore.

CHAPTER 3. EXAMINE MARKET COMPETITION IN ONLINE HOTEL BOOKING MARKET

3.1 Introduction

With the emergence of online travel platform like Expedia.com, customers start to rely heavily on Internet-based resources to book flights, reserve hotels, and rent cars. Expedia hitting 50 million clicks per month is good evidence to show the popularity of these online resources. In 2015, there are over 148 million traveling booking orders executed through the online platform. 57% of travelers book their reservations on Internet. And the traveling industry revenue has soared to $762 billions by the end of 2018.

Like competitions in other markets, hotel industry is not controlled by several branded hotels any more. More emerging independent hotels enter the market and make themselves available and appealing from the internet. The hotel industry has numerous sellers who provide differentiated products/service to customers. In major metropolitan areas like New York, hundreds of branded and independent hotels compete with each other by offering differentiated services, amenities, etc. The competition across hotels has become fiercer since searching cost for hotel availability is considerably less using online platforms. The similar trend happens in other areas like cars, second-hand goods e-commerce, and real estate property industry.

In a market with a large number of competitors like hotel industry, it is interesting to see how hotel identifies its targeting competitor set among all hotel candidates and make according pricing strategy for the competitors. It is highly impossible that firm would track every other competitors' price variation and make according adjustment for every price change in the market. The cost to keep tracking and adjusting price could be significantly huge, which make it is not practical to do. For researchers, it is also challenging to analyze the market competition pattern since some demand and supply shocks that are only observed by hotel revenue managers but not by researchers.

However, on the other hand, identifying the right competitors is a critical part for the hotel to make short and long term revenue management plan. Lack of effective analytical approach might lead to inaccurate response to demand shocks, especially price changes. Also, identifying wrong competitors can generate significant revenue loss since customers would switch to other hotels that have similar services.

Traditional marketing and industrial organization papers mainly focus on competition effect between a small number of competitors and solve a low-dimensional problem in most cases. The previous literature methodologies do not fit for a large number of competitors' problem and rarely use variables selection in their modeling. By borrowing the statistical learning approach, we study the hotel competition pattern in a high dimensional dataset. The approach I propose can find the critical factors that explain price variation and detect competition pattern across the hotels.

The proposed method I use in this paper is the simultaneous system of equations, where hotel price is jointed determined by the function of prices from all other competitors. Classic ordinary least square (OLS) regression cannot handle this case correctly and has two concerns: simultaneous endogeneity and high dimensionality.

Endogeneity concern starts from the system of simultaneous equations for prices. One competitor's hotel daily price is linearly affected by other competitors' price in the market, and vice versa. Since other competitors jointly determine prices, it will be a problem to regard other competing hotel prices as exogenous variable and straightly use OLS to estimate coefficients. Moreover, OLS regression result can only represent correlation but not causality between competitors. The econometric solution for this problem is to find valid instrument variables, which need to correlate with one competitor's price explicitly but not correlate with others' price equations.

A good instrumental variable is often tricky to find since there are hundreds of competitors in the market, and each one of them needs valid instrument variables. Furthermore, a good instrument cannot be too weakly correlate with the dependent variable. It can explain the daily price variation. The feasible instrument variable we propose in this paper uses number of times hotels have been searched and displayed to customers as instruments. When a customer searches the hotel with filters and re-sort result by orders, for example, filter hotels at least 3 stars and sort by distance to time square, a hotel candidate set is generated and sorted for the customer to check. Aggregating all search behaviors in the dataset, we can obtain hotel-level demand and construct hotel specific demand and valid instrument variable. Furthermore, filtering and sorting also possibly use price as their criterion, which might cause endogenous bias. Thus, we only keep the searches without using price for preventing potential instrument variable invalid issue.

The other concern is high dimensionality. It is difficult for researchers to estimate hundreds of coefficients in the model accurately. Getting the exact coefficient is not needed and not efficient. It is not needed since, in a high dimensional problem, straightforward OLS regression without restriction is prone to overfit the data but still not discover the truly relevant variables. No matter from the research perspective or from maximizing revenue perspective, identifying true competitors would be more crucial than obtaining exact coefficients. It is not efficient since multicollinearity is likely an issue when number of variables are over 50. The VIF (variation inflation factor) is likely outstanding, which indicates the significant level of coefficients are not reliable in regression. With considering the above two issues, a method is needed to balance the tradeoff between model bias and variance. Regularization methods are widely used to prevent model complexity and still keep a good model fit. Unlike AIC and BIC which is widely used approaches in econometrics, we propose to use a data-driven method LASSO (least absolute shrinkage and selected operator) to regularize the model complexity, and we show the model has better performance in fitting the data. The objective for these regularization methods is to penalize all non-zero coefficients and let them shrink towards zero. The method usually can get a sparser solution which makes the model clearer for researchers and decision-makers to identify price influencers in the market.

In this paper, we use customers' hotel search records in Manhattan area from Expedia.com. There are a few contributions from this paper. Firstly, we propose a method that can analyze competition pattern and estimate coefficients of system equations even in the high dimensional situation. We decide to use number of times of hotel displayed to customers as IV(instrument variable). LASSO method gives a parsimonious solution which clearly identify each hotel's competitors in the market. Secondly, though in a market with hundreds of competitors, we find each hotel price only responses to a few of competitors. For the pattern of real competition, same star rating hotels would be likely the price influencers for most cases. Hotels will more likely make response to price adjustment from same star rating hotels than that from same region hotels. Moreover, economy hotels like 3 and 4 stars have more competitors than other star rating hotels. Third, besides competition, travel characteristics like travel destination date and days of advanced booking are also essential attributes for daily price variations. In the end, this competition analysis approach also can be adapted to other similar industries who have hundreds of competitors like e-commerce market like online shopping.

The rest of paper is organized as follows. In section 3.2, we review the related literature. In section 3.3, we introduce the methodology used in this paper. In section 3.4, we summarize the Expedia.com hotel search data. Section 3.5 shows the result of system equations and analyze the competition pattern in the market. Section 3.6 concludes the paper with discussions on contributions and further area can be applied.

3.2 Related Literature

Our literature review discloses previous formal research on general industrial organization topic, high dimensional methodology, and related modern empirical work involves in online data and high dimensional problem. The first area of study is from industrial organization and marketing. Market competition theory starts from game theory and optimal decision by monopolistic seller (Tirole, 1998). For the empirical competition, Berry (1995) estimate local market demands using linear random utility model with cross price-elasticity, applying the model to data from the automobile market. Many research papers adopt random utility model from here to analyze industry competitions, but this method is more suitable for the case with limited choice. Expand choice set to high dimension makes function highly nonlinear and convergence to global optimal is unlikely. In the empirical literature, researchers use various ways to differentiate each seller in the market. When choice set is too large, no single competitor has information to estimate all interactions between competitors since it will be higher order (time complexity: $O(n^2)$). Therefore, researchers prefer to reduce dimensions and simplify the complexity of the model.

To study how to reduce dimensionality, the second area of literature review is about linear regression models in high dimensional data. In the high dimensional space problem, researchers choose regularization method to select variables in the model. The most widely used approach in the statistics area is Lasso (least absolute shrinkage and selection operator) which induces to shrink some model coefficients to zero (Tibshirani (1996)). Knight and Fu (2000) studied asymptotic results for LASSO coefficients. Belloni and Chernozhukov (2013) proposed to use post-Lasso approach to get convergence oracle rate with smaller bias. After LASSO been well known, many researchers try to modify its structure from sparsity perspective. Zou (2006) claimed a new approach called Adaptive LASSO that obtains a sparser model and still got oracle convergence rate. Yuan and Lin (2006) proposed group LASSO that clusters high dimensional variables into lower dimensional groups to get a parsimonious result. Simon et al. (2013) studied a sparse-group LASSO which can get sparsity both within group and across groups. Later in section 3, we will give more detail about these models and their performance in this dataset.

The third area of literature review is the empirical application of high dimensional problem. Economists start to turn their attention in high dimensional and big data when Varian (2014) claimed the data-driven machine learning approaches are applicable for demand estimation and policy effect estimation. Bajari et al.(2015) compared multiple data-driven machine learning approaches' convergence rate and prediction accuracy with grocery store data. Rudin and Vahn (2015) used regularization approach to solve optimal quantity in vendor for news problem. Li and Netessine (2012) used data mining and association rule to characterize demand competition network. Belloni et al. (2013) first proposed to use LASSO in endogeneity problem with variables to select. In operational management area paper, Li et al. (2012, 2016) proposed to use online click streaming data as instrument variable for demand. Unlike their setting, our paper uses multiple LASSO methods to estimate not only specific hotel competition pattern, but also pattern of a group of hotels sharing similar traits.

3.3 Model

In this section, we will introduce the methods that have been used for this paper, including LASSO (least absolute shrinkage and selection operator), adaptive LASSO, group (sparse) LASSO, and 2 stage inclusion (2SRI) estimation in this section.

3.3.1 System of Simultaneous Equations

Consider a city with N hotels, let $\mathrm{P}_{1\mathrm{jt}} \ldots P_{ijt}, \ldots P_{Njt}$denote the price per night charged by each hotel i for travel on date j booked at booking data t. As we suggested before, i characterize hotel price is jointed determined by its competitors' prices, hotel specific attributes, traveling date characteristics and error terms, like expression below:

$$\mathrm{P}_{1\mathrm{jt}} = \alpha_i + \beta_{11}P_{1jt} + \beta_{12}P_{2jt} + \cdots + \beta_{1,i-1}P_{i-1,j,t} + \beta_{1,i+1}P_{i+1,j,t} + \beta_{1N}P_{Njt} + \gamma_1 X_{jt} + \epsilon_{1jt}$$

$$\ldots$$

$$\mathrm{P}_{\mathrm{ijt}} = \alpha_i + \beta_{i1}P_{1jt} + \beta_{i2}P_{2jt} + \cdots + \beta_{i,i-1}P_{i-1,j,t} + \beta_{i,i+1}P_{i+1,j,t} + \beta_{iN}P_{Njt} + \gamma_i X_{jt} + \epsilon_{ijt} \qquad (3\text{-}1)$$

for i = 1,2,3,…N

In equation (3-1), it describes how hotel i's price per night is determined by i's competitors and other factors. α_i denotes a hotel-specific fixed effect, like star rating or room capacity or whether hotel is chain branded. $\mathrm{P}_{\mathrm{ijt}}$ is hotel i's average lowest price charge per night for travel on date j and searched action started in date t. Each search date and travel date composes a unique two element travel date tuple: (travel date j, booking date t). Each tuple has two elements in the same order. β_{ik} is the parameter for variable $\mathrm{P}_{\mathrm{ijt}}$ and represents hotel k's degree of impact to hotel i. β_{ik} is hotel pair specific and does not vary by travel date tuple. X_{jt}

represents attributes in travel tuple. It includes travel date of week and search date of week, and days of advanced booking. Date of week has 7 possible values: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday. We use a series of dummy variable with binary value to represent this information. The reason to use dummy variable is hotels might have different pricing strategy for weekday and weekend, which can distinguish leisure travelers and business travelers. X_{jt} also includes days of booking ahead of travel. Like most industry, the earlier booking always cost less than late booking. X_{jt} is vector of travel specific characteristics. γ_i is hotel i specific parameter for vector X_{jt}. ϵ_{ijt} represents unobserved noise shocks that might correlated across hotels.

This additive form without any structure assumption can directly use OLS to get coefficients and apply LASSO regression to do variables selection. However, this system of equations has concerns of endogeneity and cannot identify coefficients β_{ik}. Wooldrige (2010) explained simultaneous equations' identification restriction and condition requirement. For system equations (3-1), it needs at least one exogenous variable that is uniquely correlated with one equation in the system but uncorrelated with others. Adding this exogenous variable into the equations, system equations would be like:

$$\mathrm{P_{1jt}} = \alpha_i + \beta_{11}P_{1jt} + \beta_{12}P_{2jt} + \cdots + \beta_{1,i-1}P_{i-1,j,t} + \beta_{1,i+1}P_{i+1,j,t} + \beta_{1N}P_{Njt} + \gamma_1 X_{jt} + \delta_1 U_{1jt} + \epsilon_{1jt}$$

…

$$\mathrm{P_{ijt}} = \alpha_i + \beta_{i1}P_{1jt} + \beta_{i2}P_{2jt} + \cdots + \beta_{i,i-1}P_{i-1,j,t} + \beta_{i,i+1}P_{i+1,j,t} + \beta_{iN}P_{Njt} + \gamma_i X_{jt} + \delta_i U_{ijt} + \epsilon_{ijt}$$

(3-2)

for i = 1,2,3,…N

Different from equation (3-1), equation (3-2) U_{ijt} can also be regarded as an instrument variable that specifically corresponds to hotel price $\mathrm{P}_{\mathrm{ijt}}$. It is difficult to find a good instrument variable since most of demand shocks are only observable for hotel decision makers but not for researchers. Hotel specific attributes like room capacity are not varied by travel date tuple and thus is a invalid instrument.

However, in customer online search dataset, we are capable to observe how many times the hotel has been exposed to customers for a specific travel date tuple. It represents the demand that this hotel has fit for online customer request and is a candidate for customer's choice set. It is not necessary to be observable by hotel decision makers since instrument variables only need correlate with overall demand shock. To be a valid instrument variable, equation only need one shock is uniquely mapping to one hotel in each travel date tuple, so that it can be distinguished in each equation.

The only concern in here is we can only observe demand from one data source, i.e. Expedia.com. Hotels from Expedia.com is a partial demand shock since hotels are also exposed to other travel website, like Priceline, etc., which is not observable. However, only observing one demand shocks from one channel should not be a concern for equation identification if conditional independence holds between two different channel shocks. Conditional independence implies no cross-equation correlation between hotels and noises, like $\mathrm{E}\left(\mathrm{U}_{\mathrm{ijt}} * \epsilon_{kjt} | U_{kjt}\right) = 0$. For a simple example, unobservable demand shock ϵ_{kjt} could represent hotel displayed in another traveling platform like Priceline.com. Unobservable shock ϵ_{kjt} and ϵ_{ijt} could be correlated, and Priceline's hotel k's shock ϵ_{kjt} could correlate with hotel k's Expedia observed demand U_{kjt}. However, the Priceline's hotel k's shock ϵ_{kjt} will not directly impact Expedia's hotel i's observed times of display.

