Price Prediction and Deep Learning in Real Estate Valuation Models

Joseph D. Kintzel

A Thesis in the Field of Software Engineering

for the Degree of Master of Liberal Arts in Extension Studies

Harvard University

March 2019

Abstract

House price valuation plays a critical role in real estate decisions. Home buyers and sellers use valuation estimates to judge the fairness of an asking price, and lenders use these estimates to assess risk. This thesis will explore potential house price prediction models and describe the factors that drive these models. We will train our models on a triad of data types: numeric, spatial, and images. Real estate listings use a defined list of numeric features to describe a property. Spatial data manifested as location information is an integral component of house price prediction, as the same house located in different neighborhoods could sell for very different prices. Current advances in the field of deep learning will help us find predictors in images of a house for sale. We will explore the impact of each data type of this triad on house price prediction. We will show that each type of data adds value to predictive price models. An important contribution of this thesis is presented with respect to visual data: we will provide a mechanism by which images can be successfully used to improve a house price prediction algorithm.

Dedication

This thesis is dedicated to Cathy, Annika, Elizabeth, Sadie, Jack, and Lauren. My amazing family has not seen much of me these past nine months, but their love and support made this work possible.

Acknowledgment

I would like to thank my thesis director Dr. Peter V. Henstock for his guidance, wisdom, and encouragement in producing this thesis. Dr. Henstock was instrumental in shaping the ideas and content herein from inception to completion.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Predicting the purchase price of a marketplace item can be a difficult task. There are many factors that might impact the price for any item in question. Supply and demand, marketplace strength, craftsmanship, advertising, and brand identification are among factors that govern the purchase price. The "hedonic model" for house price prediction comes from an economic theory put forward by Lancaster (1966) that suggests each house is a collection of characteristics that is being purchased by a consumer: replacing or deleting any characteristic from the collection fundamentally alters the product being purchased and can alter any predictive price model as a result. Potential buyers for a home have a basic set of needs and the overall decision to purchase is partially based upon the collection of features meeting these needs. As a result, a hedonic price model might use a collection of numeric features as predictors, but that may not be enough to create an effective predictive model for the sale price of a home. The hedonic model does not capture supply and demand variables (Rosen, 1974), nor would it capture affordability.

This thesis explores the role of machine learning to predict the purchase price of a home, and if modern advancements in deep learning can leverage visual information to improve predictions. We first create a prediction model for estimating the purchase price of a house, exploring descriptive features that can aid in that task. We investigate the role of spatial information (i.e. location) on a house sale price. We then explore deep learning techniques for extracting features from images using deep network architectures. Using these features extracted from images, we attempt to improve our predictive model by creating a feature set that includes descriptive, spatial, and visual information. We explore whether a computer can be trained to identify which visual features of a house add value to a house for a potential buyer. We conclude this study by investigating the predictive power of a deep learning curb-appeal classifier to answer this final question.

Chapter 2

Prior Work

Accurately appraising the value of a house is an important task. For lending institutions, the appraisal process is critical for making loan decisions. These institutions must ensure the property values they are underwriting are commensurate with the amount of the loan so that they do not end up losing money on defaulted loans. Lenders typically order an inspection and appraisal during the loan process. Estimating the value of a home has been studied extensively from many perspectives. This chapter reviews relevant publications on this topic.

Computer vision techniques that extract knowledge from images play an increasing role in modern life. Our work attempts to build upon ideas from modern publications and extend computer vision into the field of real estate valuation. The use of images in house price prediction is still in early stages of development. This chapter also reviews relevant papers on how to use deep learning techniques to extract features from images, with the goal of using these extracted features in a predictive algorithm.

2.1 House Price Prediction Using Linear Regression

A traditional numeric value prediction method given a set of inputs is Ordinary Least Squares regression (OLS). This method of prediction relies upon a linear relationship between input variables or “features”, and an output variable predicted by these linear relationships. Kummerow’s paper (Kummerow, 2000) on real estate price estimation contains many relevant viewpoints to this project. While the crux of the paper is focused on teaching methods, Kummerow points out many statistical traps and flaws in typical regression models. He states that the sale price of a home is part of a probability distribution. This distribution cannot be seen, as we are only able to extract one data point for any one house. As a result, we use comparable properties to try and build our estimation model. As we add more and more samples, we are also increasing the standard error of our estimate. Kummerow believes that in the housing market, the Law of Large Numbers works against predictive accuracy and suggests as an alternate: the "Law of Medium Numbers." In this line of thinking, it is preferable to use a smaller number of similar houses to the house in question than to use many heterogeneous houses.

Kummerow's work then focuses on quantifying the differences between similar houses. He advocates choosing a submarket, or a subset of similar properties. Narrowing this list of houses down to the nearest ten, Kummerow then uses an adjustment grid to adjust the prices of the sample houses up or down based upon the presence of characteristics such as a fence, pool, or basement. This grid requires estimates of the value added or subtracted by these characteristics. Using iterative linear regression techniques to find the sample size with the least predictive error, he then generates a prediction for the house in question from a sample of these comparable properties. In Kummerow's study, his adjustment grid approach to house price estimation was accurate to within 10% of the purchase price on his test data 51.61% of the time.

Kummerow concludes that there are hundreds of statistically significant features to house price estimation, and that using a smaller number of similar properties to estimate house sale price is a statistically sound strategy. Also, there is no “true value” of a property because the randomness of a transaction leads to many possible purchase prices. Our study builds off these ideas from Kummerow, particularly the idea of using similar properties in the regression analysis.

Bourassa, et al. conducted a study evaluating many different methods for incorporating spatial characteristics into regression models (Bourassa, Cantoni, & Hoesli, 2010). Their study began by using only commonly accepted features of a home and leveraging them to create a regression model. Using 13,000 property sales from Louisville, Kentucky, the study attempted to apply several published methods to incorporate neighborhood or spatial characteristics into the model. They used three approaches: submarket dummy variables, a geo-statistical trend-surface model, and a clustering/nearest neighbor model. While the dummy variables did add value for the predictions, the geo-statistical method did not always converge in each submarket when combined with other methods. For the clustering method they used the residuals of the ten nearest neighbors in each submarket as features in an OLS model.

The major revelation from Bourassa’s paper is that spatial characteristics can be directly considered when assessing the value of a house, rather than using Kummerow’s subset of similar houses to help with price prediction. Using an accuracy metric of the percent of predictions within 10% of the actual sale price, OLS scored 36.5% accuracy on the data in this study, and that increased to 44% when submarket dummy variables were added. Using cluster analysis with eight geographic clusters, the accuracy score was

42.4%, and the trend-surface model to estimate price based upon position scored 37.5%. Using location as a feature improved the OLS predictions regardless of the spatial methodology.

Reed (2016) provides an interesting approach for introducing spatial features into an OLS predictive model. His paper describes using demographics within suburbs of Melbourne, Australia as features, regressed against the median house price within each suburb. This approach is unique within the house price prediction literature. The author not only attempts to build a price prediction model based on spatial variables but also includes information related to the ability to buy and the motivation to buy and/or sell. He argues that demographic information: economic factors, family status, and ethnicity variables, all impact the housing market and are largely ignored in many house price prediction models.

Since many of the demographic features in Reed's study are correlated, he uses PCA (Principal Component Analysis) to reduce the feature space. The study shows a linear relationship between these PCA features collected from individual suburban regions and median house sale price from each region. This demonstrates that there is potential for blending spatial data with family demographics to create additional predictors for a house prediction model. We use a similar approach of reducing the feature space using PCA but apply it to image features rather than demographic data.

Dubin's paper on estimating house prices from MLS (Multiple Listing Service) data focuses on the shortcomings of OLS (and hedonic price prediction in general) when not taking neighborhood information into account (Dubin, 1998). Some studies attempt to bring neighborhood quality metrics such as crime rate and school quality into the regression model. Dubin points out that even attempting to capture neighborhood quality variables is inherently flawed unless the neighborhood boundaries are clearly present and recognized, as often there is not a clear demarcation for what is considered a neighborhood. Since there is likely an inherent correlation between spatially proximate houses, Dubin attempts to capture some of this neighborhood-quality information by using a practice called kriging. Kriging is a maximum likelihood estimation technique that uses a weighted average of estimation errors (residuals) to predict the neighborhood effect on a house price. Dubin gathered housing data from the Baltimore, Maryland area. This housing data contained numerical descriptors of each house sale, the sale price, and the address of the house. These addresses were converted to coordinates for use in her calculations. In her trials she found that using the kriging approach for capturing neighborhood information increased her overall accuracy by 65.3% over OLS alone, clearly showing the importance of spatial variables in the predictive models for house prices.

Another significant paper in the field of house price prediction used a single set of housing data from Fairfax County, Virginia to compare several methodologies, including Dubin's kriging model. This paper provided a way to directly compare several methodologies and helped to provide an accuracy benchmark for this thesis (Case, Clapp, Dubin, & Rodriguez, 2004). Dubin's kriging approach was the first applied to this Virginia data set.

In addition to Dubin's contributions, the authors employed another model using homogeneous districts. They used a k-means clustering technique to create these districts, which were composed of similar census tracts. Census tracts are created by the

United States Census Bureau and are geographic regions roughly equivalent to neighborhoods (Weessies, n.d.). The authors then created separate two-stage regression models for each district. They used the residuals of the five nearest-neighbor homes (from the first-stage model predictions) as features in the final stage regression model. This idea is similar to industry practice, which involves several visits to the property to assess value, but also looks at recent sales of similar, nearby properties (known in the lingo as "comps") for comparison.

A third model applied to this housing data was Clapps' local regression model. This model used a large database to create price indices for each neighborhood. The model then attempted to create and smooth a function for price using these indices, as well as latitude and longitude (and time) for each house. This is a similar approach to Bourassa's trend-surface model, using the squares of latitude and longitude as additional features as well as the product of the two coordinates.

The results comparing the Case et al. models on the Virginia housing data set are shown in Table 1. The authors determined that the presence of the nearest neighbor residuals improved the performance of all the tested models in which they were not already present. Their general conclusion was that using similar houses to judge the value of any one house is beneficial to a regression model's predictive accuracy. Our work in this study also uses similar properties to assist in house price estimation, although we found the choice of prediction algorithm had a significant impact on the value added from these features.

Table 1. Comparison of House Price Models from Case, et al.

Model	OLS - (no location data)	Dubin-Kriging	Case - Homogeneous Districts	Clapp - Local Regression
Median Error Rate	0.1227	0.0834	0.0807	0.1131

Sommervoll and Sommervoll (2018) address spatial variables in a paper that uses a hedonic features model as a baseline, but uses a genetic algorithm to "evolve" a population of these models using differing approaches to spatial variables. This genetic approach is designed to evaluate many different permutations of spatial subgroups and form an aggregate model that best estimates house price. The authors employed three methodologies for capturing neighborhood information for 14,036 houses from Oslo, Norway: dummy variables for 12 well-known regions around Oslo, creating 12 homogeneous regions from the 53 Oslo postal codes, and using 367 rectangular cells to group homogeneous housing regions. The latter two approaches were applied using a genetic algorithm, which allowed the authors to create and evolve the aggregate models. The genetic algorithm optimized for region configurations, favoring the best to survive and influence the overall aggregate model.

In comparing these different spatial approaches, the authors determined the grid approach provided the most accurate models. They used r-squared values, or the percent of explained variation, as their performance metric. The authors observed an increase in r-squared for their linear regression models from 0.657 to 0.794 using grid squares and a genetic selection approach to generate 12 homogeneous regions based on demographic information with which to measure the spatial characteristics of a house. They did note however a large variance in model accuracy across their test runs, as each grid space contained only 23 house sales on average. The authors attempted a novel approach of creating spatial variables by linking small regions of homogeneous demographics and using a genetic approach to optimize the composition of the aggregate regions.

2.2 Neural Networks (and Beyond) in Price Prediction

In the last twenty years we have seen an increase in the use of a neural network to solve many classification and regression problems. Until recently, the computational speed for computers was too slow to make these networks practical. One paper that attempted to directly compare a hedonic linear regression model to an neural network was by Limsombunchao (2004).

Limsombunchao collected a housing data set from Christchurch, New Zealand containing 200 data points. He then created a linear regression model and a neural network model using this data and compared the resulting root mean squared error (RMSE) and r-squared values on the testing data. Of note in this study is the lack of quantitative location variables. Limsombunchao did not use any numerical spatial characteristics in his model, instead choosing dummy variables for the five geographic regions in Christchurch. Our work uses location as a feature on a finer scale, utilizing zip codes and geolocation coordinates to capture spatial characteristics.

Limsombunchao's work showed a much higher r-squared value on the test data (0.9000) for the neural network than for the linear regression model (0.6192). His neural network also achieved a RMSE of 449,111 (New Zealand dollars) compared to a RMSE of 876,216 for the linear regression model on the testing data. Limsombunchao concluded that neural networks had the potential to perform better than a hedonic linear regression model, though his limited sample size of 200 houses might call his results into question. His neural network architecture had three hidden layers of only six neurons each which is a limited learning capacity compared to recent models. This thesis uses a neural network to predict the price of a house as one potential algorithm but uses a larger model with three layers of up to 96 neurons each, and dropout layers to avoid overfitting.