The most common approach to solve the system equations is Two Stage Least Square (2SLS). In 1st stage, we run OLS regression with endogenous variable $\mathrm{P_{ijt}}$ as dependent variable on exogenous variable $\mathrm{X_{jt}}$ and instrument variable $\mathrm{U_{ijt}}$. We can get predicted $\widehat{\mathrm{P}}_{\mathrm{ijt}}$ and residual $\mathrm{e_{ijt}}$ from equation (3-3):

$$\mathrm{P_{ijt}} = \alpha_{\mathrm{i}} + \gamma_i X_{jt} + \delta_i U_{ijt} + \epsilon_{\mathrm{ijt}} \tag{3-3}$$

In 2nd stage, typical approach (Wooldridge, 2010) fills predicted value $\widehat{\mathrm{P}}_{\mathrm{ijt}}$ into equation and shown as equation (3-4):

$$\mathrm{P_{ijt}} = \alpha_{\mathrm{i}} + \beta_{i,-i}\, \widehat{\mathrm{P}}_{\mathrm{-i,jt}} + \gamma_i X_{jt} + \epsilon_{\mathrm{ijt}} \tag{3-4}$$

Terza et al. (2008) proposed a new method that includes both $\widehat{\mathrm{P}}_{\mathrm{ijt}}$ and $\mathrm{e_{ijt}}$ in equations. 2SPS (two stage predictor substitution) only use predicted $\widehat{\mathrm{P}}_{\mathrm{ijt}}$ in equation, which is likely to miss information from error term and leads to inconsistent results with simulation data. They explained the reason behind it is that first stage predicted value insertion only capture the partial variation of the original variable, and it is possible that predicted variable is not selected in 2nd stage regression because the first-stage predicted value captures only partial variation of original $\mathrm{P_{ijt}}$ and missed other key variations. In generic nonlinear form equations, Terza et al. (2012) had shown 2SRI is generally statistically consistent and 2SPS is not.

The 2SRI version of equation is like equation (3-5):

$$\mathrm{P_{ijt}} = \alpha_{\mathrm{i}} + \beta_{i,-i}\, \mathrm{P_{-i,jt}} + \sigma_{i,-i} e_{i,-i} + \gamma_i X_{jt} + \epsilon_{\mathrm{ijt}} \tag{3-5}$$

If write in general matrix version, it would be like:

$$\text{2SPS: } \mathrm{P} = \alpha + \beta\widehat{\mathrm{P}} + \gamma \mathrm{X} + \epsilon$$

$$\text{2SRI: } \mathrm{P} = \alpha + \beta \mathrm{P} + \sigma \mathrm{e} + \gamma \mathrm{X} + \epsilon$$

where α ,P, $\widehat{\mathrm{P}}$,U, ϵ, e are matrix with JT*N dimensions $\alpha = [\alpha_1, \alpha_2, \dots \alpha_N]$, $\mathrm{P}_{\mathrm{JT}*\mathrm{N}} = [P_1, P_2, \dots P_N]$ and $\widehat{\mathrm{P}} = [\hat{P}_1, \hat{P}_2, \dots \hat{P}_N]$, U= $[U_1, U_2, \dots U_N]$, $\epsilon = [\epsilon_1, \epsilon_2, \dots \epsilon_N]$,

$$, B_{N\times N} = \begin{bmatrix} 0 & \beta_{12} & \dots & \beta_{1N} \\ \beta_{21} & 0 & \dots & \beta_{2N} \\ \dots & \dots & \dots & \dots \\ \beta_{N1} & \beta_{N2} & \dots & 0 \end{bmatrix}, \sigma = \begin{bmatrix} 0 & \cdots & \sigma_{1N} \\ \vdots & \ddots & \vdots \\ \sigma_{N1} & \cdots & 0 \end{bmatrix}$$

J denotes number of travel dates, T is number of booking dates, N denotes numbers of competitors. In here, B is the price influence matrix that describes how competition between hotels is determined. Theoretically, the price influence matrix can be obtained by 2SPS or 2SRI regression. Moreover, the coefficients should be unbiased. In practice, high dimensionality would cause OLS inverse matrix huge burden to compute. Also, it is highly likely that matrix B and matrix σ are sparse matrix in high dimensional case. Therefore, LASSO would be a better choice than straightforward OLS in this case.

3.3.2 LASSO and Tuning Parameters

The cost of computing the whole coefficient matrix is enormous with complexity N^2. Luckily, hotel price decision-makers would not really response to all competitors in the market but choose a subset as competitor set. It has been discussed in previous literature like Lederman et al.(2014), Li and Netessine (2012) and is commonly believed that pricing strategy is often respond to a few targeting competitors instead of all of them.

3.3.2.1 LASSO

Tibshirani (1996) is first to claim to add regularization term in executing model selection. Lasso algorithm regularizes the linear model with high dimensional variables by shrinking model coefficients to zero. Lasso adds l-1 norm as penalty function in least square optimization formula, form as Equation (3-6):

$$\min_{\beta}\frac{1}{N}\sum_i(y_i-x_i\beta)^2\ +\ \lambda||\beta\,||_1 \qquad (3\text{-}6)$$

Equation (3-6) is sum of squared errors plus penalty term of all coefficients excluding intercept. The penalty term is proportional to sum of absolute values of coefficients in the first term. Parameter λ is a tuning parameter or called hyperparameter, which controls the weight of penalty term and affects equation (3-6) result. Cross-validation method can be used to find optimal λ that performs best. Cross-validation divides data into m partitions, where model is trained on m -1 subsamples and uses the left untouched subsample as test data to determine model's out-of-sample performance. This process repeats m times and the average of m times out-of-sample mean squared error is regarded as the model's out-of-sample performance. The model hyperparameter λ value is determined by λ_{CV} with the best out-of-sample performance.

For each hotel, we solve minimization of MSE, with adding a term of penalization constrain. For 2SRI model, $|\beta_{in}||\sigma_{in}|$ share the same hyperparameter λ. The LASSO would be like:

2SPS:

$$\min_{\alpha_i,\beta_{i,-i},\gamma_i,\delta_i}\frac{1}{JT}\sum_{j,t=1}^{J,\ T}(P_{ijt}-\alpha_i-\beta_{i,-i}\hat{P}_{-i,jt}-\gamma_iX_{jt})+\lambda\sum_{n=1}^{N}(|\beta_{in}|)$$

2SRI:

$$\min_{\alpha_i,\beta_{i,-i},\gamma_i,\sigma_i,\delta_i}\frac{1}{JT}\sum_{j,t=1}^{J,\ T}(P_{ijt}-\alpha_i-\beta_{i,-i}P_{-i,jt}-\gamma_iX_{jt}-\sigma_{i,-i}e_{i,-i})+\lambda\sum_{n=1}^{N}(|\beta_{in}|+|\sigma_{in}|)$$

For model selection in high dimension, the explanation power of instrument variable could be weakened in 2SPS compared to 2RI since predicted value insertion in second stage only contains partial variation. On the other hand, 2SRI approach including original prices can cover the whole price variation in the system and instrument variable should not be weak explanatory power anymore.

3.3.2.2 Adaptive LASSO (A-LASSO)

Though LASSO could shrink part of coefficients towards zero and get a sparse result. Sometimes the amount of non-zero coefficient is still too large than desired, Zou (2006) proposed to add a weighted operator $\mathrm{w_j} = \frac{1}{|\hat{\beta}_i^{ini}|}$ to control the amount of non-zero coefficients. $\hat{\beta}_i^{ini}$ usually use coefficients obtained from ridge regression. Ridge regression is similar to LASSO, with the only change is penalty function is from l-1 norm to l-2 norm. Like equation (3-7):

$$\min_{\beta} \frac{1}{N} \sum\nolimits_{i=1} (y_i - x_i\beta)^2 \; + \; \lambda||\beta||^2 \qquad (3\text{-}7)$$

After getting solution coefficients $\hat{\beta}_i^{ini}$ from ridge regression, the minimization form becomes:

2SPS:

$$\min_{\mathrm{\alpha_i},\beta_{i,-i},\gamma_i,\delta_i} \frac{1}{JT} \sum_{j,t=1}^{J,\;\;\mathrm{T}} (P_{ijt} - \alpha_i - \beta_{i,-i}\hat{P}_{\;-i,jt} - \gamma_i X_{jt}) + \lambda \sum_{l=1}^{N} (\frac{|\beta_{il}|}{|\beta_{il}^{ini}|}|)$$

2SRI:

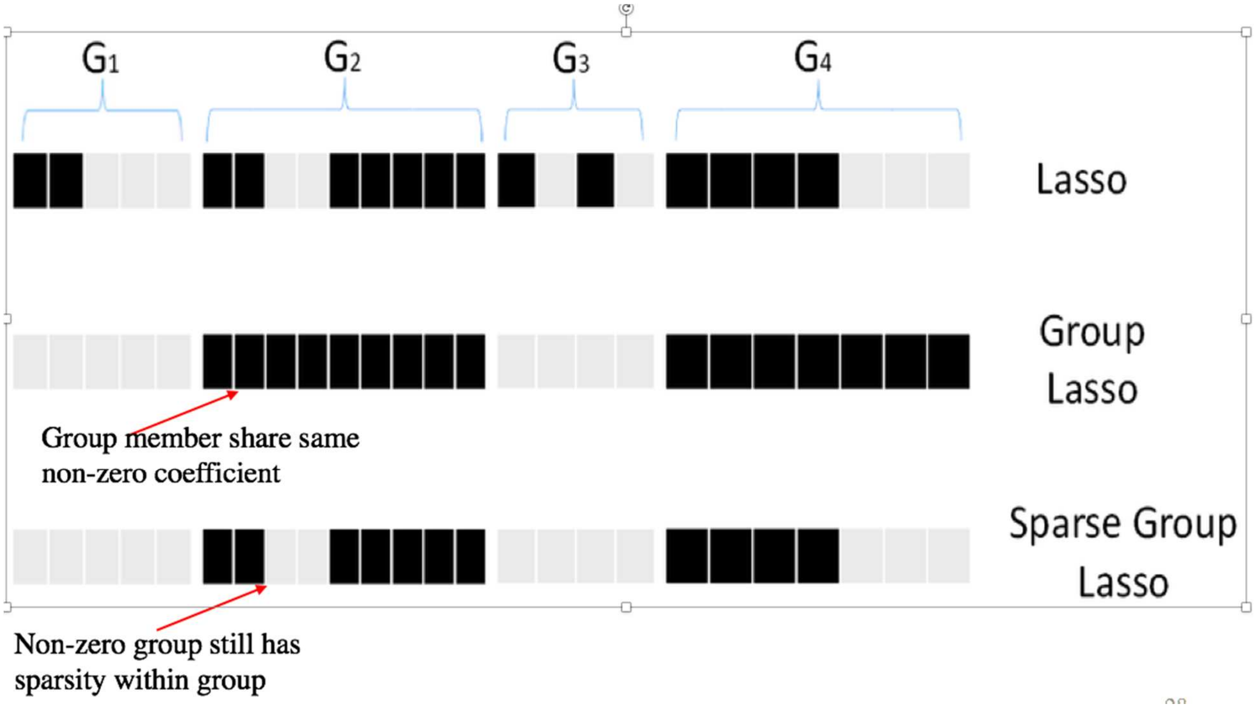
$$\min_{\mathrm{\alpha_i},\beta_{i,-i},\gamma_i,\sigma,\delta} \frac{1}{JT} \sum_{j,t=1}^{J,\;\;\mathrm{T}} (P_{ijt} - \alpha_i - \beta_{i,-i}P_{-i,jt} - \gamma_i X_{jt} - \mathrm{\sigma}_{i,-i}e_{i,-i})$$

$$+\lambda \sum_{n=1}^{N}\left(\frac{|\beta_{il}|}{|\beta_{il}^{ini}|}+\frac{|\sigma_{il}|}{|\sigma_{il}^{ini}|}\right) \qquad (3\text{-}8)$$

Zou(2006) has shown that adaptive Lasso yields consistent estimates of parameters and has the oracle property: at least as fast convergence rate as LASSO. In this paper, we will compare the performance of LASSO and Adaptive LASSO in section 3.5.

3.3.2.3 Group LASSO

In many regression problems, researchers are interested in finding important explanatory factors which may share some common characteristics and can be grouped as input variable. In the hotel competition problem, star ratings and locations are most commonly used factors to segment the hotels. After identifying competitor hotels from the individual level, researchers have more interest in whether these competitors have a pattern that helps us better understand the competition. In this case, cluster similar hotels in a group and shrink group coefficient to zero might help find the clearer pattern. Group lasso (Yuan and Lin 2006) is good way to achieve the goal. Suppose N variables can be clustered in L groups, with the n_l is the number of variables in group l. Simon et al. (2010) indicated that group LASSO's sparse effect is not ideal since the model has limitation that variables share the same coefficients in the same group even for non-zero coefficient group. They proposed a method that sparsity can extend to non-zero coefficient group which makes non-zero coefficient group has members with zero coefficient too. Figure 3.1 illustrates the relationship between LASSO, group LASSO and sparse-group LASSO.



Group member share same non-zero coefficient

Non-zero group still has sparsity within group

Figure 3.9 LASSO, Group LASSO, Sparse Group LASSO Illustration.

Dark color represents nonzero coefficients, white color represents zero coefficient in regression result.