A more recent comparison of a neural network and an OLS house price prediction model was described in paper by Feng and Jones (2015). The authors use housing data from Bristol, England. Location variables were collected from census data using geographic regions called “output areas” as neighborhoods, similar to the census tracts used by Case et al. These output areas were determined by the census process and represented homogeneous districts, although no details were provided in how those determinations were made. The authors collected 20 neighborhood variables from public data for each output area and appended those variables to the housing data.

The authors reported very similar accuracy and r-squared values and concluded that using a neural network provided no significant benefit over a standard hedonic model. Upon review of their methodology, this is not an unexpected result. The architecture they employed for their neural network consisted of a single hidden layer, creating a prediction model that was not substantially different than their OLS prediction model. With only a single hidden layer the network was unable to build any deeper, more abstract representations of the data structure. Had the authors explored a deeper architecture, they perhaps would have found an improved model. We will explore this “deep architecture” for a neural network in subsequent chapters of this thesis.

In addition to exploring neural networks for house price prediction, this research will also investigate other machine learning algorithms as possible solutions. Mullainathan and Spiess (2017) explored the question of house price prediction and determined random forest regression models performed significantly better at predicting house prices than OLS based models. They used a data set consisting of 10,000 houses with 150 house features, including a house's location (using census tracts) as a spatial dummy variable. Using r-squared values to compare OLS to a random forest regression model, they noted an improvement in the proportion of explained variation from 0.417 to 0.455 on unseen data. Interestingly, the authors' random forest regression model outperformed OLS in all but the upper-most and lower-most quintiles. This suggests these two classes of houses may behave differently with respect to price prediction models. For the most expensive and least expensive houses in the data set, the performance of the two models was roughly the same.

2.3 Current Professional Realty Estimates

Zillow Group represents one of the most well-known real estate shopping sites. Zillow's website presents users with a "Zestimate," which represents the service's best estimate of the true value of a home listed for sale. Additionally, the site lists current home value estimates for all houses in their database. Home owners can add or subtract features from their own property listings on Zillow.com to help Zillow refine the estimate. The methodology for these estimates is proprietary; however, Zillow does provide some statistics on the accuracy of their estimates. Zillow reports an overall median error rate of 4.3%, but that can vary depending on the market and the number of data points within each market. Generally, the larger the number of homes on Zillow in each market, the more accurate the estimates become (Zillow, 2018). Table 2 summarizes Zillow Group's accuracy metrics from two United States midwestern metropolitan areas.

Table 2. Comparison of Zillow's Accuracy in Select Markets as of 7/25/2018

Metro Area	Median Error Rate	Accuracy Within 10%	Number of Observations
Cleveland, OH	.053	0.695	830,600
Cincinnati, OH	.050	0.711	828,900

These accuracy numbers helped to establish a target goal for this thesis, which will use data from the Indianapolis, Indiana metropolitan area. Unfortunately, we did not have access to over three-quarters of a million homes. Nonetheless, these accuracy numbers presented a respected and established benchmark for which to strive.

Recently Zillow Group has implemented a computer vision system to help add value to its "Zestimates." Using computing power from AWS (Amazon Web Services), Zillow's algorithms search through the millions of available housing images to identify objects, such as granite countertops, that would add value to a home (Castellanos, 2016). This type of feature engineering in which specific items are identified by computer vision algorithms is different from the focus of this thesis. The specific image features we use are discovered by deep learning neural network algorithms and may have no obvious interpretations. Recent news from Zillow suggests that some of this "black-box" approach to features is being leveraged, using computer vision to judge the “upkeep” of a home (Harris, 2018).

2.4 Image Processing and Deep Learning

Convolutional Neural Networks (CNN) are currently the standard for deep learning/computer vision applications, consistently setting new performance benchmarks in image recognition competitions (Krizhevsky, Sutskever, & Hinton, 2017). This section discusses several of these CNN architectures, and how they can be leveraged by predictive models in other domains by transfer learning techniques.

There are several significant advances in deep learning architectures that are relevant to this paper. One of the first important advances is AlexNet. AlexNet is a convolutional neural network architecture that was designed for image recognition and was one of the first to use a technique called “dropout” to reduce over-fitting. AlexNet represents a “deep” architecture and contains over 650,000 neurons, giving a model using this architecture a large learning capacity (Krizhevsky et al., 2017). At the time of its inception in 2012, AlexNet made significant gains in accuracy for ImageNet image classification as compared to other entrants in image classification competitions. The ImageNet dataset is a well-known freely available data set containing images within categories and associated labels that can be used in machine learning and computer vision training.

Another advancement in convolutional architecture came from Simonyan and Zisserman, researchers at the Visual Graphics Group at The University of Oxford. Their architecture, now known as VGG Net, performed very well in the 2014 ImageNet Challenge scoring 2nd place in image classification. VGG Net also generalized well to other data sets. This new architecture differed from AlexNet by using smaller convolutional filters, which allowed for more layers and therefore a “deeper” network (Simonyan & Zisserman, 2014).

Winning first place in the 2015 ILSVRC (ImageNet Large Scale Visual Recognition Competition) for classifying images in the ImageNet data set was a new architecture known as ResNet. The “res” in ResNet refers to the word “residual.” The construction of this model was motivated by the desire to build a deeper network. Deep networks are normally composed of millions of parameters created by dozens of layers. Because of their size, it is difficult to train and manage these deep networks. The solution to this problem came in the form of “residual functions.” These functions can be learned and called in place of multiple layers within the neural network structure, speeding the training process and allowing for a much deeper architecture. The winning ResNet model on the 2015 ImageNet data consisted of 152 layers, opposed to the previously mentioned VGG Net architecture that consisted of 41 layers (He, Zhang, Ren, & Sun, 2016). ResNet’s residual function architecture allows the model to be very deep yet have a manageable number of components. One paper, summarizing modern advancements in computer vision over the last five years, referred to ResNet as “the cutting edge advent in DL (deep learning) architectures” (Abbas, Ibrahim, & Jaffar, 2018).

Transfer learning is the process of using a pre-trained model and applying that model to something new, usually with a similar target domain. Oquab et al. applied this concept to a data set consisting of images from the PASCAL VOC (Pattern, Analysis, Statistical Modeling and Computational Learning – Visual Object Classes) using the

AlexNet architecture pre-trained on the ImageNet data as a convolutional base (Oquab, Bottou, Laptev, & Sivic, 2014). Removing the output layer, they added two fully connected layers to better capture the difference in labels between the two image sets. They fixed the pre-trained weights of the convolutional base and then "fine-tuned" their new adaptation layers on the target task. Generally, the fine-tuned pre-trained AlexNet model performed well on the new data set, and the authors concluded that this type of network could learn features that were transferrable to other tasks.

One paper (Shin et al., 2016) on medical image classification suggests the options available at the current time for using CNN architectures for image recognition problems in medical detection are

1. Train a CNN from scratch.
2. Use "off the shelf CNN" models, or transfer learning, to augment hand-crafted features without altering the CNN models.
3. Fine-tune the network of a pre-trained CNN to the target classes by freezing certain layers and implanting custom layers at the top of the network.

They used these options to explore several computer vision tasks, such as object detection and image classification. Our work explores option two above to build better predictive models for house price using features derived from images using transfer learning.

Chapter 3

Approach

This chapter describes the approaches used to develop house price prediction models. It begins with a discussion of the testing methodology used in this paper. The evaluation metrics for the analysis of the models are also specified. It provides details on the experiments in this project including house price predictions from descriptive features, leveraging features extracted from house images to add value to those predictions, and classifying curb appeal using price predictions and extracted image features. Chapter 4 will describe the data and technology used in greater detail, and Chapter 5 will describe the results of the study.

3.1 Model Evaluation and Performance Metrics

In all cases of model evaluation, we used 10-fold cross validation, reporting the mean scores of multiple evaluation metrics across the folds. We re-randomized the folds for each model evaluation. Deep learning models require a large amount of training data to learn accurate representations. With a limited number of data points available, we decided to preserve as many data points as possible for training in choosing this evaluation methodology, although it meant training and testing each model ten times.

Several previously mentioned studies of house price prediction, as well as the Zillow Group, report the number of predictions that fall within ten percent of the true sale price as a measure of accuracy. In fact, to be considered for use by Freddie Mac (The

Federal Home Loan Mortgage Corporation), a house price estimation tool is required to estimate value within ten percent of the sale price more than half the time (Bourassa et al., 2010). Another popular academic and industry metric is the median absolute error rate (Case et al., 2004). The median absolute error rate is calculated by finding the median of the absolute values of percentage errors. We chose both evaluation metrics to judge the success of our models. We also chose these metrics to judge the quality of predictions versus previous studies.

3.2 Predictive Models from Descriptive Features

How accurately can an algorithm predict sale price given a set of descriptive features describing a house? We explored many different types of models to answer this question. We began this exploration with OLS, building a linear regression model from descriptive features. We also explored transformations on the features, as well as various collections of features, that contributed to good predictive models.

How much does a house's location impact the predictive accuracy of house price prediction models? We explore this question as well in this thesis, comparing several techniques for capturing spatial variables.

3.3 Feature Extraction from Images

Can images provide information that can improve house price prediction models? Pre-built architectures, trained to classify images from the ImageNet competition, have already learned from examples and can recognize structure-like entities, such as churches or outhouses. These pre-built models are also trained to recognize hundreds of other objects that are not structures, but we theorize that with an overlapping domain these models will prove useful. This overlapping domain should allow for transfer learning, where we use these pre-built models to extract features but apply the results to another task aside from the ImageNet classifier (Zhang, Wu, Liu, & Meng, 2018). We experimented with several leading pre-built architectures for use as a potential image feature extractor.

3.4 A Binary Classifier for Curb Appeal

Houses seldom sell for the listed sale price. The interaction between the buyer and the owner plays a large role in the negotiation of the final sale. This interaction is unique to the buyer/seller pairing, and therefore would be very difficult to predict without some background knowledge of both parties. Clearly collecting this massive amount of individual and personal data would be virtually impossible without some type of monitoring and data collection infrastructure, and as a result lies outside the boundaries of this study. On the other hand, one related factor in this negotiation is the buyer's desire to purchase the property in question. This leads us to the following assumption:

Assumption #1: A buyer's desire to buy a home is influenced by the aesthetic quality of the home.

A home with a fresh coat of paint, attractive landscaping, and a fresh appearance might increase a buyer's willingness to purchase a property, and thereby increase the price they are willing to pay. A property that appears to have faded paint, overgrown landscaping, or with a generally dilapidated appearance might not be as attractive to a buyer and sell for less than what the house might otherwise be worth if it were in good condition. Architectural features, style, or even color might also influence a buyer to see value in a property and influence the amount they are willing to spend.

Our prediction algorithms that are based on descriptive features do not take any visual information into account when making predictions about sale price. They predict price based upon functional, spatial, or temporal features, and often under-predict or over-predict price. Some of the variance in these predictions is unaccounted for by the features used in the predictive models. This leads to the next assumption:

Assumption #2: Houses that sell for more than their prediction do so in part because the buyer "sees" more value in the home. Conversely, houses that sell for less than their prediction do so in part because the buyer "sees" less value in the home. (Aesthetic features impact purchase price in some way.)

We will attempt to address this missing piece of house price prediction by training a model to analyze an image of a house and make a prediction about the house's aesthetic quality. Working under Assumption #2 above, we will train the model to match certain image characteristics to houses that sell for more or less than their predicted sale prices. In the best of cases, we would expect that aesthetics would play only a limited roll in the valuation process, so we would not expect our predictive accuracy for such a classifier to be extremely high. We do expect to see aesthetic quality play a role however and would therefore expect our classifier to perform at a level higher than random guessing.

Chapter 4

Data Set and Technology

This chapter describes the data collected for use in this study and includes the geographic area and the time period from where this data was collected. We will describe the data cleaning and preparation process, and discuss the images collected for the computer vision experiments. This chapter also describes the technologies employed for this study.

4.1 Data

There are many housing datasets available from prior data science projects and competitions, but none possessed both the descriptive data and exterior images of the houses in question. We decided the best course of action would be to create a data set specifically for this project using online real estate data. Our data collection focused on Indianapolis, Indiana in the United States and the surrounding communities that would qualify as suburbs of Indianapolis (see Figure 1).