The optimized function form for sparse group LASSO would be like equation (9):

2SPS:

$$\min_{\alpha_i,\beta_{i,-i},\gamma_i} \frac{1}{JT} \sum_{j,t=1}^{J,\ T} (P_{ijt} - \alpha_i - \beta_{i,-i}\hat{P}_{-i,jt} - \gamma_i X_{jt}) + (1-\alpha)\lambda \sum_{l=1}^{N} \sqrt{n_l}|\beta_{il}| + \alpha\lambda|\beta|_1$$

2SRI:

$$\min_{\alpha_i,\beta_{i,-i},\gamma_i,\sigma} \frac{1}{JT} \sum_{j,t=1}^{J,\ T} (P_{ijt} - \alpha_i - \beta_{i,-i}P_{-i,jt} - \gamma_i X_{jt} - \sigma_{i,-i}e_{i,-i})$$

$$+(1-\alpha)\lambda \textstyle\sum_{n=1}^{N} \sqrt{n_l}(|\beta_{il}| + |\sigma_{in}|) + \alpha\lambda(|\beta|_1) \qquad (3\text{-}9)$$

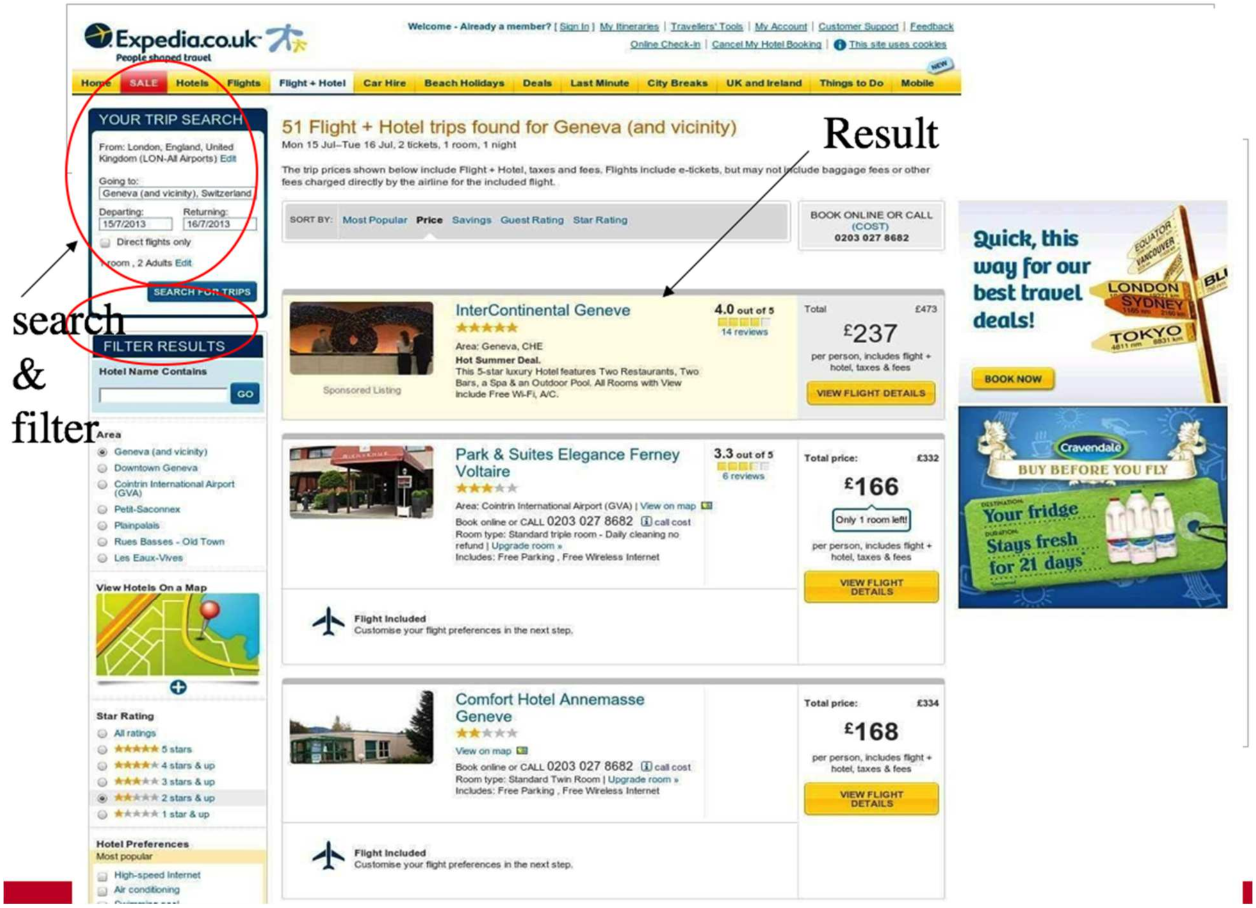
In this paper, we will use sparse group LASSO to compare the price competition impact between the two most commonly discussed factors in hotel industry: location and quality.

3.4 Data

3.4.1 Data Exploration

The customer search dataset is provided by Wharton Customer Analytics Initiative (WCAI) and Expedia traveling website. In this dataset, we can observe demand shock from the online platform: the hotel choices offered to traveler when customers search with filters, which page the hotel stands in the search, and click through action to record customers actual booking. In previous literature, the demand shock is only observable to hotel managers but not available to researchers. Researchers have no aware of the choice set that customers are exposed to. Assuming positive booking probability to each hotel using random utility approach is a popular choice in previous research, but it clearly cannot correctly describe customers' choice set in the hotel industry. From the search data, we can observe customers level choice sets that they face when making a search action. Then demand shock for each hotel can be obtained by aggregating choice sets of all users.

Dataset also includes basic information for 322 hotels located in Manhattan area. We focus on 229 of hotels that have records of star rating and room capacity as hotel's basic characteristic for the system of competitors. Among the 229 hotels, 41% of hotels are chain branded hotels, and they have average 3.3 stars rating as service quality. 59% of independent hotels has an average 3.0 stars rating which is slightly lower than brand hotel.



YOUR TRIP SEARCH

From: London, England, United Kingdom (LON-All Airports) Edit

Going to: Geneva (and vicinity), Switzerland

Departing: 15/7/2013 Returning: 16/7/2013

□ Direct flights only

1 room , 2 Adults Edit

FILTER RESULTS

Hotel Name Contains

Area
- ● Geneva (and vicinity)
- ○ Downtown Geneva
- ○ Cointrin International Airport (GVA)
- ○ Petit-Saconnex
- ○ Plainpalais
- ○ Rues Basses - Old Town
- ○ Les Eaux-Vives

View Hotels On a Map

Star Rating
- ○ All ratings
- ○ ★★★★★ 5 stars
- ○ ★★★★ 4 stars & up
- ○ ★★★ 3 stars & up
- ● ★★ 2 stars & up
- ○ ★ 1 star & up

Hotel Preferences
Most popular
- □ High-speed Internet
- □ Air conditioning

51 Flight + Hotel trips found for Geneva (and vicinity)

Mon 15 Jul–Tue 16 Jul, 2 tickets, 1 room, 1 night

The trip prices shown below include Flight + Hotel, taxes and fees. Flights include e-tickets, but may not include baggage fees or other fees charged directly by the airline for the included flight.

BOOK ONLINE OR CALL (COST) 0203 027 8682

InterContinental Geneve ★★★★★
4.0 out of 5 — 14 reviews
Area: Geneva, CHE
Hot Summer Deal.
This 5-star luxury Hotel features Two Restaurants, Two Bars, a Spa & an Outdoor Pool. All Rooms with View include Free Wi-Fi, A/C.
Sponsored Listing
Total £473 — £237 per person, includes flight + hotel, taxes & fees

Park & Suites Elegance Ferney Voltaire ★★★
3.3 out of 5 — 6 reviews
Area: Cointrin International Airport (GVA) | View on map
Book online or CALL 0203 027 8682 call cost
Room type: Standard triple room - Daily cleaning no refund | Upgrade room »
Includes: Free Parking , Free Wireless Internet
Total price: £332 — £166 Only 1 room left! per person, includes flight + hotel, taxes & fees
Flight Included
Customise your flight preferences in the next step.

Comfort Hotel Annemasse Geneve ★★
View on map
Book online or CALL 0203 027 8682 call cost
Room type: Standard Twin Room | Upgrade room »
Includes: Free Parking , Free Wireless Internet
Total price: £334 — £168 per person, includes flight + hotel, taxes & fees
Flight Included
Customise your flight preferences in the next step.

Figure 3.10 Illustration of Search Page in Expedia.com

3.4.2 Summary of Search

Customer search data collected 4778 cookie-based users searched 15,000 times and 1,546,296 lines of search results during period October 1st- October 16th 2009 on hotels located in Manhattan. Dataset records travel destination and travel dates for each search query, filters and sorting criterion customers have used. A standard hotel search query also reveal customers' basic information, like number of rooms which can identify demands for each search. A search query will filter 100- 200 hotels fit the criterion, but the website only shows maximum of 25 hotels in one page. Figure 3.2 shows what customer shall see in result page after input search query with travel date, travel destination, room need with filter and sorting order. The search data we get recovered what customer observed in a webpage using database format.

From the search statistics, about half of the searches used default search and did not use any advanced filter. 25% of customers view detail of the hotel page, 6% of customers reach to price page, and 1% of customers book the hotel in the end. Among all the searches, 38% of customers at least filter or re-sort the result by distance or star rating once which is collaborated with Lederman et al.(2014) finding.

After summarizing the search data, we decide to drop some missing values and outliers that not include meaningful information. It includes: no specific check-in date or check-out date; search 3 months before check in; stay in one hotel for more than 1 month; search more than 50 pages. Applying these criteria leads to a finalized sample of 3,954 users and over 9,000 times search.

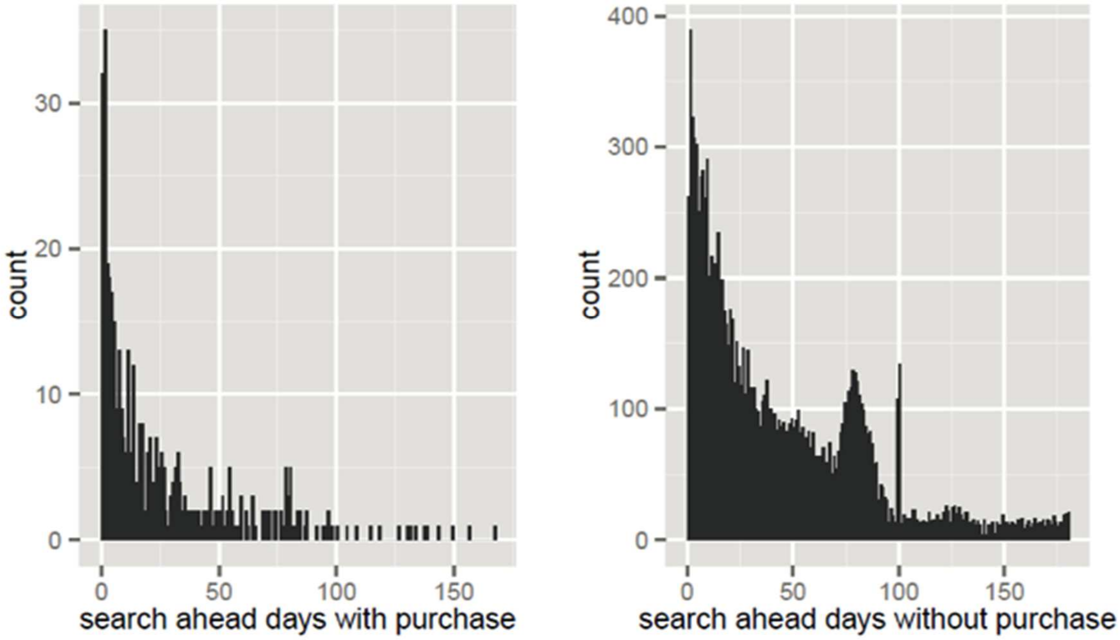


Figure 3.11Illustration of Search Page in Expedia.com

The Right graph has a peak in the 100 days. It indicates there is a large proportion of customers who search bookings 3 months ahead of traveling in the without purchase group, which indicates that non-purchasers are more casual about what they search.

Figure 3.3 shows that most customers who start to search the hotel long time before travel date tend not to make the purchase, indicating they are not serious in booking. On the other hand, frequency shows that a large proportion of users tend to book hotels within three weeks before their trip. For business purpose visit, customers sometimes decide to take the trip in the very last moment. Moreover, this type of booking probably only takes hours to make a decision, and it often happens within one week before they travel.

3.4.3 Daily Price

In the Expedia customer search dataset, we cannot directly observe hotel's daily pricing strategy. The price we observed is exactly the webpage customers see in the search website, lowest average price per night. From all of the observed lowest average price data, mean price for all hotels in Manhattan is \$281, and standard deviation is \$190, which indicates a wide range of price between the highest and lowest.

In equation (3-2), we denote P_{ijt} as hotel i's lowest average price for travel date j when book on date t. Days of booking advanced to travel is automatically determined when travel date and book date are known. Hotels have strategies to set different prices for weekday and weekend to distinguish customers from tourism purpose and business purpose. Thus, travel date of the week could be a critical attribute to explain price variation. On the other hand, in many industries like hotel and concert, prices tend to be higher for booking several days before travel than several weeks before. It is likely another crucial factor to affect the price.

For the hotel price, there is another issue that single night price is not observable for some hotels. Instead, we observe the average price of multiple travel days. Some data cleaning rule is needed to obtain a reasonable estimate of single night price for each hotel. First, for each travel date tuple (travel date j, booking date t), only keep travel date within a month (30 days) from booking date. A shown in Figure 3.3, 80% of hotel searches have search advance to travel within 30 days. For searches advance to travel more than 30 days, the data point is too sparse to make inference on competition effect. Second, we will use single night stay price observation as reference for price $\mathrm{P_{ijt}}$. It is likely that multiple observations with different prices are found for specific single night stays. In this case, we will use average price of observations to estimate $\mathrm{P_{ijt}}$. Moreover, there are scenarios that for specific travel date tuple, there is no single night stay price observation. However, multiple-night-stays overlap the travel date which can be used as a proxy. In this case, the average daily price of multiple-night-stay will be regarded as a single night price. After applying these data cleaning rules, we remove the hotels which have half of the prices are unobserved in travel date tuple (travel date, booking date). These hotels are usually no star or 1-star hotel. Moreover, we remove the travel date tuple with more than half of hotels' prices are missing, which are unlikely to use imputation to make reliable inference on this date. After processing the data, there are 162 available hotels left in the dataset.