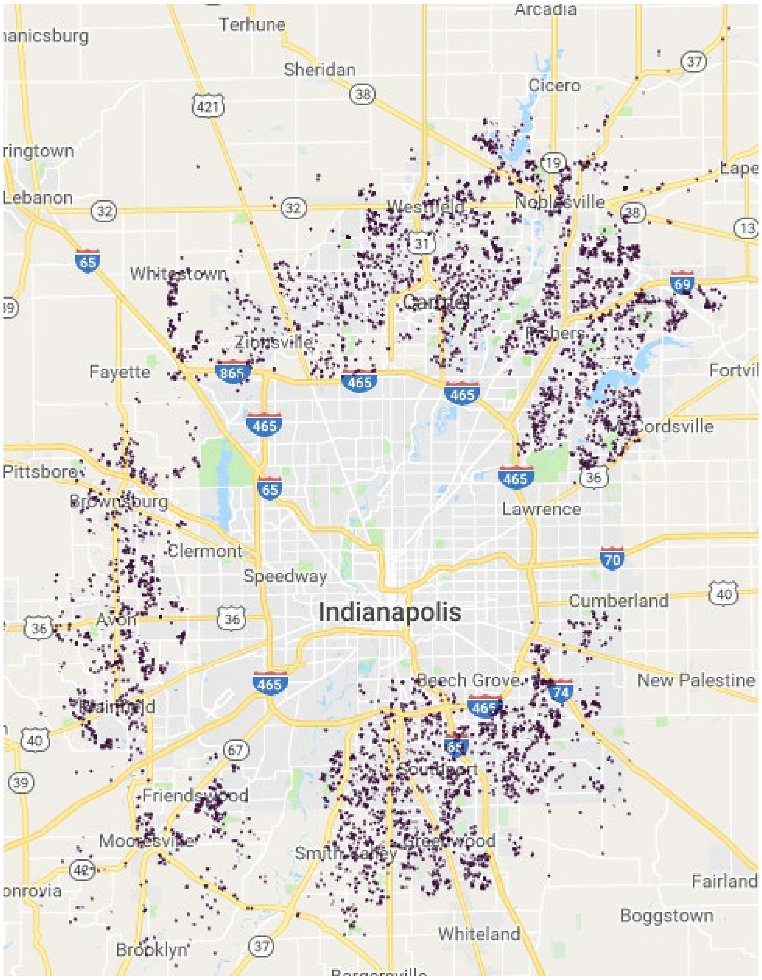


Figure 1. Plot of Locations for House Sales

Needing both house data and images of those houses, we explored several popular real estate websites to attempt to gather publicly available data. Using data provided by Redfin, a national real estate brokerage, we were able to create a data set consisting of 6007 recent sales contained within a six-month window from October 2017 to March 2018. This data set contained the major descriptive features of each house, the location, the date of sale, and the price for which the house was sold. Data was primarily collected from the suburban areas due to the sparsity of sales data from the city. This sparsity made the collection process difficult when collecting data from one zip code region at a time, presenting few data points per region during our limited time window for sales.

This data set was the result of a search within each relevant zip code for single-family homes, but it did not yet contain images of the houses. To capture these images, we enlisted the services of the vendor *import.io*. This service allowed us to build a custom web-crawler to retrieve and process images from the search results returned by Redfin. Additionally, we were able to append the MLS number for each house to each image's filename. A parsing function was written to extract the MLS number from the filename. This allowed us to download the images and then retrieve them by MLS number. The house image selected for each house was the primary image displayed on the Redfin.com website when a property was viewed by a user.

Images of houses are represented by three two-dimensional arrays of pixel values that range from 0 to 255. Each array is composed of a numerical representation of the color triad RGB, or the red-green-blue triad. These layers produce color combinations and represent digital images with up to 16.7 million different colors. The input data collected for our computer vision system has a maximum resolution of 400 pixels (rows) by 600 pixels (columns) by 3 color layers, or 400x600x3. These are typical dimensions of images used on many commercial real estate websites.

The images collected were of varying quality. Many images were as expected, such as a good curb-side representation of the house to make it appear appealing to a potential buyer. Some images, however, were not useful in judging the aesthetic value of the house. Many houses that were still under construction when listed showed the house being constructed, or a drawing of the house. Many images were scraped from Google Street View by Redfin.com when they did not have access to an actual photo of the property. Some photos were taken in portrait orientation, whereas most photos were taken in landscape orientation. Because of these photo irregularities, we manually processed the entire photo library. We cropped the houses away from any nearby houses, and deleted images that did not provide a view of the house. This process discarded about ten percent of the available images, and likely impacted about 20% of the remaining images through the cropping process.

The descriptive house data required cleaning before it could reliably be used by a prediction algorithm. Many individual records were thought to be missing data for various features, but upon review had been stored as non-numeric data types. Applying the python ‘isfinite’ function repaired these errors across several categories. The lot size feature was troublesome, as there were 131 missing lot size values in the data set. We set these missing values to the median lot size for the data. The distribution of lot sizes was skewed toward a few very large values, making the mean less representative of the center. We then created nominal variables using the lot size (by quartiles) and stored those values in our data set as either a 0, 1, 2, or 3. Duplicate copies of records were removed from the data.

Other issues appeared in the data set as well. Approximately one-third of the records for the homeowner association dues were empty, so we interpreted and replaced these missing values with zero. We attempted to preserve as many records as possible through this process, as the image processing component of this project will require as many data points as possible to learn patterns. In a few cases the sale price was missing which resulted in the removal of that record from the data set. Similarly, if the MLS number was missing the record was removed, as we will need the MLS number at later stages of the project to pair data with our images. Several of the reported zip code values did not exist in the real world, and individual house records were investigated on Redfin.com to correct these errors.

After the cleaning process, 5926 house records remained. The resulting set of features collected to predict sale price were:

- DAYS_ON_MARKET
- ZIP_CODE
- NUM_BEDS
- NUM_BATHS
- SQUARE_FEET
- LOT_SIZE (nominal)
- YEAR_BUILT
- HOA_DUES
- LATITUDE
- LONGITUDE

4.2 Data Standardization

The data were standardized, and we employed a min/max scaler to ensure each input feature was mapped to a value between 0 and 1. For each set of training data, we fit the scaler using the training data then applied that fit to both the training and testing data. We used this approach to avoid any information leakage from our training data into our testing set. We applied this scaler primarily to provide standardized input for our neural network model, even though some of our models did not require standardization to function. For consistency each type of model received the same data, standardized in this way. This process was consistent across all the training/testing data in this study.

4.3 Technologies

Python was selected as the primary language for work in this study. This language choice was made because of our familiarity with the platform and the rich library system available for Python in the machine learning field. We used *pandas* data frames and *numpy* arrays for our data structures, and coded our work using *Jupyter* notebooks. Data visualizations were created using *matplotlib* and *seaborn* libraries.

Our hardware configuration consisted of a Microsoft Windows 10 environment using an AMD 8-core CPU and 32GB of RAM. We used the Keras package with a TensorFlow backend for our work with neural networks. To fully utilize the TensorFlow backend we used a GeForce GTX 1080 GPU from Nvidia, coupled with the CUDA 9.0 toolkit installed on our system.

Chapter 5

Process and Results

This chapter details the three major components in this thesis research: 1) predicting price from descriptive data, 2) leveraging image data to enhance those predictions, and 3) the creation of a curb appeal classifier. Results and analysis are provided within each section of this chapter for the experimental components.

5.1 Predicting Price from Features

Historically, housing price prediction has been viewed as a classical use-case for linear regression (Harrison & Rubinfeld, 1978), and so we began our experiments with this approach. We applied feature selection techniques to find the linear regression model with the best predictive accuracy. We then explored other popular machine learning algorithms, including neural networks, to compare success rates of the algorithms on the housing data described in this chapter.

Exploratory Feature Analysis

There are potentially hundreds of functional attributes associated with a house for sale (Kummerow, 2000). Depending on the source of the housing data, not all these attributes will be available. Our study uses a small, common subset of attributes based on available data. In addition to functional attributes, our study uses spatial attributes to represent house location. Temporal attributes are included (number of days for sale and year of construction) except for the date of sale. While seasonality has been shown to impact house sale price (Goodman, 1993), the added complexity of a time-series analysis was omitted for this project.

We created scatterplots of each feature versus sale price to evaluate the nature of the relationships (see Figure 2). Many of the available features correlated with sale price. The YEAR_BUILT feature (the year the house was built), appeared to a have a non-linear relationship with price. Therefore, a new feature was added to the data set called ‘logAge’, which represents the natural log of the age of the house. The natural log of the age of the house appeared to have a moderately linear relationship with price and was expected to be a better predictor of price in linear models. We used this new feature with all tested models.

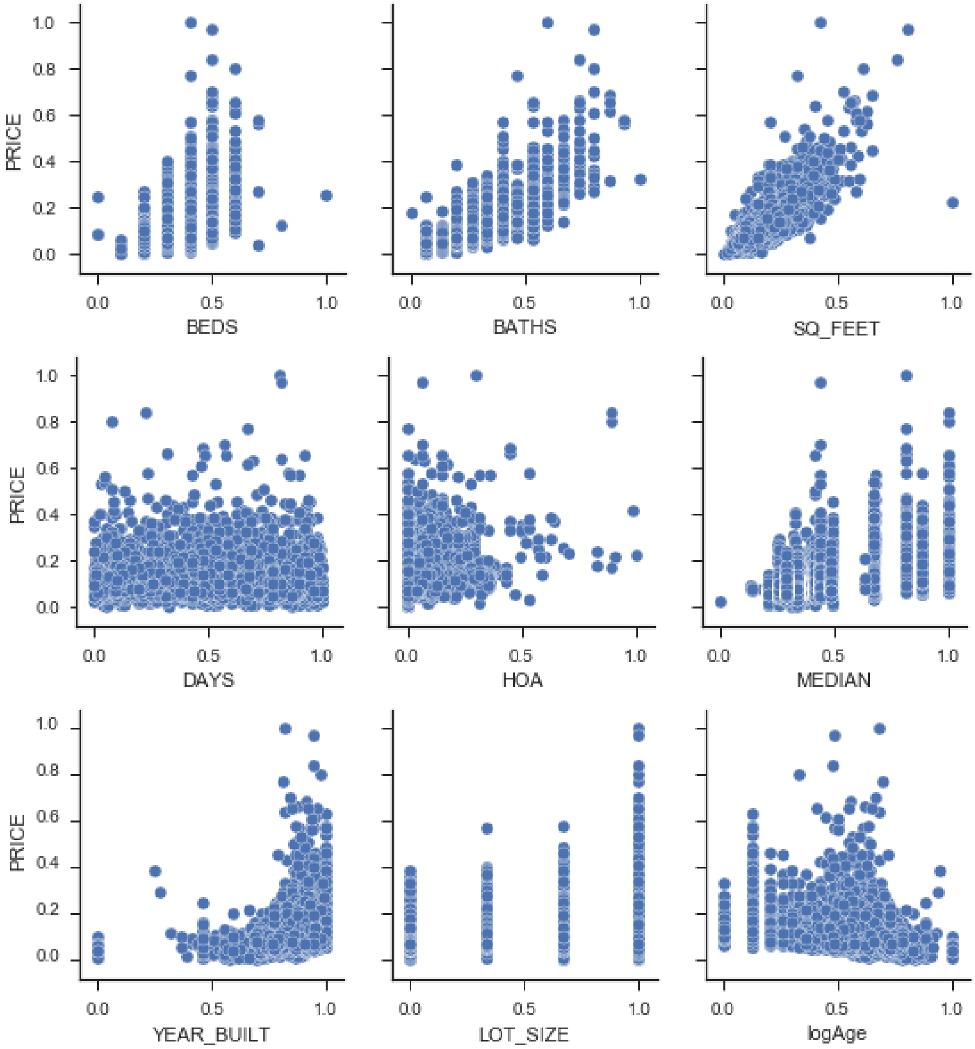


Figure 2. Scatterplots of Features vs. Sale Price

Many of the house prices register as outliers using the 1.5 x IQR (inter-quartile range) threshold. This was not unexpected. Figure 3 shows the extreme nature of some of the house sale points in the collected data, appearing as dots beyond the end of the whiskers of the boxplot. We found no obvious erroneous data entries for price. We performed an analysis on these extreme cases and determined that these values were typically under-predicted by our prediction models by a significant amount. This atypical behavior of extrema was also noted in the study by Mullainathan & Spiess (2017), where the upper and lower quintiles displayed different behavior in predictive models. These houses may represent a very different purchasing dynamic than a typical home. As a result, we decided to restrict our data set to 5853 houses that sold for $800,000 or less from the 5926 remaining houses.

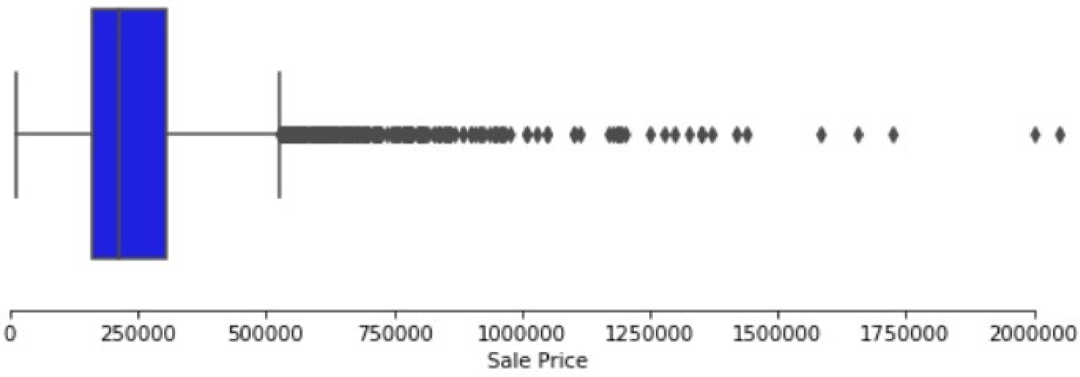


Figure 3. Boxplot of Sale Price Including All Collected Data

On the other side of the price spectrum, we discovered that prices of houses that sold for under $120,000 were significantly over-predicted by our algorithms. Upon inspection, many of these houses were sold at auction after repossession from a lending institution. This sale price does not seem to reflect the sales dynamic that most of our data points represent. We therefore also excluded 435 houses that sold for under $120,000, leaving 5418 houses in our data set.

Figure 4 shows boxplots of the standardized features. Price is also shown in a standardized format for comparison purposes. Many features show several extreme values on one side or the other. These extreme values on only one side will skew the distributions, but as such that does not make them bad predictors of price.

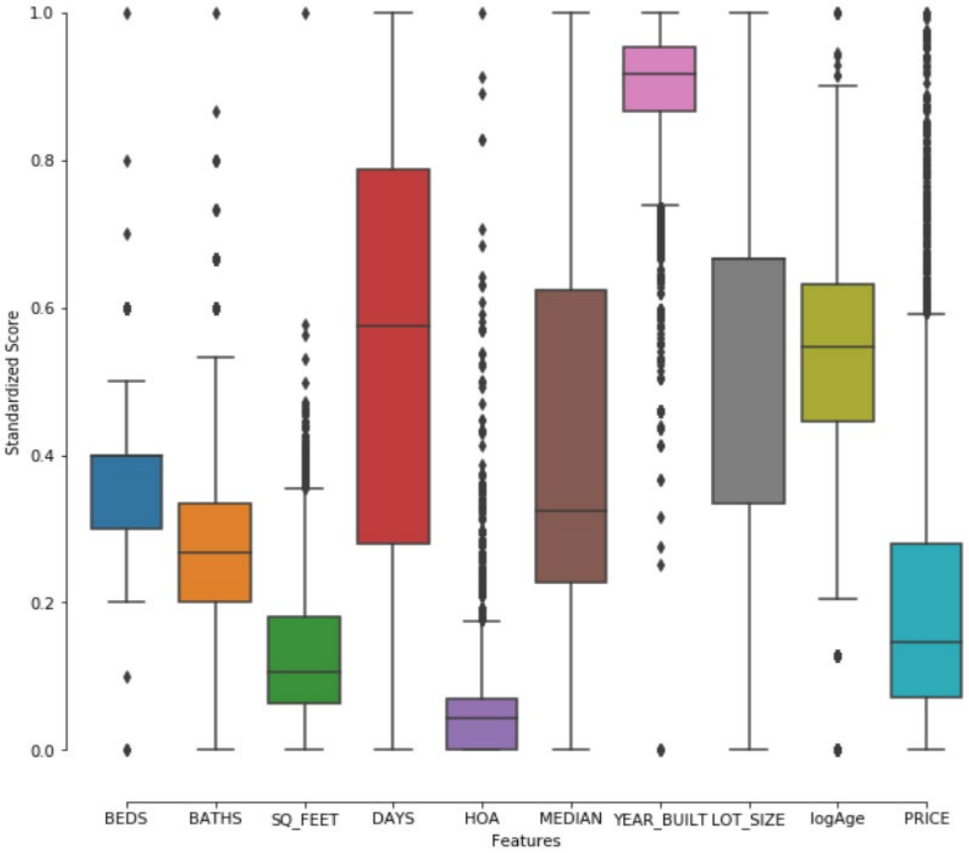


Figure 4. Boxplots of Standardized Features and Price

We can see in Figure 4 the distribution of the DAYS variable seems uniformly distributed over the range of 0 to 1. This distribution, coupled with the scatter plot of DAYS versus PRICE in Figure 2, suggests that DAYS will not be a strong candidate for a descriptive feature in a predictive model. Our modified feature LOT_SIZE, in which we used nominal values in place of actual lot sizes, shares this uniform shape, however the scatter plot of LOT_SIZE versus PRICE in Figure 2 suggests a linear relationship may be possible. We can use this knowledge when selecting the best feature subsets for our models.

Location as a Feature

One of the greater challenges in house price prediction is addressing the question of location. The same house in different neighborhoods could have dramatically different sale prices. To utilize location as a feature, we found the median house sale price in each zip code from publicly available historical data spanning the time window from which we collected data. We then added this median as a feature to our housing data set. The intention was to give the predictive algorithm some indication of the impact of the general geographic area on sale price, although ideally a finer-grained approach to location would be preferable. Many factors may impact the distribution of sale prices within a zip code, or even within a cluster of neighborhoods. Our goal in the creation of this feature was to capture an aggregation of these factors.

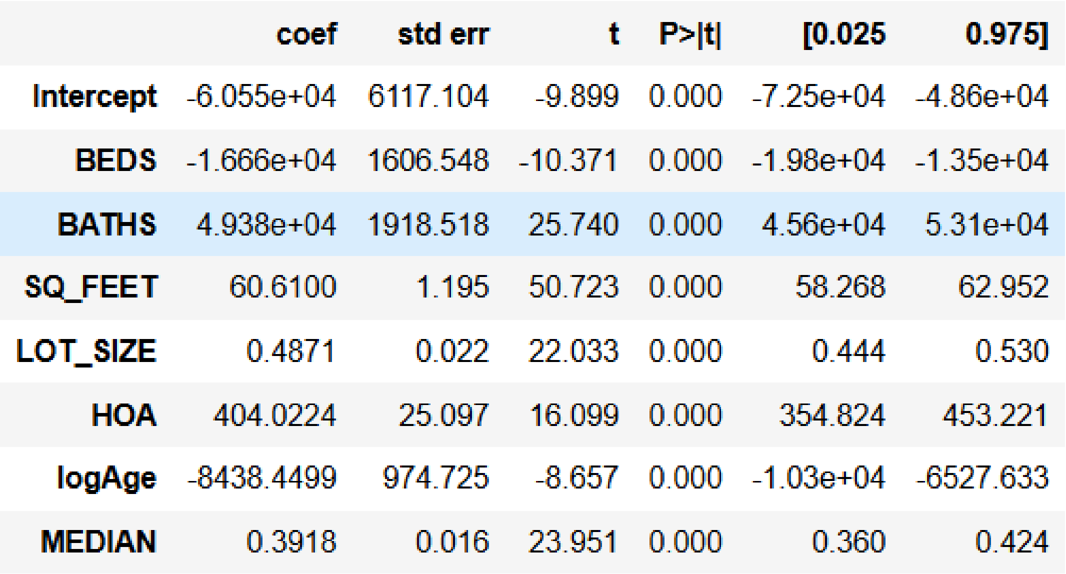
To capture the location variable on a finer scale than the zip code associated medians, we identified the five real estate properties in the data set closet to each house using latitude and longitude coordinates. This process provided five new features for each data record, each a sale price of a geographically similar (spatially close) property. A potential issue with this methodology is that the data set is quite limited in size. Given a very large database of homes, one could conclude the five nearest properties would exist in approximately the same neighborhood. In a data set only containing sales from the previous six months, it is possible that some of the nearest neighbors will not be in the same neighborhood or same zip code region.

The choice of specifically using five nearest neighbors stems from the work of Case et al. (2004) who used the five nearest neighbor residuals to represent spatial characteristics of houses. Bourassa et al. (2010) similarly applied ten nearest neighbors using residuals as the feature value. Both studies show additional information from spatially proximate houses adds value to some predictive algorithms.

During testing, the validation data's nearest neighbor features were created using a scan of the training data to find those five features, ensuring that only the training data were used to create new features for each validation record. This was important to ensure no data leakage took place from the training set. We followed the approach of hiding as much of the training data information from the testing data for all model training in this study.

Ordinary Least Squares Regression

We used the python library *statsmodels* to build the regression models. Some features had strong linear correlations with house sale price, and some did not. Using backward feature elimination, we removed the feature with the highest p-value (with respect to a significantly different than zero slope) iteratively until only features with p-values less than 0.05 remained (see Figure 5). These remaining features were used with the linear regression model. We then fine-tuned the feature set, by pruning features that met the criterion for remaining, but when removed resulted in better predictive models. The LONGITUDE feature was once such variable that displayed a statistically significant result but when the feature was removed the accuracy of the predictive model increased. Figure 5 shows the remaining feature set and resulting p-values after analysis and feature pruning.



	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-6.055e+04	6117.104	-9.899	0.000	-7.25e+04	-4.86e+04
BEDS	-1.666e+04	1606.548	-10.371	0.000	-1.98e+04	-1.35e+04
BATHS	4.938e+04	1918.518	25.740	0.000	4.56e+04	5.31e+04
SQ_FEET	60.6100	1.195	50.723	0.000	58.268	62.952
LOT_SIZE	0.4871	0.022	22.033	0.000	0.444	0.530
HOA	404.0224	25.097	16.099	0.000	354.824	453.221
logAge	-8438.4499	974.725	-8.657	0.000	-1.03e+04	-6527.633
MEDIAN	0.3918	0.016	23.951	0.000	0.360	0.424

Figure 5. Fit Statistics for the Final Linear Regression Model

Upon further analysis of Figure 2, there appears to be some degree of collinearity between the features. Collinearity is present when independent features are linearly related, which increases the variance of the estimates. It is expected that within housing data there will be a degree of collinearity between features. More living area may result in more bedrooms. More bedrooms may result in more bathrooms. Figure 6 displays a heatmap of correlations where we see many of our features are correlated.

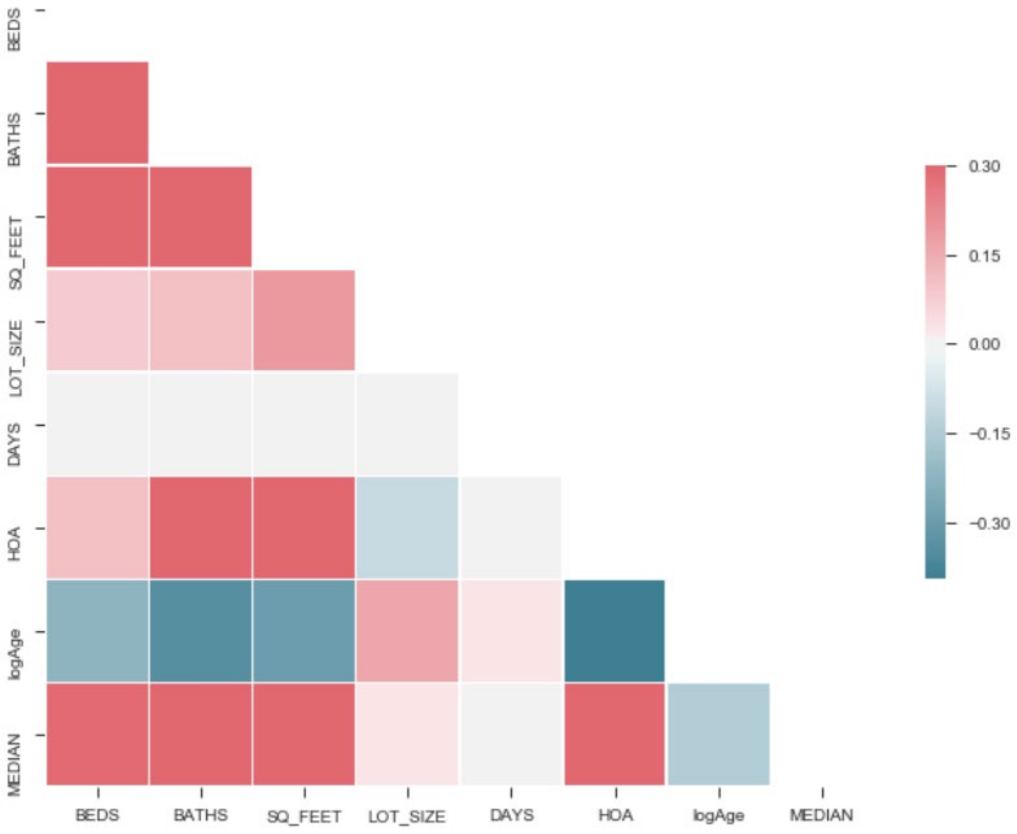


Figure 6. Heatmap of Final Feature Correlations

The descriptive data gathered for the houses in this dataset resulted in an OLS regression model with good accuracy (see Table 3). However, the limitation of this model was believed to come from the lack of fine-tuned location variables. The addition of the five nearest neighbor prices significantly improved the accuracy of this model, as over half of this model's predictions fell within 10% of the true sale price with these added features. Table 3 displays the accuracy and median absolute error rate comparison of the two models.