For the rest of traveling tuple with missing price data and missing data is less than 50% of them, it is not wise to delete all of them since nearly half of hotels have at least one travel date tuple missing, which is unrealistic to delete them all. We decide to fill in predicted price by 2SPS as imputation for missing price in the matrix. From equation (3-3):

$$\mathrm{P_{ijt}} = \upalpha_{\mathrm{i}} + \gamma_i X_{jt} + \delta_i U_{ijt} + \epsilon_{\mathrm{ijt}} \tag{3-3}$$

Coefficients $\alpha_i, \gamma_i, \delta_i$ are all not travel date tuple specific. It is feasible to obtain coefficient with real observations, and we have real observations for X_{jt}, U_{ijt} to get predicted value $\widehat{P}_{ijt}$ that can be imputed into price matrix. The result should not be affected since 2SPS should have similar performance as 2SRI in most cases.

3.5 Result

Before running model with the real data, it is the best interest for this paper to compare the performance among all models listed in section 3.3 and make a reasonable model selection among LASSO and Adaptive LASSO, 2SPS and 2SRI. As many previous literatures justified in their model (Terza et al,(2008), Li et al.(2012), Zou (2006), Yuan and Lin (2006)). We use generic data for simulation analysis first.

3.5.1 Simulation

We simulate a virtual Manhattan competition with 150 hotels as competitors in the market. Since simulation focuses on selecting models that can recover competition actual pattern, we will drop variables X_{jt}, α_i in simulation and only focus on price in here.

$$p = BP + \epsilon \tag{3-9}$$

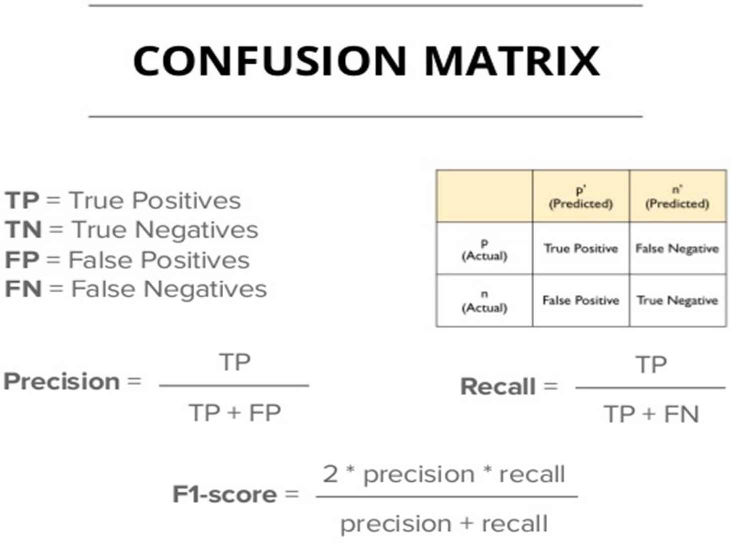
$$\epsilon = \gamma U + e \tag{3-10}$$

B is simulation price coefficient matrix. Generate coefficient matrix B with dimension 150*150, and diagonal line is 0. In each row of matrix B, choose 15 of them as non-zero to generate coefficient β_{ij} that follow normal distribution with N(0.5,0.1) and keep all other elements in the row equal to zero. ϵ is denoted as unobserved error term that potentially correlate across error. U represents the instrument variable and e is unobserved error term. In the first stage of two stage least square, we will have

$$p = \delta U + v \tag{3-11}$$

where use OLS that get predicted $\hat{p}$ and v for 2SPS and 2SRI method. γ in simulation set to 1. And both U and e will be set to randomly normal distribution from N(0,1).

We use F1-score as metric to evaluate which model performs the best in recovering the simulation result. Figure 3.4 gives an illustration of confusion matrix and concept for precision, recall and F1-score. In this specific price simulation, if I run a model A and get competition matrix B with a number of non-zero coefficients N, precision rate means proportion of number of correctly identified non-zero coefficient divided by N. Recall rate means the proportion of number of correctly identified non-zero coefficient divided by number of actual non-zero coefficients in this simulation. F1-score is harmonic mean of precision rate and recall rate, which is widely used to measure the performance of model.



CONFUSION MATRIX

TP = True Positives
TN = True Negatives
FP = False Positives
FN = False Negatives

	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

$$\textbf{Precision} = \frac{TP}{TP + FP} \qquad \textbf{Recall} = \frac{TP}{TP + FN}$$

$$\textbf{F1-score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Figure 3.12Confusion Matrix for Classification

Table 3.1 column illustrates the comparison results among models. The first column represents variable selection approaches from left to right are LASSO CV, LASSO CV 1se, BIC and Adaptive LASSO. LASSO CV is the minimum cross-validation result mentioned in Section 3.2. LASSO CV 1se chooses the result of minimum cross validation plus 1 standard error. CV 1se is based on rule of thumb of statistical literature, usually can get more sparser result than LASSO CV. BIC is commonly used in time series for variable selection and penalize model coefficients heavier than AIC and other models.

Table 3.11. Variable Selection F1-Score in Simulation

	CV	**CV 1se**	**BIC**	**A-LASSO**
2SPS				
# variables	72	31	11	19
Precision	14.4%	28.5%	85.1%	37.2%
Recall	91.5%	69.0%	12.3%	81.4%
F1-score	0.25	0.40	0.21	0.51
2SRI				
# variables	51	25	12	21
Precision	28.5%	41%	68%	61%
Recall	98.5%	99.5%	86.4%	97.7%
F1-score	0.44	0.58	0.76	0.75

From the row comparison, Table 3.1 shows that the recall rate is usually higher than the precision rate in data drive model LASSO and Adaptive LASSO. The precision rate is low since the models pick much more variables than actual simulation. Among all 4 variable selection approaches, Adaptive LASSO has best combination of precision and recall in both 2SPS and 2RI case, which indicates by highest F-1 score. From the vertical comparison, 2SRI model in all columns has better F-1 score than 2SPS model. Table 1 results suggest 2SRI and Adaptive LASSO model have the best performance in recovering simulation result. Thus, in the later real data regression, we will mostly use adaptive LASSO with 2SRI for model interpretation.

Another thing that is worthy of mentioning is that we did not use group LASSO for this simulation data since group LASSO needs cluster variables into different group by common features in the group. It is difficult to simulate the group LASSO in this case without any prior information about variables. I will show the comparison of sparse-group LASSO result with other models in the real dataset.

3.5.2 Instrument Variable

A valid instrument variable should satisfy two conditions: 1) instrumental variable is uncorrelated with unobserved error 2) the instrument variable should be correlated to the endogenous variable. It is difficult to find a valid instrument variable that fit both conditions. In customers' search data, there are two types of observable demand shock in our data, displays and clicks. In most cases, display represents customers are exposed to the hotel in webpage before clicked in. This filter and sorting come from exogenous demands need, like geographic preference and star rating. Price can be a filter or sort criterion for searches in the dataset. We exclude these types of searches since these demand shocks are directly correlated with prices. In the dataset, there are 8 different types of search sorting (sort by hotel name, sort by city name, sort by star rating, sort by distance, sort by star rating descending, sort by airport code, sort by price, sort by traveler reviews). Except for sorting by price, other sorting criteria are valid without price involved. When a hotel is displayed to a user multiple times, only one display is counted since there is only one demand. On the other hand, clicks are not ideal instrument variables. As what search results example in Figure 1, customers can observe the price information for a specific hotel and then decide to click afterward. Clicks are likely correlated variable with prices and are not valid for instrument variable. The more detail about the validity of instruments is shown in Appendix B.

After deciding the valid instrument variable, there is still undecided variable like how many days of accumulated display should be chosen as instrument variable. In here, we have tried 3 different days:

1. times of display in past one day $\mathrm{U}_{\mathrm{ij,t-1}}$: measures number of hotel displayed by hotel i at travel date j for booking date t-1.

2. times of display in past three days $\sum_{t-3}^{t-1} U_{ijt}$: measures number of hotel displayed by hotel i at travel date j for booking date from t-3 to t-1.

3. times of display in past seven days $\sum_{t-7}^{t-1} U_{ijt}$: measures number of hotel displayed by hotel i at travel date j for booking date from t-7 to t-1. Times of display in past 7 days cancels the week effect, which might potential affect the result. The only concern in here is dataset has only 16 days of search from customers which makes past week display is expensive to use. The observations are fewer in this case.

The concern about instrument variables is that they might be too weak to explain price variation in the equations. Thus, we compare the increasing explanatory power (R-squared) when adding instrument variable in the equations with all 3 candidates (past 1 day display, past 3 day display, past 7 days display). Result is shown in Table 3.2.

Table 3.2 shows that adjusted R-square increase when 3 different types of instrument variable are included in the first stage of equations (3). Column 2 represents mean of R-squared increase, and column 3-5 represents 25 percentile, median and 75 percentile of R-squared increase. Times of display in past 7 days have the most substantial R-squared increment among all 3 types of instrument variables. It can explain additional 6.1% of price variation, which is higher than the other 2.

Table 3.12. R-Squared Addition When Instrument Included

	Increase Explanatory Power			
Instrument	mean	25 percentile	median	75 percentile
Past 1 day display	2.3%	0.3%	0.6%	2.8%
Past 3 days display	3.9%	1.0%	2.8%	5.2%
Past 7 days display	6.1%	2.1%	3.9%	8.4%

3.5.3 Travel Characteristics Impact on Price Variation

In the previous section 3.4.3, we have discussed that travel date of week and days of booking ahead of traveling are two possible factors that explain daily price variation. Table 3.3 is a good demonstration of how travel date characteristics X_{jt} affect the hotel's pricing strategy.

In Table 3.3, three different star rating hotels are listed for comparison: Econo Lodge Time Square (2 stars, brand of Choice), Four Point by Sheraton (3 stars, brand of Starwood), Trump International Hotel (5 stars, independent). Three hotels listed in Table 3.3 are from different branded, with different stars rating, but all located in Midtown of Manhattan. Table 3.3 indicates that travel dates have a significant weekday-weekend effect on hotel pricing. The weekday-weekend effect is shown as opposite in 2 stars hotel and 5 stars hotel. Low quality hotel (2 stars) charge slightly higher prices on weekend than weekday. However, high quality hotel (5 stars) charge higher prices on weekday than weekend. This observation might be explained as 5 stars rating hotels have a large proportion of customers that are business travelers who book the hotel more often on weekday than weekend. On the other hand, 2 stars rating hotels accommodate more customers for tourism who often travel on weekend. For 3 stars rating hotel, the weekday-weekend effect is mixed and the price in weekend (Friday, Saturday) is significantly higher than Sunday.

Table 3.13. Travel Date Characteristic Coefficient on Price

	2 stars rating	3 stars rating	5 stars rating
Travel Date			
Monday	5.75 (5.46)	6.67 (4.33)	123.84*** (23.58)
Tuesday	-15.67*** (5.71)	16.92*** (4.55)	182.55*** (23.19)
Wednesday	-15.19*** (5.70)	7.58 (4.50)	178.97*** (23.09)
Thursday	25.93*** (5.61)	-0.15 (4.53)	128.04*** (23.05)
Friday	32.81*** (5.46)	14.49*** (4.26)	-51.55** (23.1)
Saturday	33.25*** (5.89)	16.78*** (4.41)	-102.55*** (23.7)
Sunday	Baseline	--	--
Days of advanced booking			
1-2 days	-63.62*** (7.26)	-63.81*** (9.56)	65.53*** (27.89)
3-7 days	-41.68*** (5.39)	-40.80*** (6.12)	86.51*** (21.76)
8-14 days	-5.54 (4.25)	2.12 (3.28)	90.28*** (17.81)
15-21 days	8.19** (4.12)	4.84 (3.19)	88.83*** (18.24)
22-30 days	Baseline	--	--

For days of advanced booking, it is believed the hotel room price is not linear (Li et al. (2016)) with days of advanced booking. Daily prices usually keep higher when booking date is close to travel date, but the prices drop when the book date is 1 or 2 days from travel date. To investigate their relationship, I split the days of advanced booking into groups 1 -2 days, 3-7 days, 8-14 days, 15 -21 days, 22 -28 days. Row 10-15 in Table 3 shows the relationship between prices and advanced booking. The result also suggests a very distinct pricing strategy between 2 stars economic hotels to 5 stars high quality hotels.

Table 3.3 shows, for 5 stars rating hotel, the price is always significantly higher for advanced booking within 3 weeks compared to the baseline price, advanced booking beyond 3 weeks. It also suggests that price for days of advanced booking within 2 days is second lowest in the column. For 2 stars and 3 stars rating hotels, prices are lower when booking date is within one week to travel date. Moreover, the price is consistent and not significantly change when for booking date to travel date is longer than a week.

3.5.4 Competition Result

So far, the results only use information from the first stage of 2SLS. In Table 3.4, we show the results of second stage regression. For second stage regression, there are many variable selection methods can be applied. As previously discussed in section 3.5.1, Adaptive LASSO with 2SRI recovered coefficients of the simulation data better than other models. In here, I will only use this method to run the second stage regression.

The second column in Table 3.4 is the average number of non-zero coefficients for different star rating hotel. As shown, 3 stars and 4 stars hotels have the most competitors that affect their pricing strategy. Moreover, we try to analyze a very prevalent question in hotel industry: quality and locations, which type of competitors affect hotel pricing strategy most? Thus column 3 and column 4 compare the components of each hotel’s competitor set. Column 3 shows the proportion of competitors belong to the same star rating, and column 4 shows the proportion of competitors belong to the same submarket region. Results show hotels are influenced more by competitors with the same quality than competitor within the same geographical submarket. The possible reason behind this is Manhattan is a small area that hotels are not far away from each other. It takes less than 50 minutes from north end to south end of Manhattan by subway. Thus the location competition is less fierce than star rating competition.

Table 3.14. Hotels with Non-Zero Coefficients in the Second Stage

Star Rating	Average # of competitors	Same star ratings	Same submarket
1	5.2	39%	1%
2	10.1	52%	15%
3	13.4	49%	27%
4	10.8	51%	35%
5	7.2	17%	16%

3.5.5 Cluster Competition

Section 3.5.4 shows the competition pattern for individual hotel level. The result is significant for a hotel revenue manager to identities correct competitor set and makes the right adjustment. From research perspective, it is more interesting that if it is true that the same star rating hotels have a larger impact on pricing strategy than the same submarket region. Thus group LASSO model can be brought to test whether the finding in Section 3.5.4 is valid.

We cluster hotels into groups by two factors: star rating and submarket region. It has 5 different star rating and 8 different submarket region which makes 40 groups. Hotels in each group share the same star rating and the same region. We use sparse group LASSO discussed in Section 3.3.2.3 to construct the model. Compared to Table 3.4, Table 3.5 has fewer number of average competitors for each hotel since sparse group LASSO is easy to obtain a sparser model than adaptive LASSO. It has a similar competition pattern that star rating effect weights more than regional effect for influencing hotel's pricing strategy.

Table 3.15. Hotels with Non-Zero Coefficients in Sparse Group LASSO

Star Rating	Average # of competitors	Same star ratings	Same submarket
1	1.3	69%	50%
2	7.4	36%	21%
3	13.2	39%	27%
4	8.6	33%	29%
5	7.2	17%	15%

3.6 Conclusion

This paper tries to understand competition pattern across firms with a large number of competitors. Simultaneous equations need valid instrumental variables that correct the endogenous problem and estimate the causality effect in prices. Moreover, high dimensionality makes the variable selection is challenging. This paper proposes an idea to use adaptive LASSO and group LASSO to solve the system of equations in the high dimensional situation.

Using customer search data, we find that online past seven day displays are a valid and robust instrument to represent price variation in the hotel market. Travel date of week and days of advanced booking both have significant changes in prices, but the effect varies by different hotel star ratings. Competition pattern shows that hotel pricing strategy is more affected by peers who have the same star rating than peers located in the same region. It is a valuable result that can help hotel revenue manager to correctly find its competition set and make responses to its price change. It is also a valuable finding for researchers to find star rating quality has a larger impact than the location in hotel's competition pattern.

Practically, this analysis can be applied to other highly competition areas like online shopping. Moreover, this methodology is also a good complementary for researcher use customer demand to estimate price competition. It is interesting to check if customer demand approach and firm demand approach will get a consistent result in analyzing the competition.

References

Athey, Susan, and Guido W. Imbens. 2015 "Machine learning methods for estimating heterogeneous causal effects*." Stat* 1050 5.

Bajari, P., Nekipelov, D., Ryan, S.P. and Yang, M., 2015. Machine learning methods for demand estimation. *The American Economic Review*, 105(5), pp.481-485.

Belloni A, Chen D, Chernozhukov V, et al. Sparse models and methods for optimal instruments with an application to eminent domain[J]. *Econometrica*, 2012, 80(6): 2369-2429.

Belloni A, Chernozhukov V. Least squares after model selection in high-dimensional sparse models[J]. *Bernoulli*, 2013, 19(2): 521-547.

Berry, S., J. Levinsohn, A. Pakes. 1995. Automobile prices in market equilibrium. *Econometrica* 63(4) 841-890.

Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent [J]. Journal of statistical software, 2010, 33(1): 1.

Hausman J A. Specification tests in econometrics [J]. *Econometrica*: Journal of the Econometric Society, 1978: 1251-1271.

James G, Witten D, Hastie T, et al. An introduction to statistical learning [M]. *New York: springer*, 2013.

Koulayev S. Estimating demand in online search markets, with application to hotel bookings [J]. *Federal Reserve Bank of Boston Working Paper*, 2010 (09-16).

Koulayev S. Search for differentiated products: identification and estimation [J]. *The RAND Journal of Economics*, 2014, 45(3): 553-575.

Li J, Netessine S. Who are my competitors?-Let the customer decide[J]. 2012.

Li J, Netessine S, Koulayev S. Price to compete…with Many: How to identify price competition in high dimensional space. *Management Science*, Forthcoming.

Nikulkov, Alex, Kostas Bimpikis, Yonatan Gur. 2015. Dynamic Pricing Under Competition: An Empirical Study of the Hospitality Industry. Working Paper, *Standford Business School*, Palo Alto, CA.

Simon N, Friedman J, Hastie T, et al. A sparse-group lasso[J]. *Journal of Computational and Graphical Statistics*, 2013, 22(2): 231-245.

Tereyagoglu, Necati, Peter Fader, Senthil K. Veeraraghavan. 2015. Multi-attribute loss aversion and reference dependence: Evidence from the performing arts industry. Working paper, *Georgia Institute of Technology*, Atlanta, GA.

Terza J V, Basu A, Rathouz P J. Two-stage residual inclusion estimation: addressing endogeneity in health econometric modeling [J]. *Journal of health economics*, 2008, 27(3): 531-543.

Tibshirani R. Regression shrinkage and selection via the lasso [J]. *Journal of the Royal Statistical Society*. Series B (Methodological), 1996: 267-288.

Varian, Hal R. 2014. "Big data: New tricks for econometrics." *The Journal of Economic Perspectives* 28.2 : 3-27.

Wang, H., R. Li, C. Tsai. 2007. Turning parameter selector for the smoothly clipped absolute deviation method. *Biometrika* 94 553-568.

Wooldridge, J.M. 2010. Econometric analysis of cross section and panel data, chap. 12. *MIT Press, Cambridge, MA*.

Yuan M, Lin Y. Model selection and estimation in regression with grouped variables [J]. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2006, 68(1): 49-67.

Zou, H. 2006. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association* 101(476) 1418-1429.

CHAPTER 4. RE-EXAMINE REAL ESTATE PROBLEM USING NEURAL NETWORK MODELS

4.1 Introduction

E-commerce business has been widely developed and become one of the largest marketplace since 2000. The online platform is an effective way for both sides of buyers and sellers to make transactions. Many online marketplace researchers have a significant interest in algorithmic methods to predict the correct prices and make advices to sellers. It would reduce potential asymmetric information and market inefficiency. In house market, prediction becomes even complicated and vital since there are different sources of factors could affect the prices, tabular, image, text, etc. Among all the factors, words description is one of the most complex issues to handle since it is unstructured and is challenging to propose models correctly represent it. Choosing the appropriate model to manage unstructured text is paramount to house price prediction problem. In this chapter, our goal is to build an advanced and robust model which is capable of extracting valuable information from free text and predicting house prices based on it.

In previous Chapter 2, we collected free text information from Zillow.com and used these features with other house's numerical variable to build models to predict prices. In conventional, this method is called bag-of-words and is popularly used in natural language processing field. Vectors in the model have the same length as the number of words in the collection of free text. One limitation of these type models is that variable selections are still necessary steps in both linear regression model and tree-based models. In both models, subsets of variables are selected to find the optimal solution depending on what algorithm and hyperparameters are used. However, the dropped variables are still likely to contain valuable information deciding the house prices but are not selected by model. Word embedding vector, the base unit in Word2Vec approach, is a smaller dimension vector used to represent the entire collection of free text. The dimension of word embedding vectors is independent of the length of words and is flexible to choose. The difference between this approach and bag-of-words are dropping subset of variables are not necessarily step anymore, and instead of that, lower dimension of embedding vectors is used to represent text information.

In this paper, we use Word2vec to obtain word embedding vectors and apply the vector representation to train two distinct neural network models, specifically Long Short Term Memory(LSTM) and Convolutional Neural Network (CNN). we will show neural network models bring more accurate prediction of house prices compared to models used in Chapter 2. In detail, there are two steps to build these models. In the first step, we use a comprehensive and abundant source of data, Google News, to pre-train our word vector representation and get initial weight for the neural network model. In the second step, we use word vectors from the previous step to train neural network model on Zillow house prices data, and compare the result with linear regression model and tree-based machine learning model used in Chapter 2. The result shows that CNN and LSTM models are better than linear regression and tree-based model in term of the accuracy of predicting house prices. We also show the weight of word vectors along with other variables, numerically represent the feature importance in this architecture, though it is not entirely accurate.

It is a popular topic since there is no previous research paper in economics applying neural network architecture to extract text descriptions and predict prices. While there has been much work on pricing algorithms in the financial market using text information like news data or Twitter data, but financial market and real estate are two quite distinct areas in pricing.

Moreover, Word2Vec approach to extract text information is innovative in economic research ,especially in real estate property topic.

This paper is organized as follows. In Section 4.2, we discuss the background of house price predictions and works of others in this topic. In Section 4.3, we introduce the model of Word2Vec, LSTM and CNN model. Section 4.4 briefs introduce the dataset we use in this paper. Section 4.5 discusses a series of experiments to optimize the hyperparameters in the model and find the best performed model. Section 4.6 is a conclusion of the work with discussion on contribution and further area can be applied.

4.2 Related Literature

Using neural network to analyze economic data was new, and it is rare to find related economics literature that addresses the same problem that we try to figure. We collect previous ideas and works from different topics and areas, including sentiment analysis and deep learning to develop our model. There were four ingredients that we found is related: sentimental analysis, word embedding, recurrent neural network (RNN) and convolutional neural network (CNN).

The first area of study is from sentimental analysis(SA). Though applying free text for pricing is relatively new, online customer reviews for sentiment analysis has been attracted to massive attention from researchers in business and economics area for a long time. With the prevalence of user generated content (UGC) on website like Facebook, Twitter, Reddit, users can collect vast amounts of opinions in various web sites. Many researchers begin to use UGC data to analyze sentiment in reviews. Snyder and Barzilay (2007) evaluated the sentiment of multiple aspects of restaurants like environment or food quality. Kang et al. (2013) used sentimental analysis from Yelp review to estimate restaurant hygiene quality. Gentzkow et al. (2015) used the sentiment from public speeches to analyze political affiliations. The critical difference between our proposed problem and sentiment analysis is response variable y in sentiment is often discrete variable. Generally, there are only three different categories like “positive”, “negative”, “neutral” in sentiment topic. However, price prediction is to solve a continuous variable problem. Sentiment analysis prefers to use logistic as active function, whereas continuous variable problem uses the linear function.

The second area of literature is word embedding. The conventional approach in natural language processing uses bag-of-words model (shown in Chapter 1). It uses vector that has the same length as the number of words in collections of documents. Each component in the vector represents the frequency of words in one document. However, when documents have massive words, bag-of-words approach is too sparse and has difficulty in computing. Word embedding vector uses a smaller size of vector to represent the entire set of words, and each small size vector is condensed and continuous value. Word2Vec (Mikolov et al. 2013), a quite popular word embedding tool in computer science area, used a deep-learning approach to represent vectors, seeking semantic relation between words. Word2Vec can find a group of words having a tight semantic or synthetic relationship with a specific starting seed word. In this paper, we choose Skip-Gram model, one of successful implementation of Word2Vec, to process text data. The detail of Skip-Gram is introduced in Chapter 3 and Appendix C.

The third area of literature is one of the most famous architecture in deep learning, Recurrent Neural Network(RNN). Mikolov et al. (2010) showed that RNN model outperforms conventional n-gram bag-of-words approach in natural language processing area. The bright side of RNN modeling is keeping previous state information to compute its next state, where the idea is similar to time series in economics. Nevertheless, traditional RNN model suffers a problem in conveying information in a longer period sequence. Long Short Term Memory (LSTM), a revised RNN model with long term memory part, is believed as an advanced architecture that solves this issue. Wang (2015) proposed to use LSTM architecture with pre-trained Word2Vec (Mikolov et al.2013) to analyze Twitter sentiment analysis, and their performance was better than the traditional RNN models. In this paper, we follow this similar approach to construct our LSTM model to model house unstructured text and predict the prices.

The fourth area of literature is Convolutional Neural Network (CNN). CNN has been proved as an effective approach in solving vision problem. CNN contains two different types of layers: the convolution layers and the pooling layers. In general, CNN model was limited by computing resources and did not perform well until recently. Thanks to the increasing GPU computing power, training deep convolutional neural network becomes feasible and less time consuming. In recent years, CNN model also has been proposed to solve natural language processing problem. Kalchbrenner et al. (2014) studied CNN model to analyze semantic modeling of sentences, and their model had slightly better performance as well as much less computing time than RNN model. The reason behind this is CNN architecture can utilize parallel computing to accelerate its speed, but LSTM can only run its model in sequence order. In this paper, we construct a CNN model following their approach and compare the performance with LSTM model.

4.3 Methodology

In this section, we introduce the methods that have been used for this paper, including Word2Vec, Long Short Term Memory (LSTM) and Convolutional Neural Network (CNN) models. Word2Vec is an alternative approach to bag-of-words, using low dimension vectors to represent a large number of words in the dictionary. LSTM and CNN are two distinct neural network models that widely used in processing text data. The computation of the last layer output (activate function) uses a liner function to compute output. Thus, we can make some comparison between regression models and neural networks.

4.3.1 Word2Vec

Word2Vec is a dimension reduction method proposed by Mikolov et. al. (2013). It is open source and has been applied in multiple open source packages in Python. In this work, we used a gensim package provided by Stanford to implement it. Word2Vec creates vectors without human selection or intervention. Feeding enough data, Word2Vec can make a highly accurate guess about word semantic meaning in the context. Another significant advantage of Word2Vec is it runs fast in massive dataset and result can be widely applied as initial weight in other text data source, which is called transfer learning. Transfer learning means learning semantic relation between words in one dataset should be able to transfer and help understand another text data since semantic meaning among words should be the same in most cases. In this paper, we use pre-trained word vectors in Google News dataset (around 100 billion words). Google News dataset already has available computing result trained by other researchers and can be directly downloaded online.