Table 3. Regression Model Performance

Model	Percent of Predictions within 10%	Median Absolute Error Rate
OLS	0.4478	0.1140
OLS with Nearest Neighbors	0.5321	0.0942

The accuracy and median absolute error rate of the OLS models provide a baseline value for our next predictive experiments. These results are comparable with previous studies that used linear regression techniques as house price prediction models, although these studies used different data sets. Case et al. found a median error rate of 0.1227 for their OLS model without location data, and 0.1131 for their local regression model (Case et al., 2004). Bourassa's best model reported 44% of predictions within 10% of the actual sale price (Bourassa et al., 2010).

Neural Network Models

While linear regression provides a standard approach for house price prediction, modern advances in the field of data science provide other options for this type of problem. We implemented a neural network to compare with the performance of the linear regression models. The learning capacity and number of layers needed to build a good predictive network required iterative testing of many possible parameters and architectures. The challenge in this process is to find a balance between accuracy in predictions and the ability to generalize to unseen data, known as the bias-variance tradeoff (Geman, Bienenstock, & Doursat, 1992). The neural network model that was ultimately chosen maximized the predictive accuracy while minimizing the amount of 'overfitting.' In early tests, it was simple to create a network to learn the training data structure, but much more difficult to create a model that could generalize.

The creation of the structure of the neural network was integral to our model's success. We used the 'RMSProp' optimizer after testing several alternatives, such as stochastic gradient descent. RMSProp tended to converge faster and provide better results than alternative optimizers. The loss function used to evaluate the models was 'mean squared error' and each fully connected layer used the rectified linear unit ('relu') activation function.

The process of iteratively selecting the best parameters for the model involved first creating a model that 'overfit' the data, then adding dropout to help the model generalize. Adding layers and nodes adds learning capacity to the neural network, and we slowly increased both until the model could reliably learn to predict house price from the numeric features in the training data. The use of dropout layers has been shown to help models generalize by randomly deactivating a percentage of neurons during each training pass of the data, thereby forcing the remaining neurons to learn better data representations (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). Adding dropout between the fully connected layers allowed the model to better generalize to unseen data. We started with small levels of dropout and slowly increased the aggressiveness of the dropout ratio until a balance was discovered between overall accuracy and validation set generalization. Figure 7 shows the final neural network model, including dropout levels.

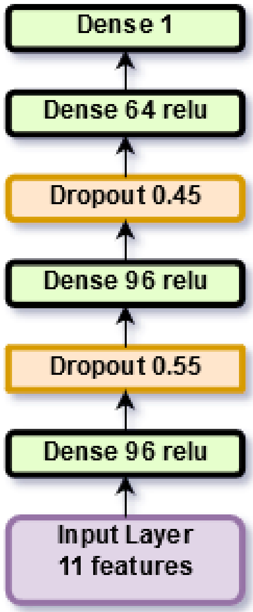


Figure 7. Neural Network Model Architecture

This neural network design is ‘deep’, meaning there is more than one hidden layer in the design. The purpose of the hidden layers is to find non-linear patterns from data beyond the linear relationships that were investigated in the previous section. We used all eleven available features as inputs to the network, including the spatial variables: latitude, longitude, and the median house price per zip code region. Additional hidden layers did increase the ability of the neural network to learn the training data; however, adding more than three hidden layers diminished the ability of the model to generalize to unseen data.

Figure 8 displays the training versus validation loss values for the final neural network models, using the eleven numeric features describing each house and the eleven numeric features appended with the five nearest neighbor sale prices. We trained the same neural network architecture a second time on this expanded feature set. Both graphs show the convergence of the validation loss, however the network with the nearest neighbors converges using fewer epochs.

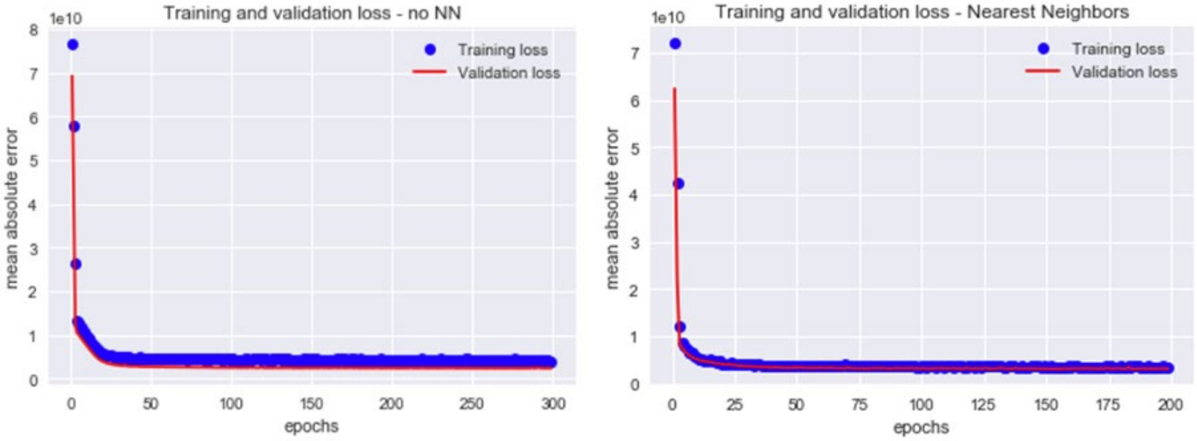


Figure 8. Neural Network Training and Validation Loss

The use of dropout as a tool to control overfitting can be observed in the validation versus training loss plots in Figure 8. As the network trains the dropout is applied, and a set number of random neurons are not trained during each epoch. In the validation testing mode however, each neuron is active for predictions on the validation data, resulting in the two converging plots. The training loss converges slowly, as the individual neurons take longer to train with the dropout layers applied. The red validation set lines in Figure 8 quickly take on a low loss value over a few epochs, whereas the blue training validation takes longer to converge upon a minimal loss value. The validation curves benefit almost immediately from the better-trained neurons and converge using fewer epochs. This shows that the model generalizes well to unseen data.

Other Predictive Models

We also implemented both a random forest regression model and a gradient boosting regression model using our descriptive features. We used a randomized search with cross-validation to explore the best parameters for each model. A grid search was implemented to fine tune the best parameters discovered from the randomized search. This two-stage approach was more efficient than using a grid search exclusively. The randomized search provides estimates of optimal parameters relatively quickly compared to a grid search, allowing us to search more of the parameter space. Once we had a target for optimal parameters, we tuned them with the much more computationally expensive grid search.

There are many possible parameters to optimize for a random forest regression algorithm. Parameters included in the optimization process for our random forest regression model are:

- Number of estimators
- Maximum depth in each tree

- Maximum features considered for splitting
- Minimum samples per split
- Minimum samples per leaf
- Bootstrap mode

This set of parameters was chosen in part due to research on hyperparameter tuning for the random forest algorithm (Bernard, Heutte, & Adam, 2007). Using the best hyperparameters provided by the optimization algorithm, we implemented the random forest regression model using the same eleven features from the neural network model, which included the spatial characteristics but initially not the nearest neighbors. We then appended the five nearest neighbors to the 11 features and fit the model a second time, using the same set of optimized parameters as before. This helped us compare the two models given only a difference in feature sets.

We used a similar approach to optimize a gradient boosting regression model. Applying the randomized search followed by a fine-tuned grid search, we explored the following parameter space for an optimal model:

- Number of estimators
- Maximum depth
- Maximum features
- Minimum samples per leaf
- Learning rate

This set of parameters was chosen in part due to research on hyperparameter tuning for the gradient boosting algorithm (Ganjisaffar, Caruana, & Lopes, 2011). Using the hyperparameters provided by the optimization algorithm, we once again used the eleven inputs with the best gradient boosting regression model. As with the random forest regression models, we then appended the five nearest neighbors and fit the model a second time using the same parameters. Tables 4 and 5 display the optimal parameter values, as determined by a grid search of the parameter space.

Table 4. Optimized Parameters for the Random Forest Regression Model

Parameter	Optimized Value
Maximum depth	20
Minimum samples per split	2
Number of estimators	500
Bootstrap mode	True
Minimum samples per leaf	2
Maximum features	auto

Table 5. Optimized Parameters for the Gradient Boosting Regression Model

Parameter	Optimized Value
Learning rate	0.01
Maximum depth	8
Minimum samples per leaf	30
Maximum features	30%
Number of estimators	1000

Results

Random forest regression and gradient boosting regression all saw performance slightly degrade when using the nearest neighbor features in addition to geospatial coordinates. It is possible that re-optimizing these models for the expanded feature sets would improve their performance, however using the same parameters the models did not improve. OLS, which did not have access to spatial coordinates, improved significantly when given the nearest neighbor data. The neural network also improved significantly with the added nearest neighbor information. The results from each model are included in Table 6.

Table 6. Comparison of Model Performance Using the Descriptive Feature Set

Model	Proportion of Predictions within 10%	Median Absolute Error Rate
OLS	0.4478	0.1140
OLS w/Nearest Neighbors	0.5321	0.0942
Neural Network	0.4806	0.1040
Neural Network w/Nearest Neighbors	0.5319	0.0930
Random Forest	0.6220	0.0760
Random Forest w/Nearest Neighbors	0.6183	0.0756
Gradient Boosting	0.6347	0.0758
Gradient Boosting w/Nearest Neighbors	0.6282	0.0788

A comparison of the results in Table 6 shows superior performance from both the random forest regression algorithm and the gradient boosting regression algorithm. These two algorithms were best able to predict price given both the linear and non-linear features available. The 0.0758 median error rate reported for the gradient boosting regression algorithms from this Indianapolis, Indiana metro area data is higher than Zillow Group's reported median error rate of 0.053 for the similar metro area of Cleveland, Ohio (see Table 2). However, the Indianapolis data set contains roughly 138 times fewer records, so the performance of the model seems competitive given the size of the data set and a small number (11) of available features. Zillow Group's feature count is proprietary and unpublished, so we are assuming they use more than 11 features as predictors given the vast amount of information they presumably maintain.

5.2 Combining Image Features with Descriptive Data

Images of houses represent a seemingly untapped resource for house price prediction. This section represents our approach for extracting characteristics from house images for use with predictive algorithms. We utilize pre-trained models to explore this idea, then integrate these new features with our best predictive models from the previous section.

Feature Extraction from Images

We used pre-built models to extract features from our house images. Each pre-built model in the Keras package has a pre-processing function to prepare the images for use in the model. We applied that function to our images and used them as input for the models. Using Keras and the pre-trained models for the extraction of features required the removal of the final (classification) layer of each model. This was achieved through setting the *include_top* parameter to *false* for the constructor of the model's object.

Several popular architectures are available within the Keras package, and we experimented with several as feature extraction algorithms. We chose the ResNet, VGG, and the Inception architecture families as our primary models, as they all use ImageNet data for pre-training and therefore have some domain overlap with our housing images. Also, they are relatively compact architectures (albeit with millions of trainable weights) and have proven successful in image classification competitions (Krizhevsky et al., 2017).

Table 7 shows a comparison of the number of extracted features from each of our chosen models. ResNet50 returns 2048 features, which is considerably smaller than the VGG or Inception models. Adding over 2000 new features however is not ideal, as this would result in a large increase in the dimensionality of the feature set. This "Curse of Dimensionality" implies that as the number of features increases past a certain point, accuracy of machine learning models tend to decrease (Trunk, 1979). We applied principal component analysis (PCA) to the resulting features to reduce the dimensionality of the extracted feature set. Reducing the ResNet features to 50 principal components preserved approximately 72% of the explained variation from the larger feature set, compared with 37% for VGG and 44% for Inception. We chose this level of reduction as a compromise between explained variance and feature set dimensionality after iterative testing of different combinations of model type and reduction level. Tests using greater number of features retrieved from principal component analysis showed diminishing returns. As the number of PCA generated features used as input increased, the accuracy of the model tended to decrease from the increasing complexity of the feature space. This drove our initial choice of 50 principal components, which in turn drove our decision to use ResNet as our primary extraction technique. The remaining two models required a much higher number of features to be returned to match the explained variance benchmark set by ResNet with a PCA reduction to 50 features (see Table 7).

Table 7. Number of Parameters and Extracted Features by Pre-Trained Model.

Model	Number of Extracted Features	Number of Parameters	Required Feature Count: 72% Explained Variation
VGG16	27,648	138,357,544	538
ResNet50	2,048	25,636,712	50
InceptionResNetV2	49,152	55,873,736	486

The new feature set (11 numeric features appended with 50 image features) was optimized with the gradient boosting regression algorithm and evaluated as before. We chose the gradient boosting regression model as it achieved the highest performance with the descriptive data and outperformed the random forest regression model and neural network in initial tests with the combined feature set.

The Gradient Boosting Regression Approach

The best performance achieved by the optimized gradient boosting regression model using combined feature sets is shown in Table 8. The performance of the models with 50 additional features did not show a significant improvement over the original gradient boosting regression algorithm.

Table 8. Accuracy and Median Error Rate of Combined Feature Models

Model	Proportion of Predictions within 10%	Median Absolute Error Rate
Gradient Boosting: 0 Additional Features	0.6347	0.0758
Gradient Boosting: 50 Additional Features	0.6251	0.0752
Gradient Boosting: 2048 Additional Features	0.5840	0.0841
Gradient Boosting: Selected 3 Additional Features	0.6388	0.0728

We discovered that using three of these new PCA generated features did improve our model. The key to finding features that added value to our models was discovered when analyzing the individual principal components. Figure 9 shows scatter plots between the first four principal components and sale price. Three of the four plots show a slight linear pattern with price. These three principal components with linear relationships with price were appended to the 11 existing descriptive features to create the best model in this study, referenced above in Table 8. The remaining unused principal components had distributions like PCA2, which showed no linear relationship with price (see Figure 9).

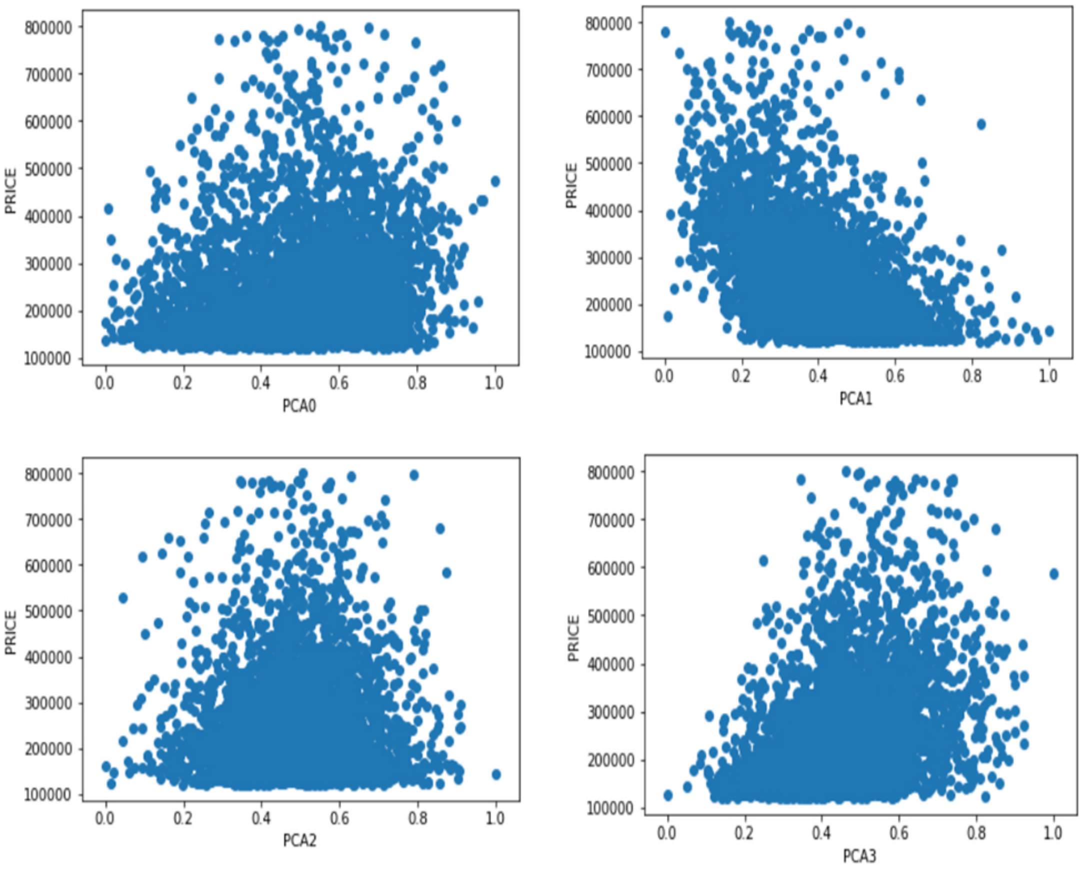


Figure 9. Top Principal Components from ResNet50 Features vs. Price

Figure 10 confirms the linear relationship between the three selected principal components and sale price using a correlation heatmap. The correlation between each of the three components and price is evident, matching the scatterplots from Figure 9. Note the lack of correlation between the principal components themselves, as we would expect in an orthogonal transformation (Shlens, 2014).

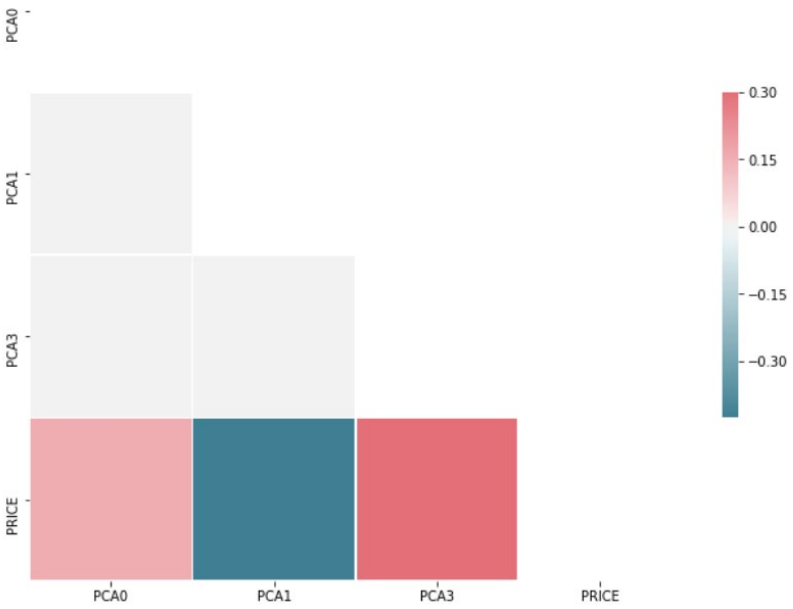


Figure 10. Heatmap of Selected Three Principal Components and Price

The difference between the evaluation metric values for the gradient boosting regression algorithm with and without the three additional features is very small, and possibly attributable to chance. To verify that there is a significant difference between the outcomes of the two models, we ran each 10-fold cross validation 10 times and performed a one-sided 2-sample t-test for the difference between the resulting means. We chose this method in place of a paired-t test using paired folds due to problems with the independence condition when using a k-fold algorithm. We also considered running a 5x2cv hypothesis test based on the work of Dietterich (1998) however this test is more suited to comparing different types of classifiers on the same feature set. Table 9 displays the results of the 10 runs of 10-fold cross validation, and the resulting t-scores and p-values from the significance tests.

Table 9. Mean Results of 10 Runs of 10-Fold Cross Validation

Model	Number of Features Used	Proportion of Predictions within 10%	Median Absolute Error Rate
Gradient Boosting Regression	11	0.6347	0.0728
Gradient Boosting Regression w/ Three Selected Features	14	0.6388	0.0724
t-statistic (df = 18)		-3.240	2.0878
p-value		$P(t < -3.240) = 0.0023$	$P(t > 2.0878) = 0.0257$

The p-value for the proportion of predictions within 10% of the actual sale price shows significant result at a 1% level of significance. This implies that given a null hypothesis where the two sets of data have equal means, the probability of achieving a result as extreme or more extreme than our difference in means by chance is less than 1%. This evidence suggests that predictive accuracy using the 10% accuracy metric is higher for the gradient boosting regression model with the additional 3 principal components than using the gradient boosting regression model with only the descriptive features.

Similarly, we see that the difference in means for the median error rate listed in Table 9 is also statistically significant at a 5% level of significance. This implies there is evidence that the median absolute error rate of the gradient boosting regression with added features is lower than the model that did not use the three additional features. The evidence is not as strong for this metric. However, the two tests viewed together present a case that the three additional features extracted from the images do add value to our best predictive algorithm. On a more basic level of comparison, for the model with the added features the accuracy score was higher for 8 of the 10 mean scores from 10-fold cross validation, and the median absolute error rate was lower in 8 of the 10 mean rates from the 10-fold cross validation. Appendix 1 contains the resulting data tables.

The Neural Network Approach

We next built a neural network architecture for the new combined feature set. Initial attempts at designing a neural network architecture that could utilize the added features from the image data were unsuccessful at improving the predictive accuracy over using only the descriptive features. Our initial attempts to build this network appended the 50 PCA reduced ResNet features to the original 11 descriptive features. The architectures tested for the combined 61 features were insufficient for the task and failed to create a better predictive model than the descriptive features alone. We decided instead to treat the two types of data (descriptive features with location data and extracted image features) differently inside the network itself, applying differing numbers of layers and neurons per layer to the two data types.

The creation of a neural network architecture that could leverage the new features required a moderately complex design. We created two paths through our network, one for the eleven descriptive features and one for the 50 image features. The two paths then merged into a fully connected layer shown in Figure 11, the architecture schematic. This design allows for different levels of adjustment between the two types of data. For example, the image data responded well to aggressive levels of dropout with larger hidden layers (containing more neurons), whereas the descriptive data responded well to moderate levels of dropout and slightly smaller hidden layers. We will refer to this model as the custom merging neural network model (CMNN) from this point forward.

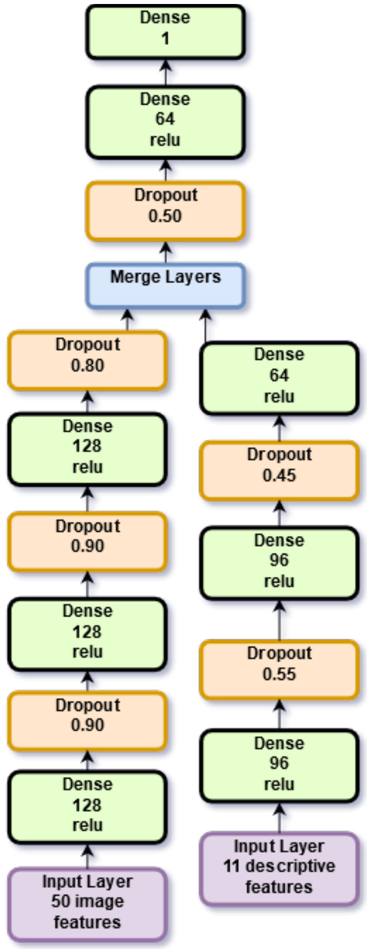


Figure 11. Custom Merged Neural Network (CMNN) Design

The CMNN gained significant predictive accuracy (see Table 10) over the features-only neural network model with the addition of the image features. This model was able to make use of the image features and improve the resulting predictions.

Table 10. Performance of the Custom Merged Neural Network

Model	Proportion of Predictions within 10%	Median Absolute Error Rate
Descriptive Features Neural Network	0.4806	0.1040
Custom Merging Neural Network w/PCA50	0.5200	0.0930

Once again, the use of dropout layers between the fully connected layers can be seen in the training versus validation loss plot in Figure 12. The validation loss (in red) tends to level off very quickly and makes minor adjustments over the epochs, whereas the training loss (in blue) converges more slowly. The dropout layers are forcing the individual neurons that are not randomly dropped during each training epoch to learn better representations of the data. Although this architecture improved the performance over the baseline neural network, the gradient boosting approach was still better.

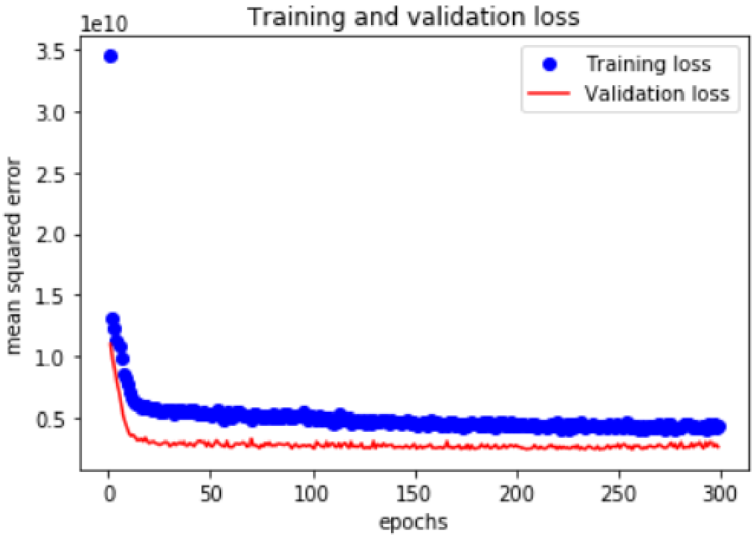


Figure 12. CMNN Training and Validation Loss

5.3 A Binary Classifier for Curb Appeal

This section describes an approach we developed to use the image feature extractor to predict whether our best house price prediction models would over- or under-predict the prices. The overall pipeline is depicted in Figure 13. The remainder of the section will describe the details and give the results.