The text information in Google News contains massive words, and Word2Vec approach can benefit to understand the semantic relation of words used in Zillow.com dataset. The semantic relation among words usually can be accurately extracted from millions or above of data. For example, the top frequently words in the dataset are adjective words either positive or negative. Word2Vec average vectors for these words and return the closest neighbors (i.e.words) to the representing vectors. Table 1 illustrates the computed words distance between adjective words. In Figure 1, the graph (Ouyang et al. 2015) shows the relationship of words in the dataset. The nodes in the graph represent the words in the pre-trained dataset and the edges represent the strength of word similarity. It is a clear illustration that synonyms are clustered in a closer distance. The words such as “good”, “nice”, “great” have edges connection which represents strong semantic similarity. The Word2Vec approach represents the similarity of the words in a precise manner, which will make a further prediction model more accurate.

Another thing that needs to mention is the implementations of Word2Vec approaches. We found literature that argued for training Word2Vec on GloVe (Pennington and Socher 2014) rather than Skip-Gram with negative sampling (SGNS). However, there is also contradictory conclusion from other research that SGNS has better performance than GloVe. Since this paper follows the Mikolov et al.(2013) paper, Skip-Gram model is the approach I choose to pre-train the word embedding vectors. The detail of this method will be introduced in the Appendix C.

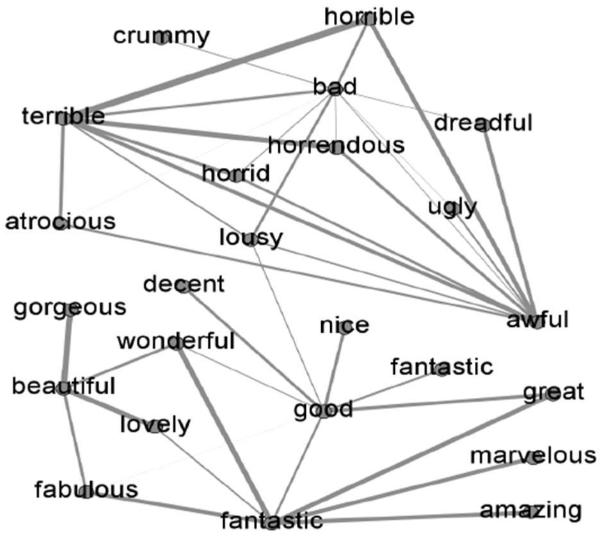


Figure 4.13 Network represents the bond between words.

The node represents the words in the dataset. The edge thickness represents the strength of word similarity

Table 4.16. example of output ./distance GoogleNewsvectors-negative300.bin

The number in the table represents the strength of word similarity. The larger value means the stronger similarity. The table was initially generated by Ouyang et al.2015, and I use pre-trained weights to recalculate it.

good	decent 0.664	excellent 0.691	fantastic 0.523	great 0.712	Nice 0.724
bad	crummy 0.518	horrible 0.606	horrid 0.559	lousy 0.646	terrible 0.626
awful	dreadful 0.711	horrendous 0.673	horrible 0.796	horrid 0.621	terrible 0.776
beautiful	fabulous 0.608	gorgeous 0.841	loveliest 0.615	lovely 0.804	wonderful 0.659
terrible	awful 0.749	dreadful 0.827	horrendous 0.862	horrible 0.943	horrid 0.771
fantastic	amazing 0.789	fabulous 0.708	great 0.742	marvelous 0.796	wonderful 0.859

4.3.2 Long Short Term Memory (LSTM)

Recurrent Neural Network (RNN) model is the predecessor of LSTM model. It has been widely researched in the area like speech recognition, natural language processing, etc. This type of model was prevalent in language topic since traditional machine learning and neural network have trouble to simulate the way our memory cells understand the contextual meaning. However, RNN has a different architecture with feedback loops which make the memory cell state persistent. The loop passes the memory cell from the previous state to the current state and the way it works like time series structure in econometrics.

While RNN was important in speech and text mining due to its persistency between previous state and current state, it still had flaws. RNN was capable of passing previous information cell to the current cell, only when the distance between these cells was small. As distance gets long, the performance of RNNs drops significantly. Moreover, the performance also drops significantly as the depth of layers is increasing. With the number of layers increasing, some coefficients in the model are close to zero after multiplications. This gradient vanishing problem caused the difficulty in model training. Long Short Term Memory (LSTM) is an extension of RNN model, which ease this vanishing problem by memorizing the dependency in long distance. Hochrieter and Schmidhuber (1997) first proposed this model and assumed the linear dependency between previous state and current state. Input and Output gates were introduced to control the ingredients of the input and output. Moreover, the Forget gates in this model, which control proportion of memory of current state, are allowed to pass to next state. The gates are computed as:

$$G_i^t = \sigma(W_i x^t + U_i h^{t-1} + b_i \ ) \qquad (4\text{-}1)$$

$$G_f^t = \sigma\left(W_f x^t + U_f h^{t-1} + b_f \ \right) \qquad (4\text{-}2)$$

$$G_o^t = \sigma(W_o x^t + U_o h^{t-1} + b_o \ ) \qquad (4\text{-}3)$$

Where G^t is gate function in time t, h^{t-1} is hidden activation at time t-l, x^t is input in time t. W and U represents the coefficient matrix of each gate, and b is the interception coefficient. Different subscript i, f and o represent the equation in Input gate, Forget gate and Output gate and σ is an activation function for different gates. In LSTM model, activation functions usually are logistic function or Rectified Linear function (ReLu). The memory cell at time t is computed in this way:

$$C^t = G_f^t \times C^{t-1} + G_i^t \times tanh(W_C x^t + U_C h^{t-1} + b_C) \qquad (4\text{-}4)$$

Upper case C represents the variable for cell state. tanh is hyperbolic tangent function ($\tanh x = sinh\, x \div \cosh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$). C^t is a combination of adding previous state C^{t-1} with current input x^t by proportion of gate. The activation function h^t is computed as:

$$h^t = G_o^t \times \tanh(C^t) \tag{4-5}$$

Figure 2.a illustrates the single memory cell architecture of LSTM, and Figure 2.b illustrates the expanded multiple stages of LSTM. The sequence of word vector is used in each time step. The outputs and previous memory cells are used to compute in current cell. In the end, the output from the penultimate LSTM cell is fed into a fully connected (FC) linear combined layer to output predicted price. The last FC layer function is similar to the linear regression form.

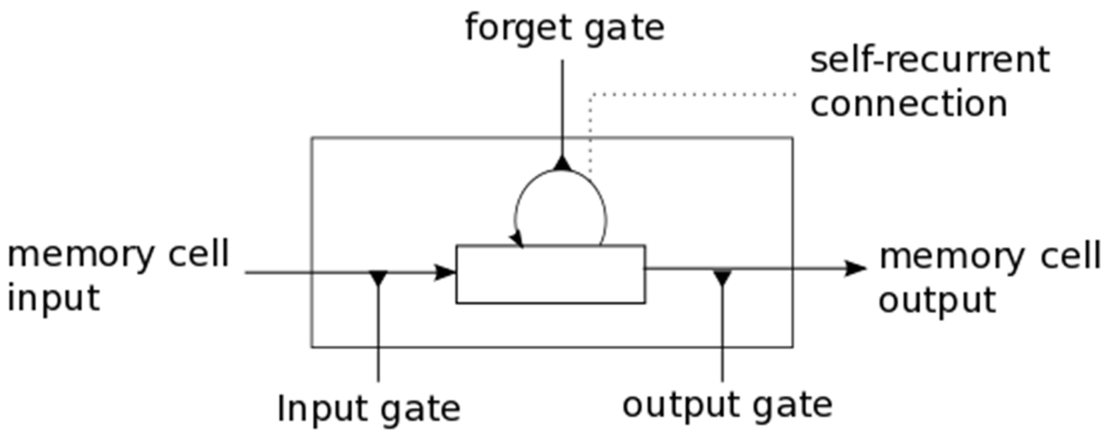


Figure 4.14Illustration of A Long Short Term Memory

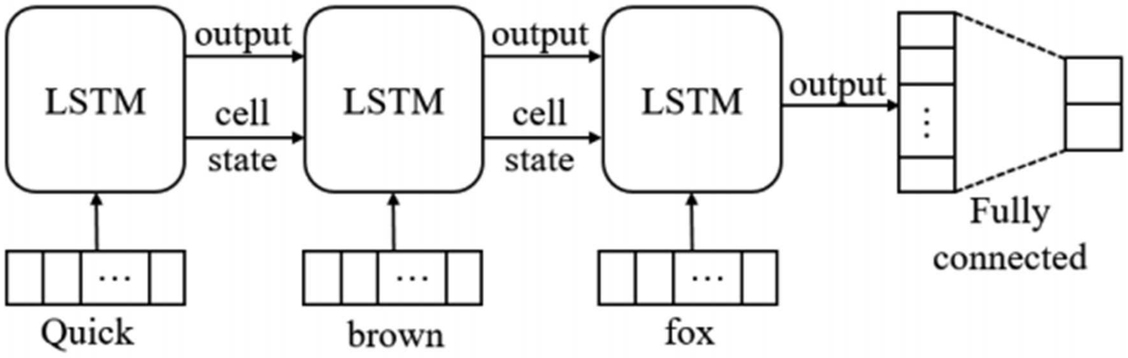


Figure 4.15 Illustration of A Long Short Term Memory with Fully Connected Layer

4.3.3 Convolutional Neural Network (CNN)

4.3.3.1 Convolutional Layer

The inputs in word embedding model are text and are treated as a sequence of words:$[w_i, \dots w_{|s|}]$, each word is drawn from vocabulary set V. Words are represented by vector $w \in R^{1\times d}$ in a word embedding matrix $s \in R^{d\times|s|}$. S is the length of the sentence and d is the length word embedding vectors. Sentence matrix s is a concatenation of word vectors $\mathrm{w_i}$.

In the text description, the lengths of words in each sentence are different. However, CNN model requires each input has the same size, which is not a natural fit for sentences. The solution is to set an arbitrary cutoff length W which most sentences are shorter than this value. Then add zero padding for sentence length is shorter than W and shorten the sentence length if it is longer than W. In this layer, there are filter matrix $F \in R^{d\times m}$, where m is the filter size. Filters run convolution operations on sentence matrix and generate m-dimension features from original sentences. This convolution layer result is a matrix c of dimension $\mathrm{n} \times (\mathrm{s} + \mathrm{m} - 1)$, is calculated:

$$\mathrm{c}_{\mathrm{i,j}} = m_{i:}s_{i,j-m+1:j} \tag{4-6}$$

4.3.3.2 Pooling Layer

The output from the convolutional layer is passed to pooling layer as Figure 3 illustrates, where the goal is to aggregate the convolution and extract the significant signal among them. The form of pooling is:

$$\mathrm{c}_{\mathrm{pooling}} = \begin{bmatrix} pool(c_1 + b_1) \\ \cdots \\ pool(c_n + b_n) \end{bmatrix} \tag{4-7}$$

where c_{i} is ith convolutional layer element adding with bias term b_{i}. The common choices for pooling operation are max or average pooling. Currently, max-pooling has been used more frequently since maximum function is a non-linear transformation.

In a nutshell, the convolution layer utilizes a linear transformation to accumulate neighborhood information signal, and pooling layer selects the significant signal to the next stage. The architecture of the convolutional layer and pooling layer are illustrated in Figure 3.

4.3.4 Loss Function

Regardless of which neural network function is used, the output of the penultimate layer is passed to a fully connected regression layer since it is a price prediction problem. Same as Chapter 2, we use log-form mean square error as the loss function. The last layer in neural network is a linear form with bias term just like classic linear regression:

$$\ln p_{ikt} = x_{ikt}\beta + w_{ikt}\gamma + \epsilon_{ik} \tag{4-8}$$

where $\mathrm{x}_{\mathrm{ikt}}$ is the vector of house basic and structural variables. And w_{ikt} is the vector of neural network output computed from text description i.e. unstructured text variables.

The critical difference between penultimate layer and linear regression is w_{ikt} here is not a vector with discrete numbers, but a vector with continuous intermediate value obtained from the lower layer. Moreover, the length of the vector is not length of words used in the collection of documents, but a hyper-parameter obtained from the neural network. In the end, Neural network is a non-convex problem with many parameters in lower layers. The optimization method to search maximum is different to linear regression and the coefficient in here does not have explaining power.

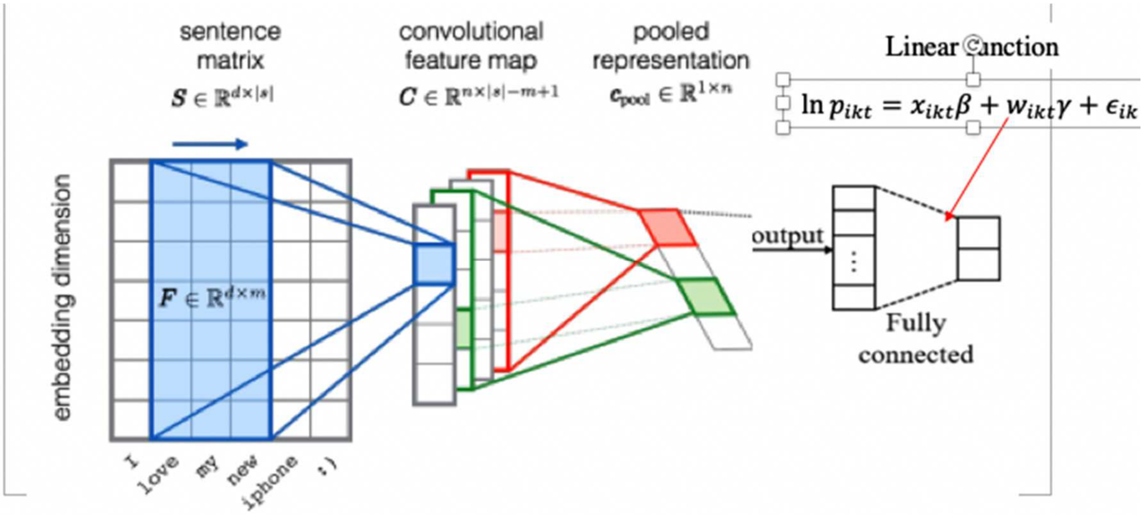

sentence matrix
$S \in \mathbb{R}^{d\times|s|}$
convolutional feature map
$C \in \mathbb{R}^{n\times|s|-m+1}$
pooled representation
$c_{pool} \in \mathbb{R}^{1\times n}$
Linear Function
$\ln p_{ikt} = x_{ikt}\beta + w_{ikt}\gamma + \epsilon_{ik}$
embedding dimension
$F \in \mathbb{R}^{d\times m}$
output
⋮
Fully connected
I
love
my
new
iphone
:)

Figure 4.16 Architecture of Deep Learning for Sentence

4.4 Data

In this paper, there are two types of data used for different purposes. One source of data is for price prediction, and I use the same Zillow.com data as Chapter 2. The other data is source for Word2Vec pre-training, which is from Google News and can be public downloaded.