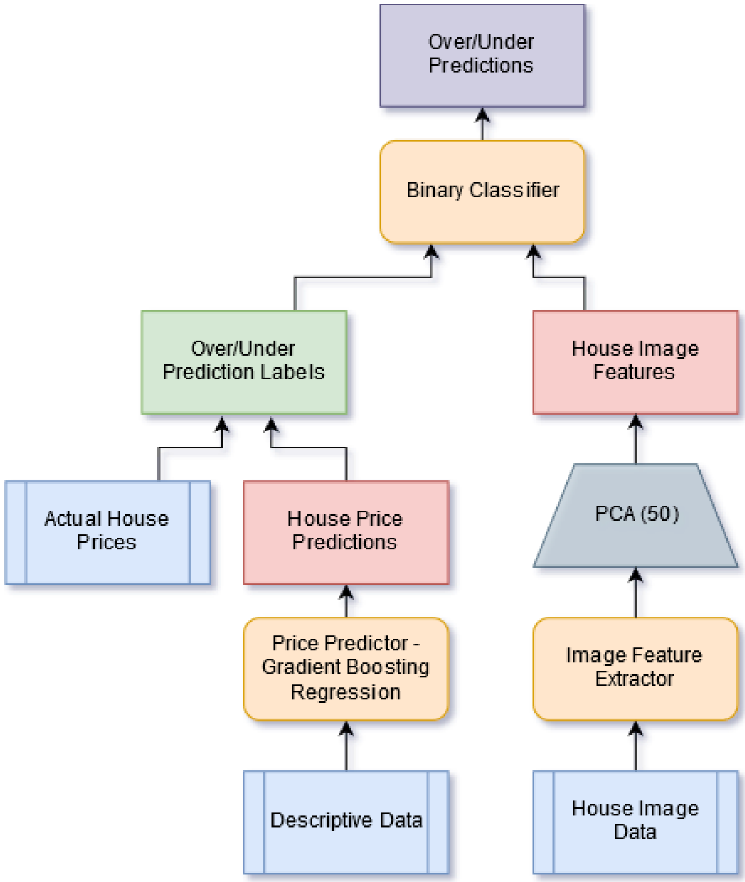


Figure 13. Binary Classifier System Diagram.

As in previous experiments, we chose the ResNet model pre-trained with the ImageNet weights for an image feature extractor. The images were resized to 224x224 pixels, preprocessed, then input into the ResNet50 model in Keras. This resulted in 2048 extracted features. PCA was applied to these extracted features as in previous experiments, creating a reduced feature set of 50 principal components. The descriptive feature set will not be provided to our classifier, so reduction to the first three or four principal components might rob the classifier of useful information. Conversely, providing all 2048 features to the classifier creates a much higher-dimensional feature space. We chose the first 50 principal components after testing multiple levels of PCA. The results of those tests are shown in Table 11.

Table 11. Performance of Binary Classifier vs. PCA Level on Image Features

PCA Level (number of principal components)	AUC Score	Margin of Error
4	0.5409	0.0296
25	0.5556	0.0338
50	0.5681	0.0343
100	0.5528	0.0294
2048 (no PCA reduction)	0.5536	0.0381

We used our best model for predicting sale price from descriptive features, the gradient boosting regression algorithm, to predict prices for each house in the data set. These predictions were compared to the actual sale prices and then each house was labeled as an under-prediction (Class 0) or an over-prediction (Class 1). An over-prediction implies the curb appeal of the house is low, correlating with the under-performance of the sale price versus the prediction. An under-prediction implies the curb appeal is high, correlating with the over-performance of the sale price versus the prediction. We then created a binary classifier to learn the corresponding over-prediction or under-prediction class labels directly from the house images. Once trained, the classifier predicted over-prediction or under-prediction from the house images.

As curb appeal is only one part of a larger set of decisions in purchasing a home, the expected performance of the classifier is limited. Our assumptions for this section imply that a buyer's visual reaction to a home plays only a part in the decision of what to pay for a home.

Figure 14 displays a comparison of receiver operating characteristic (ROC) curves for the binary curb-appeal classifier over ten folds of the data. The algorithm evaluates the image features and decides about whether the house will sell for more or for less than its predicted sale price. As we see in Figure 14, the model displays the ability to predict over-/under-prediction on sale price better than random guessing.

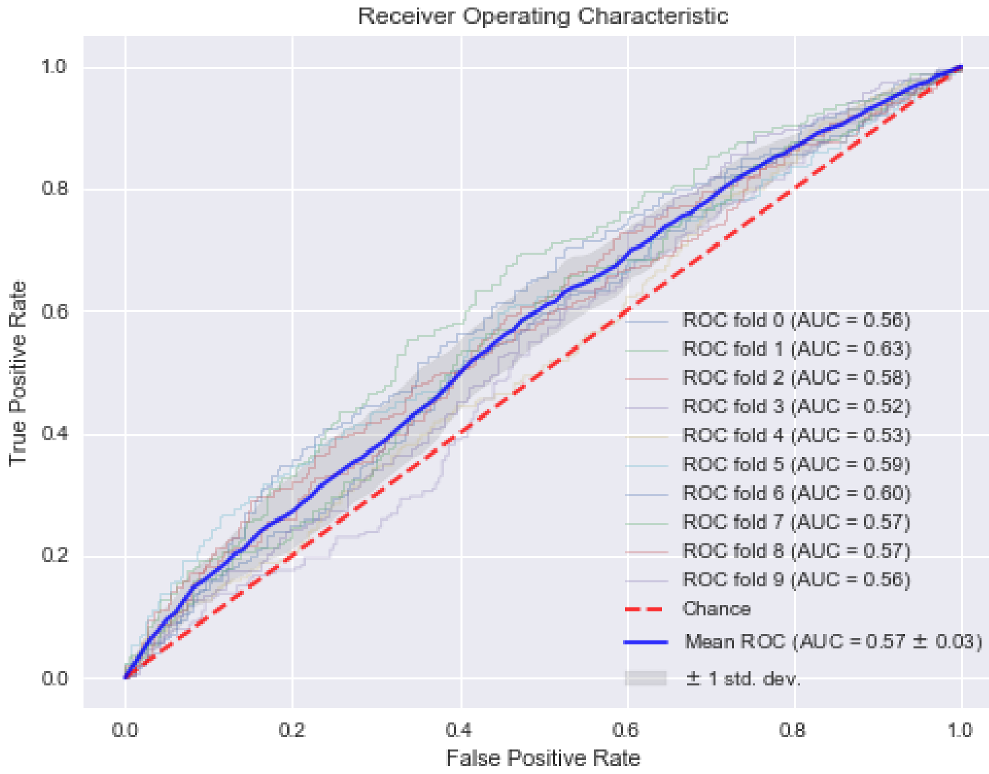


Figure 14. ROC for Binary Curb-Appeal Classifier

Clearly there are other factors at work in the sale price of a home, such as the buyer's ability to buy, the seller's willingness to sell, and the location of the home. As a result, we would not expect this classifier to have a high degree of accuracy. However, this classifier leverages the image of the home to make a prediction and does so to a degree that suggests the algorithm can learn this component of a house value decision.

Figure 15 shows a sample of six houses in a validation set with among the highest probabilities of being correctly categorized as under-predictions. The probability was reported by the algorithm. We chose the six representatives based upon human-level comparable characteristics. There are several similarities in these six photos. Each house has a large front yard, mature trees, and well-groomed landscaping. There are many visible windows and many triangular points along the roofline. The frontage material appears to be brick on several houses, either red or tan in color.

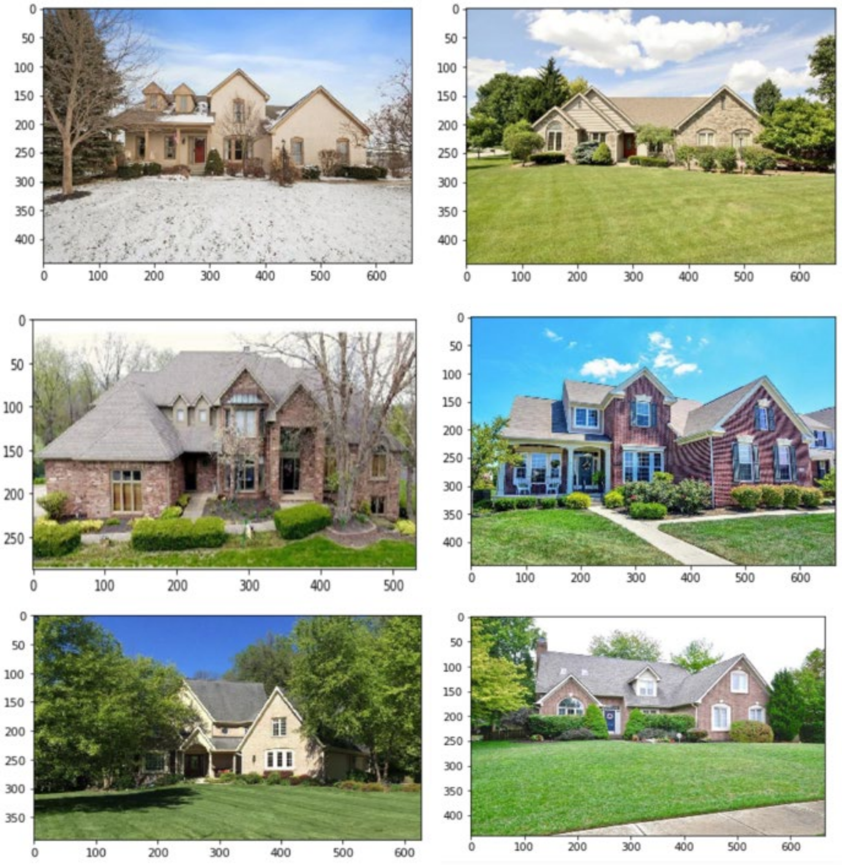


Figure 15. Best Under-Predictions for “Good” Curb Appeal

In Figure 16 we see six houses among the most likely to be over-predicted, based on probabilities reported by the algorithm. Again, we chose the six representatives based upon human-level visual comparisons. Several of these houses all share a similar style: A large front garage door that dominates the curbside view of the home. There are fewer visible windows and a small number of triangular rooftop peaks. The siding appears to by vinyl on several houses and there are few trees present. The trees that are present are small and not yet mature.

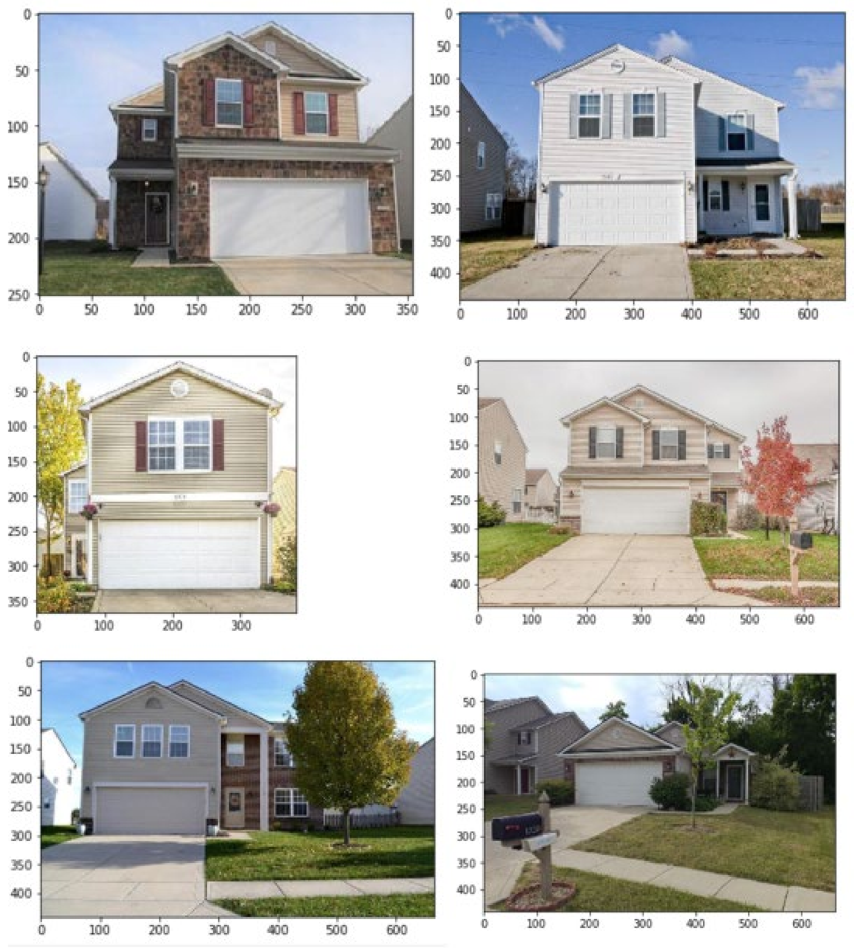


Figure 16. Best Over-Predictions for "Poor" Curb Appeal

A potentially troublesome issue with this classifier stems from an observation found earlier in the paper: the residuals produced by the predictive regression algorithms tend to under-predict houses that are relatively expensive and over-predict houses that are relatively low-cost. In removing the extreme cases we have diminished this impact of this potential bias, but it is unlikely we have removed it completely from the estimators. This bias may have the impact of teaching the classifier to indirectly classify based on the price of the home, and not the intended goal of classifying based on curb-appeal.