There are 15,545 houses sampled in the U.S. from Zillow.com, and all of them are single family houses with last sold dates from January 2016 to September 2016. In this work, our main attention is on information extraction from unstructured house description. One fundamental assumption in here is home owners deliver correct property information and will not deliberately miss house attributes in their description. Moreover, we don't need TF-IDF to filter low-frequency words like conventional language model, but keep all words that have appeared in pre-training source of data. The assumption here is pre-training Google News data source is much larger than Zillow house data and should have covered word's semantic meaning in most scenarios. Thus, it is not necessary to filter out the low-frequency words. There are around 9,000 different words used in Zillow.com dataset. The Open source Google News data has around 100 million different articles covering different topics with a total 100 billion words.

4.5 Result

4.5.1 Initialization of Model Parameters

The essence of neural network is to solve a non-convex optimization problem. We choose stochastic gradient descent (SGD) to train both LSTM and CNN neural network and apply back-propagation algorithm to compute the gradients. Neural network model has much more parameters compared to other models, and overfitting is the typical problem it will suffer. We choose one of the famous approach to avoid this problem in neural networks: dropout (Srivastava et al.2014). Dropout technique prevents overfitting by setting a portion of hidden nodes' coefficient to zero (drop out) during forward phase when computing the output layer.

In this paper, we use 5-fold cross validation in model evaluation. Each subset contains around 3,000 houses data. The evaluation metric for this work is Root Mean Squared Logarithmic Error (RMSLE) and calculated as:

$$\epsilon = \sqrt{\frac{1}{n} \sum_{\mathrm{i=1}}^{\mathrm{n}} (\log(p_i + 1) - \log(\hat{p}_i + 1))^2} \tag{4-9}$$

where ϵ is RMSLE value; n is the total number of observations in dataset, $\mathrm{p_i}$ is actual price and $\hat{p}_i$ is the predicted price by model.

Before train a neural network model, we compare the quality of using pre-trained word embedding vectors with no pre-trained vectors. The pre-trained word embedding vectors give us a better initial weight to start computing and better result. we make the comparison between (1) entirely at random weight of training model; (2) using word embedding from pre-trained Google News dataset. Without tuning any other parameter, the (2) has RMSLE at 0.134, which is much smaller than RMSLE in (1) at 0.155. For the rest of this paper, we use pre-trained word vectors to initialize the weight for LSTM and CNN models.

4.5.2 Experiment 1: Size of Word Embedding Vectors for Neural Network

Word embedding vectors is a key component in both LSTM and CNN model. Word embedding vectors are trained with Skip-Gram model, which will be explained in the Appendix C. The length of the word embedding vector is the goal to optimize in Experiment 1.

The LSTM model receives a sequence of word vectors and creates an output layer with a length of 50. Then at the end of LSTM model, there is a one last fully connected linear form layer. The linear form layer adds house basic information and structured information not been used in previous LSTM layer. The output will be the predicted price for the house.

For the CNN model, this experiment uses two convolutional layers and two pooling layers. Both convolutional layers have 6 filters with a width of 3. The top of the network has dropout layer with 0.5 probability and fully connected layer. The network is trained by “Adaptive Gradient” (Adagrad).

The result of experiment shows in Table 4.2. The word vectors with length of less than 40 is less accurate than using longer vectors. The accuracy increases as word vector length is increasing. In the end, the word vector length at 160 in the remaining experiments to optimize the performance. For the rest of this paper, we will set word embedding vector at the length of 160 in pre-training.

Table 4.17 RMSLE on Predicted Price Model with Various Word Vector Length

	RMSLE by Word Vector Length					
	20	40	80	100	160	200
LSTM	0.164	0.142	0.126	0.113	**0.112**	0.118
CNN	0.156	0.149	0.135	0.125	**0.119**	0.119

4.5.3 Experiment 2: Tuning Parameter for LSTM

In this part, we try to find the best size of the penultimate output layer in LSTM that finally fully connected to the last linear form function(RMSLE). We use the optimized word vector length 160 in experiment 1. We have tried both experiments without the hidden node by directly using RMSLE on LSTM output. Table 4.3 shows the result of the best accuracy. The result displays that best accuracy comes from using 16 hidden nodes. Moreover, results also show the accuracy drops when hidden node layer is dropped compare to with hidden node layer. It indicates that an appropriate number of hidden nodes improves the model performance. We will use 16 hidden nodes for rest of LSTM model analysis.

Table 4.18: RMSLE on Various Number of Hidden Nodes in LSTM

	RMSLE by Number of Hidden Nodes				
	No	4	8	16	32
LSTM	0.1417	0.1213	0.1165	**0.1105**	0.1109

4.5.4 Experiment 3: Tuning Parameter for CNN

In this part, we use the CNN model as described in Section 4.3. There are a pair of parameters that can be optimized: number of filters and filter widths. Table 4.4 shows the optimal number of filters that get best RMSLE error is 4. Similarly, Table 4.5 shows the optimal filter width for CNN model is 3. It is worthy of mentioning that we use the same filter width for both convolution layers for simplifying assumption and reducing the computation burden.

Table 4.19: RMSLE on Various Number of Filters in CNN

	RMSLE by Number of Filters				
	2	4	6	8	10
CNN	0.1211	0.1191	**0.1179**	**0.1179**	0.1189

Table 4.20: RMSLE on Various Number of Filter Size in CNN

	RMSLE by Filter Size				
	1	3	5	7	9
CNN	0.1184	**0.1179**	0.1181	0.1191	0.1202

4.5.5 Experiment 4: Comparison of Neural Network and Other Machine Leaning Model

In this part, we compare the neural network model result with traditional machine learning models. The traditional machine learning models include linear regression model and tree-based model discussed in Chapter 2, assuming all traditional models use bag-of-words approach for word vectors. The bag-of-word approach uses TF-IDF to calculate the relevant frequency of words and filtered out some meaningless words, which is the same as Chapter 1.

Table 4.6 shows the result of the comparison. Linear regression, random forest and gradient boosting decision tree(GBDT) are used as baseline models in Experiment 4. The result shows that both models, LSTM and CNN, have lower prediction error compared to other traditional machine learning models.

Table 4.21: RMSLE Comparison between Baseline and Neural Network Model

Model	RMSLE of Model
LSTM	**0.1105**
CNN	**0.1179**
Linear Regression	0.2172
Random Forest	0.1927
GBDT	0.1784

4.5.6 Experiment 5: Words Order

In this part, we analyze whether words order in a sentence would affect the result or not. It is worthy of mentioning for baseline models like bag-of-words, input order does not affect the result. However, for the neural network model, re-shuffling the order of the words in sentences sometimes causes different results. Table 4.7 validates this opinion. The prediction error increases but not in a significant way when the order of the words is re-shuffled. Moreover, RMSLE in the re-shuffled case is still better than regression and tree-based models. The reason behind this is pre-trained word embedding vectors already learn the semantic and synthetic meaning of words well, but re-shuffled input order will still cause some semantic errors, but is not a large proportion in house description data. One example is "I am glad not to buy the house" and " I am not glad to buy the house". Switch the order of "glad not" and "not glad" will cause different sentiment meanings in the word. However, this type of sentence only takes a small proportion of data, and the effect will not impact the result significantly.

Table 4.22: RMSLE Comparison between Input with the Original Order and Re-shuffled Input Order

	RMSLE	
	Original Order	Re-shuffled words order
LSTM	**0.1105**	0.1213
CNN	**0.1179**	0.1258

4.6 Conclusion

In this paper, we apply two distinct neural network models with Zillow house data to compare their performance in house price prediction. First, we propose an experiment to find the best parameter of LSTM and CNN. Then I show the best price prediction model is LSTM followed by CNN. Both models have significantly lower prediction errors compared to traditional models, such as linear regression, random forest and GBDT. Lastly, we re-shuffle the word orders to show the input order of the sentence also has a critical impact on the neural network model performance. These six experiments find the optimized parameters in LSTM and CNN model. Last but not the least, this paper clearly shows that neural network model has a significant advantage in utilizing text information to predict the house price.

This study shed light on other economic research paper to apply neural network technique on price prediction topic. Moreover, this paper applies Word2Vec technique to extract free text information, which is also innovative and can be applied for a similar research topic.

References

Hochreiter S, Schmidhuber J. Long short-term memory[J]. *Neural computation*, 1997, 9(8): 1735-1780.

Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences[J]. *arXiv* preprint arXiv:1404.2188, 2014.

Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. *arXiv* preprint arXiv:1301.3781, 2013.

Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model[C]//*Eleventh annual conference of the international speech communication association*. 2010.

Ouyang X, Zhou P, Li C H, et al. Sentiment analysis using convolutional neural network[C]//2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. *IEEE*, 2015: 2359-2364.

Pennington J, Socher R, Manning C. Glove: Global vectors for word representation[C]//*Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014: 1532-1543.

Socher, Richard. " CS224D: Deep Learning for NLP Part 1". https://cs224d.stanford.edu/

Severyn A, Moschitti A. Twitter sentiment analysis with deep convolutional neural networks[C]//*Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM*, 2015: 959-962.

Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. *The Journal of Machine Learning Research*, 2014, 15(1): 1929-1958.

Vateekul, Peerapon, and Thanabhat Koomsubha. "A study of sentiment analysis using deep learning techniques on Thai Twitter data." *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. IEEE, 2016.

Wang X, Liu Y, Chengjie S U N, et al. Predicting polarities of tweets by composing word embeddings with long short-term memory[C]//*Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2015, 1: 1343-1353.

CHAPTER 5. GENERAL DISCUSSIONS

My dissertation develops and extends data-driven machine learning tools based on high-dimensional business data to help researchers and companies better understand consumers' or users' decision patterns, and subsequently predict and influence their following behaviors.

Findings in my dissertation show nonlinear machine learning model often have better performance than linear model in high-dimensional data problem. The nonlinear model has shown better out-of-sample prediction accuracy and robust model and variable selection process. All three chapters in this dissertation are empirical work on this topic.

Based on the three studies in the dissertation, interesting research directions are emerging. Firstly, since the data quantity and quality both have a crucial influence on the insights, it would be interesting to develop reliable causality and heterogeneous treatment effect, which can help understand consumer behavior pattern deeply and fully generalize to the counterfactual case. Secondly, the free text information can be integrated into the neural network model variously and have different performance. The model performance might have the potential to optimize if we try the top-notched and complicated neural network framework, for example, BERT. However, due to the high requirement of computation power, it is challenging to implement and get the result.

APPENDIX A. ALGORITHM USED IN CHAPTER 2

A.1 Bagging, Ensembling and Random Forest

Bagging is short term for bootstrap aggregation. The idea of bagging method is re-sampling N examples from data set D with replacement to obtain a new dataset $\mathrm{D_i}$. Bagging generates m new training dataset $\mathrm{D_1}, D_2, \ldots, D_m$. Using the m bootstrap samples to obtain m different fitted models $\mathrm{m_1}, \mathrm{m_2}, \ldots \mathrm{m_m}$ and average the output (for m models) for aggregation result. Bagging method works reasonably well if each fitted model $\mathrm{m_i}$ is sensitive to data randomness.

Ensembling is a method for combining many models by averaging each of them. It is a form of meta learning and focus on how to combine multiple results. Generally, ensembles of models perform better than single model. Bagging and boosting in section A.2 are both one of ensemble technique.

Random Forest is a special case of bagging method by generating m samples with replacement and building m different regression trees. The specialty of random forest is algorithm subsets attributes without replacement for each generated dataset $\mathrm{D_i}$. Choosing subset of variables reduces correlations across variables and prevents overfitting by lower variance of averaging result. Friedman et al.(2001) claims that random forest can be applied with parallel computing and deal with irrelevant attributes.

A.2 Gradient Boosting Regression Tree

Boosting definition is an ensemble algorithm which converts weak learner (weak performance model) to strong learners. If the way to convert weak learner is using regression trees and averaging all the weak learners, it is same as random forest.

For gradient boosting regression tree, it also uses tree models as base model. A new tree is added to weighted sum of previous trees instead of average all trees in random forest.

Model is initialed set to be average of the outcomes, $\hat{y}$ for N observations. At each recursive iteration, model parameters are selected to correct error term from previous iteration. The math formula can be considered as additive model with form:

$$\mathrm{F(x)} = \sum_{\mathrm{m=1}}^{\mathrm{M}} \gamma_{\mathrm{m}} \mathrm{h}_{\mathrm{m}}(\mathrm{x})$$

Where $\mathrm{h}_{\mathrm{m}}(\mathrm{x})$ are basis functions which is also weak learner in boosting. Gradient boosting uses decision tree of fixed size as weak learner $\mathrm{h}_{\mathrm{m}}(\mathrm{x})$. For each recursive iteration, additive model is using a forward stagewise:

$$\mathrm{F}_{\mathrm{m}}(\mathrm{x}) = \mathrm{F}_{\mathrm{m-1}}(\mathrm{x}) + \gamma_{\mathrm{m}} \mathrm{h}_{\mathrm{m}}(\mathrm{x})$$

At each stage, $\mathrm{h}_{\mathrm{m}}(\mathrm{x})$ is regression tree model chosen to minimize loss function L given current boosted model $\mathrm{F}_{\mathrm{m-1}}$

$$\mathrm{F}_{\mathrm{m}}(x) = F_{m-1}(x) + \underset{h}{argmin} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + h(x))$$

Different to other boosting method, gradient boosting solves loss minimization problem with steepest descent. And direction of steepest descent is negative gradient of loss function evaluated at current model $\mathrm{F}_{\mathrm{m-1}}$.

$$\mathrm{F_m}(x) = F_{m-1}(x) - \gamma_m \sum_{i=1}^{n} \nabla_F L(y_i,\ F_{m-1}(x_i))$$

where γ_m is linear search along the gradient:

$$\gamma_\mathrm{m} = \underset{\gamma}{argmin} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) - \gamma \frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)})$$

The algorithm is shown in table A.1. In practice, gradient boosting has a hyper parameter μ as shrinkage parameter to prevent overfitting. Gradient boosting model directly learn descent from residuals of previous model which effectively reduce error from training set. Shrinkage parameter effectively control the influence of each additional tree with form:

$$\mathrm{F_m}(x) = F_{m-1}(x) + \mu\gamma_\mathrm{m} h_m(x)$$

This form prevents errors in one iteration have too much impact on final boosted tree model. In practice, both number of trees M and shrinkage parameter μ affect the selection of optimal model.

Table A.1 Gradient Boosting Algorithm roadmap:

1. $\mathrm{F_0(x)} = \underset{\gamma}{argmin}\ \sum_{\mathrm{i=1}}^{\mathrm{N}} L(y_i, \gamma)$
2. For m = 1 to M do:

   (1) Compute $\mathrm{e_m}(x_i) = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x) = F_{m-1}(x)}$, for i = 1,2,…N

   (2) Fit regression tree $\mathrm{h_m}(x)$, to the errors, $\mathrm{e_m}(x_i)$ with J cuts and compute

$$\gamma_{\mathrm{m}} = argmin_{\gamma} \sum_{\mathrm{i=1}}^{\mathrm{N}} L\big(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)\big)$$

(3) $\mathrm{F_m}(x) = F_{m-1}(x) + \gamma_{\mathrm{m}} h_m(x)$

end for

3. Get predicted outcome for observation x, $\mathrm{F_M}(x)$

A.3 Tree Model Honesty Assumption and Asymptotic Efficient

In Wager and Athey (2017), they have proved the tree-based model estimate is asymptotic normal if it is propensity tree or double sample trees. The tree model used in this paper obeys the propensity tree. The propensity tree has procedure like this :

Propensity trees use only the treatment assignment indicator $\mathrm{W_i}$ to place splits, and save response $\mathrm{Y_i}$ for estimating τ.

Input: n training examples $(\mathrm{X_i}, Y_i, W_i)$, where $\mathrm{X_i}$ are features, , Y_i is response and $\mathrm{W_i}$ is treatment assignment. Then do the following steps:

1. Draw a random subsample I ∈ {1, … n} of size |I| =s (no replacement)

2. Train a classification tree using sample I where outcome is the treatment assignment i.e. on the $(\mathrm{X_i}, W_i)$, pairs with i ∈ I.

3. Estimate τ(x) using (A.3.1) on the leaf containing x.

$$\tau(\mathrm{x}) = \frac{1}{|\{i: W_i = 1, X_i \in \mathrm{L}\}|} \Sigma \sum_{\{i:W_i=1,X_i\in\mathrm{L}\}} Y_i - \frac{1}{|\{i: W_i = 0, X_i \in \mathrm{L}\}|} \Sigma \sum_{\{i:W_i=0,X_i\in\mathrm{L}\}} Y_i \quad (A.3.1)$$

If tree follows propensity tree procedure and get estimator $\hat{\mu}(\mathrm{x})$. Then there exists a $\sigma(\mathrm{x})$ →0 such that:

$$\frac{\hat{\mu}(\mathrm{x}) - \mathrm{E}(\hat{\mu}(\mathrm{x}))}{\sigma(x)} \rightarrow N(0,1)$$

where N(0,1) is standard normal distribution.

APPENDIX B. INSTRUMENT VARIABLE VERIFICATION FOR CHAPTER 3

In this Appendix, we will show more detail about the validity of instrument variable that proposed in this chapter. Specifically, we check the feasibility of conditional independence assumption under different types of unobservable shocks, both from the demand and supply side. The signals potentially affect the performance of the instruments.

1. Demand Shocks:

Unobserved demand shocks might come from local temporary events (e.g. conference or Super-Bowl). These shocks create correlations in prices across hotels through correlated demand. To simplify the illustration, we will use two competitors' simultaneous system to make the explanation:

$$\mathrm{P}_{1\mathrm{jt}} = \alpha_1 + \beta_{12} P_{2jt} + \delta_1 U_{1jt} + \epsilon_{1jt} \tag{B-1}$$

$$\mathrm{P}_{2\mathrm{jt}} = \alpha_2 + \beta_{21} P_{1jt} + \delta_2 U_{2jt} + \epsilon_{2jt} \tag{B-2}$$

$\mathrm{U}_{\mathrm{ijt}}$ represents the instrument to estimate $\mathrm{P}_{\mathrm{ijt}}$ in the system of equations. When hotels are competing, the hotel-specific display is likely correlated, the instrument is still an unbiased estimate if the other hotel's display is explicitly controlled. For example, in equation (B-1), even instrument $\mathrm{U}_{2\mathrm{jt}}$ is likely correlate with response variable $\mathrm{P}_{1\mathrm{jt}}$ through its correlation with $\mathrm{U}_{1\mathrm{jt}}$, the estimate of competitive price impact β_{12} is still unbiased if we control $\mathrm{E}_{1\mathrm{jt}}$ follows the conditional independence condition. In here, it means:

$$\mathrm{Cov}\left(\mathrm{U}_{1\mathrm{jt}}, \epsilon_{2\mathrm{jt}} \middle| U_{2jt}\right) = 0.$$

We use simulation data to estimate this scenario in Table B.1. Table B.1. shows the estimated result based on data simulated from equation (B-1) and (B-2). Column 1 and 2 show the result with independent display, and column 3 and 4 shows the results with correlated display. In independent display case, the controlling for hotel-specific display in the system is not significant. The IV obtains unbiased estimates with (Column 2) and without (Column 1) accounting for exposure E_1. On the other side, if demand is correlated across hotels, the IV only obtains unbiased estimates when display U_1 is explicitly controlled in the price equation (Column 4), but not otherwise (Column 3).

Moreover, the unobserved shocks ϵ_{ijt} are also likely correlate across shocks. The data is from a single channel (other competitor websites and travel agencies are not observable) thus only partial demand variations are observed. The unobserved display from other channels are in the error term ϵ_{ijt}. Both observed and unobserved display can be correlated across hotels. In the same time, display of one specific hotel can be correlated across channels. That is, not only $\mathrm{U}_{1\mathrm{jt}}$ and $\mathrm{U}_{2\mathrm{jt}}$ are correlated, but also unobserved errors term ϵ_{i1jt} and ϵ_{2jt} can be correlated due to unobserved display from other channels. Furthermore, U_{ijt} and ϵ_{ijt} in same hotel i can be correlated due to the correlation of exposures from different channels. However, this possible correlation would not hurt the IV's validity. The estimates would be unbiased if the conditional independent assumption holds. This is a reasonable assumption in this specific problem. For simplicity, we use “Expedia channel” and “Other channel” for illustration. The conditional independence requires that hotel 1's “Other channel” display is not correlated with hotel 2's “Expedia channel” display once hotel 1's “Expedia channel” display is partial out. It means the correlation will exist only through hotel 1's “Expedia channel” display. Column 5 and 6 in Table B.3 show the simulation result in this case. Table B.3 shows that even with the correlations across hotels and across channels, the estimate is still unbiased (Column 6) when the displays are explicitly controlled in the system equations.

Table B.23Variable of Instruments:

	Indep. Hotel Demand Indep. Channel		Corr. Hotel Demand Indep. Channel Demand		Corr. Hotel Demand Corr. Channel Demand	
	W/O IV (1)	W/ IV (2)	W/O IV (3)	W/ IV (4)	W/O IV (5)	W/ IV (6)
P2	0.783 (0.009)	0.489 (0.012)	0.921 (0.005)	0.481 (0.013)	0.912 (0.005)	0.487 (0.007)
U1		1.007 (0.014)		1.022 (0.022)		1.492 (0.021)
const	38.51 (2.112)	103.5 (2.762)	14.85 (1.153)	103.2 (2.525)	14.85 (1.024)	103.9 (1.742)
Adj R-Square	0.641	0.754	0.853	0.840	0.853	0.892

All estimates are significant. Data are simulating on equation (B-1) and (B-2). True value for parameters: $\alpha_1 = \alpha_2 = 100, \beta_{12} = \beta_{21} = 0.5, \delta_1 = \delta_2 = 1$.

- Column (1) and (2) assumes $\text{cov}(\mathrm{U}_{1\mathrm{jt}}, U_{2jt}) = 0$ and $\text{cov}(\mathrm{U}_{\mathrm{ijt}}, \epsilon_{ijt}) = 0$
- Column (3) and (4) assumes $\text{cov}(\mathrm{U}_{1\mathrm{jt}}, U_{2jt}) \neq 0$ but $\text{cov}(\mathrm{U}_{\mathrm{ijt}}, \epsilon_{ijt}) = 0$
- Column (5) and (6) assumes $\text{cov}(\mathrm{U}_{1\mathrm{jt}}, U_{2jt}) \neq 0,\ \text{cov}(\mathrm{U}_{\mathrm{ijt}}, \epsilon_{ijt}) = 0$

Conditional independence holds in all time. $\text{Cov}(\mathrm{U}_{1\mathrm{jt}}, \epsilon_{2\mathrm{jt}} | U_{2jt}) = cov\left(\left(\mathrm{U}_{2\mathrm{jt}}, \epsilon_{1\mathrm{jt}} | U_{1jt}\right)\right) = 0$.

2. Supply Shocks

Unobserved supply shocks might come from costs or capacity of the hotels, and it could affect multiple hotels at the same time. For example, a fire in the cities spread to multiple blocks in the city and reduces the capacity of multiple hotels. Or chain hotels like Hilton is acquired by a competitor which is likely to bring price adjustment for every hotel in this brand. Such shocks cause correlations in supply side. If these shocks are uncorrelated with hotel displays, like customers do not know about this adjustment, then correlated supply shocks will not affect the validity of IV and there is no need to discuss. If these shocks are correlated with displays, then the uncontrolled correlation might affect the validity of instrument. However, the random nature of these incidences, like fire or acquisition, are infrequent events during the short one-month time span in our data. This type of event needs to be very frequently and consistently happening to affect the system equation. In this part, we will not assume this rare case shocks happen and undermine the validity of IV.

APPENDIX C. SKIP-GRAM MODEL FOR CHAPTER 4

Appendix C follow Socher (2016) lecture notes in Stanford CS224N to explain how the Skip-Gram model works. First, we need to create such a model that will assign a probability to a sequence of tokens. We're going to train the neural network to do the following.

Given a specific word in the middle of a sentence (the input word), look at the words nearby and pick one at random. The network is going to tell us the probability for every word in our vocabulary of being the "nearby word" that we chose. When I say "nearby", there is a "window size" parameter to the algorithm. A typical window size might be 5, meaning 5 words behind and 5 words ahead (i.e 10 in total). The output probabilities are going to relate to how likely it is to find each vocabulary word nearby our input word. For example, if you gave the trained network the input word "Soviet", the output probabilities are going to be much higher for words like "Union" and "Russia" than for unrelated words like "watermelon" and "kangaroo".

We can breakdown the way this model works in these 6 steps:

1. Generate one hot input vector x

2. Get embedded word vectors for the context $\mathrm{v_c} = Vx$

3. set $\hat{\mathrm{v}} = \mathrm{v_c}$

4. Generate 2m score vectors, $\mathrm{u_{c-m}}, \dots u_{c-1}, u_{c+1}, \dots u_{c+m}$ using $\mathrm{u} = \mathrm{Uv_c}$

5. Turn each of the scores into probabilities, $\mathrm{y_i} = \mathrm{softmax(u_i)} = \frac{\mathrm{e^{u_i}}}{\sum_{i=1}^{n} e^{u_i}}$

6. Probability vector generated to match the true probabilities which is $\mathrm{y^{c-m}}, \dots y^{c-1}, y^{c+1}, \dots y^{c+m}$, the one hot vectors of the actual output.

After these steps, an objective function is needed to evaluate the model. I will minimize a log likelihood function based on known center word $\mathrm{w_c}$.

$$\min J = -logP(\mathrm{w}_{\mathrm{c-m}}, \dots w_{c-1}, w_{c+1}, \dots w_{c+m} \,|w_c)$$

$$= -\log\prod_{j=0,j\neq \mathrm{m}}^{2m} P(w_{c-m+j}|w_c)$$

$$= -\log\prod_{j=0,j\neq \mathrm{m}}^{2m} P(u_{c-m+j}|w_c)$$

$$= -\log\prod_{j=0,j\neq \mathrm{m}}^{2m} \frac{\exp(\mathrm{u}_{\mathrm{c-m+j}}^{\mathrm{T}} v_c}{\sum_{i=1}^{|V|} \exp(u_i^T v_c)}$$

$$= -\sum_{\mathrm{j=0,j\neq m}}^{\mathrm{2m}} \mathrm{u}_{\mathrm{c-m+j}}^{\mathrm{T}} v_c + 2mlog \sum_{\mathrm{k=1}}^{|V|} \exp(u_i^T v_c)$$

With the minimize objective function J, we can compute the gradients to unknown parameter at each iteration update via stochastic gradient descent (SGD).