With this issue in mind, we can still observe the ability of the classifier to make predictions based upon extracted image features. The potential bias in the system may change what the classifier is learning, but not the classifier's ability to learn structure from image data. We therefore view this experiment as a limited success, presenting a model that needs further study and exploration.

Chapter 6

Discussion

Predicting the sale price of a house is a difficult task. Identifying factors and algorithms that can improve these predictions are critical components for building a house price prediction model. This study explores the field of house price prediction, incorporating deep learning to leverage a largely untapped source of new predictive features. This chapter summarizes discoveries, explains motivation, and answers the critical questions about house price prediction posed by this study.

Given a limited set of numerical features and location information, how well can algorithms predict the sale price of a house? We discovered that a gradient boosting regression algorithm outperformed other popular prediction algorithms with respect to our data. Although the comparison of results from different studies is not perfect due to differences in geography, housing markets, and time, we found that our house price predictions were competitive with predictions from prior studies. Our results were similar to work from Case et al. (2004), and superior to results from Kummerow (2000), and Bourassa (2010). Given a set of numerical features describing a house, it is possible to reasonably predict the sale price of a house, although there are other factors that impact sale price beyond numerical features alone.

As with Case et al. (2004), the use of location variables proved important for capturing the best estimates for house prices. The use of the median house price in each zip code region captured local area characteristics and provided the predictive models additional information about geo-spatial differences in price. Attempts at normalizing house price (adjusting the price up or down based on location) were not as successful at improving predictions as providing these medians. We subsequently chose this approach over capturing many individual neighborhood descriptive features. That is, instead of adding dozens of potential descriptors we used just median house price. This kept our dimensionality low and still captured the spatial region data. Coupled with latitude and longitude coordinates, this spatial feature improved our models' overall predictive accuracy for house price.

The use of only latitude and longitude coordinates for calculating nearest neighbors for each house requires further study. Initial models attempted to capture more variables for the Euclidean distance calculation of similarity, including living area and number of bedrooms. Hoping to create an algorithm that not only found a spatially proximate house but also one with similar characteristics, we included these numerical features in the distance algorithm. These additional features should have created better, more similar representations of any given house, but in experimental testing the houses in closer proximity gave us better results for our predictive models.

Model type plays a large role in house price prediction. Our OLS and deep neural network models were unable to keep pace with the random forest regression model and gradient boosting regression model. The random forest regression and gradient boosting regression algorithms performed well given a limited number of numerical features, zip code level spatial information, and latitude and longitude geo-spatial coordinates. A regional approach with finer granularity than zip codes is likely to further improve these algorithms with house price medians. Future work should include a spatial analysis that targets house price medians for individual neighborhoods, perhaps using a clustering technique to create homogeneous neighborhoods.

Filtering the numeric data contributed to the success of the prediction models. We removed the highest and lowest priced homes from our data to create a sample of houses that better reflected a typical house sale transaction. This filtering process excluded 8.6% of our available houses from the sample and limited the range of houses for which our model was trained. These restrictions did improve the accuracy of the models for what we considered typical sales. Another possible path in our data cleaning process might have been a log transformation of the house sale price. Because of the skewed nature of the house price distribution, this transformation might have allowed us to retain the excluded higher priced extremes, making our model more inclusive for a wider range of homes. Future work could include an analysis of possible transformations of price and the impact of these transformations on the price prediction models.

We chose to re-randomize our cross-validation folds for each experimental test. The alternative, using the same ten sets for each experiment, could have simplified the testing of our models. We decided that re-randomization would provide completely independent evaluations from model to model, and this independence outweighed the reduced testing complexity using the other approach. To avoid any information about the training data splits leaking into the testing data splits during cross-validation, we required new nearest neighbors be created from the training data for every fold in our 10-fold cross validation experiments.

Can deep learning be used to improve house price prediction algorithms? Shin et al. (2016) and Oquab et al. (2014) suggest that using transfer learning, new predictive features can be created from images. We were able to show that using pre-trained deep architectures and principal component analysis, features can be extracted from images of houses and used to improve house price predictions. This feature extraction process allows the deep network to extract the features it deems useful rather than seeking out specific features engineered by humans (LeCun, Bengio, & Hinton, 2015), and therefore may provide features that humans might not think to include. Our process of feature extraction demonstrates that useful predictive features exist within images and appending these features to descriptive and spatial data represents a new approach to real estate valuation model-building.

As a possible solution to the question of feature extraction, we built a convolutional autoencoder network designed to create a new representation of each image. An autoencoder is a self-trained neural network in which high-dimensional data, such as our images, are condensed to a low-dimensional latent space and then reconstructed again using the input images as the targeted outputs (Lange & Riedmiller, 2010). In short, an image is encoded using convolutional and pooling layers to a low-dimensional representation, and then the low-dimensional representation is decoded to reconstruct the original image using another series of convolutional and up-sampling layers. In training the encoding and decoding, the autoencoder creates the underlying latent structure (the “bottle-neck”) of the images. We used this latent encoded structure as a feature set, much like the resulting feature set from the pre-trained ResNet model. Although this process was successful in learning a set of latent features, these features did not improve the performance of our existing models. However, the process was instructive and provided insight into the feature extraction and evaluation process that we later used with the ResNet pre-trained models.

Can we teach a computer to learn curb-appeal? The approach for answering this question stemmed from a belief that there are many variables unaccounted for by typical house price prediction models. Reed (2016) attempted to capture some of these unknown variables related to the human agents involved in the transaction in the hopes of filling in some of the missing pieces to the puzzle. Our work also attempts to fill in some of the missing pieces from the visual realm. Using features extracted from our pre-trained deep learning network we were able to build a classifier that can identify houses that sold for more than their predictions or less than their predictions based only on the images themselves.

Our model learned visual patterns from the images to predict whether the estimated prices were over- or under-valued. However, our binary classifier does not account for predictions that were very close to the actual sale price. For example, houses that were over-predicted by one dollar were placed in a different class than houses that were under-predicted by one dollar. This placement potentially undermined our classifier's accuracy, and upon reflection we could have added a buffer between our over-prediction and under-prediction classes. By excluding predictions that were very close to the actual sale price, we would anticipate better performance from our classifier.

Chapter 7

Conclusions

In this study, we created a prediction algorithm using machine learning techniques that predicted the sale price of homes. We accomplished this using a small set of available features taken from approximately 6000 listings available online from the Indianapolis, Indiana metro area. The impact on sale price for the location of a house was also explored. We demonstrated that spatial information provided to our predictive algorithm was important in making accurate price predictions. We built a simple and accurate model, making the most of limited available data. Our best model created predictions of house price that were accurate within 10% of the actual sale price over 63% of the time, substantially surpassing the threshold used by The Federal Home Loan Mortgage Corporation (50%) for evaluating potential house price prediction models.

We then turned our attention to making our models better by integrating house images with our input data. House images contain data that can potentially help a predictive algorithm estimate the sales price of a house. We showed that image data can be effectively leveraged to improve price prediction models. We achieved this without using specific image feature engineering, but instead relied on learned deep network representations. The measured effect was small, but present.

In addition to using these deep network representations of images to predict house price, we also explored whether a machine learning algorithm can be trained to see value in a way that reflects human behavior. Our curb-appeal classifier successfully differentiated between over-prediction and under-prediction using the images. The accuracy was modest but likely limited by accurate predictors. This is an area that is worth further exploration.

To our knowledge, little research exists in using images in house price valuation techniques. Through these experiments we conclude that deep network representations of images coupled with transfer learning, can be used to improve real estate pricing models. In particular, the ResNet architecture was successful because of an overlapping domain between the ImageNet images and our housing images. Our contribution to the field of house price prediction is a blueprint that may help real estate valuation practitioners utilize new sources of data to improve existing house price prediction models. To help others continue this research, we have provided our code base, housing data, and image files at https://github.com/jokintz/house_price_preds.

This thesis shows that images can provide features that improve already strong predictive models for house price. Using pre-built models and PCA, we have demonstrated an approach to this improvement. Furthermore, this work presents a scenario in which a machine learning algorithm can predict the nature of a house sale from only the image of the house. These elements of house price prediction represent a stepping stone into what may be possible when using deep learning and convolutional networks with visual data representations.

Appendix 1.

Data Tables

The following table represents the results of 10 trials of 10-fold cross validation evaluating the performance metrics of two feature sets. Gradient boosting regression using only descriptive features is contrasted with gradient boosting regression using descriptive features and three selected principal components.

Table 12. Table of Results Comparing Feature Sets for Gradient Boosting Regression

Trial Number	Grad. Boost	Grad. Boost w/ PCA features	Grad. Boost	Grad. Boost w/PCA features
	Accuracy	Accuracy	Median Error Rate	Median Error Rate
1	0.630060120	0.637875752	0.073516528	0.072262875
2	0.630460922	0.636673347	0.073890932	0.071688675
3	0.636873747	0.636472946	0.072183551	0.072898402
4	0.635671343	0.642885772	0.073291908	0.072280598
5	0.631462926	0.640480962	0.073315571	0.072381586
6	0.640480962	0.637074148	0.073016144	0.0727656
7	0.635470942	0.63747495	0.07214741	0.071986248
8	0.635671343	0.638076152	0.072429074	0.072303136
9	0.637074148	0.639278557	0.072342938	0.072746339
10	0.634268537	0.641683367	0.072273919	0.072221728
Mean	0.634749499	0.638797595	0.072840798	0.072353519
Variance	1.07112E-05	4.90E-06	4.07838E-07	1.37E-07

References

Abbas, Q., Ibrahim, M. E. A., & Jaffar, M. A. (2018). A comprehensive review of recent advances on deep vision systems. *Artificial Intelligence Review*. https://doi.org/10.1007/s10462-018-9633-3

Bernard, S., Heutte, L., & Adam, S. (2007). Using Random Forests for Handwritten Digit Recognition. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on* (Vol. 2, pp. 1043–1047). IEEE.

Bourassa, S., Cantoni, E., & Hoesli, M. (2010). Predicting House Prices with Spatial Dependence: A Comparison of Alternative Methods. *Journal of Real Estate Research*, *32*(2), 139–159. https://doi.org/10.5555/rees.32.2.l154237243831 57x

Case, B., Clapp, J., Dubin, R., & Rodriguez, M. (2004). Modeling Spatial and Temporal House Price Patterns: A Comparison of Four Models. *The Journal of Real Estate Finance and Economics*, *29*(2), 167–191. https://doi.org/10.1023/B:REAL.0000035309.60607.53

Castellanos, S. (2016, November 11). Zillow Develops Neural Network to ‘See’ Like a House Hunter. Retrieved November 21, 2018, from https://blogs.wsj.com/cio/2016/11/11/zillow-develops-neural-network-to-see-like-a-home-buyer/

Dietterich, T. G. (1998). Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, *10*(7), 1895–1923. https://doi.org/10.1162/089976698300017197

Dubin, R. A. (1998). Spatial Autocorrelation: A Primer. *Journal of Housing Economics*, *7*(4), 304–327. https://doi.org/10.1006/jhec.1998.0236

Feng, Y., & Jones, K. (2015). Comparing multilevel modelling and artificial neural networks in house price prediction. In *2015 2nd IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services (ICSDM)* (pp. 108–114). https://doi.org/10.1109/ICSDM.2015.7298035

Ganjisaffar, Y., Caruana, R., & Lopes, C. (2011). Bagging gradient-boosted trees for high precision, low variance ranking models. In *Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval* (pp. 85–94). ACM.

Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, *4*(1), 1–58. https://doi.org/10.1162/neco.1992.4.1.1

Goodman, J. (1993). A Housing Market Matching Model of the Seasonality in Geographic Mobility. *Journal of Real Estate Research*, *8*(1), 117–137. https://doi.org/10.5555/rees.8.1.a216163200411364

Harris, D. (2018, April 28). Zillow analytics boss on making better predictions thanks to data and deep learning. Retrieved November 21, 2018, from https://architecht.io/zillow-analytics-boss-on-making-better-predictions-thanks-to-data-and-deep-learning-163b2c90100c

Harrison, D. J., & Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *J. Environ. Econ. Manage.*, *5*(1).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

Kummerow, M. (2000). Theory for Real Estate Valuation: An Alternative Way to Teach Real Estate Price Estimation Methods. Department of Property Studies. Curtin University.

Lancaster, K. J. (1966). A New Approach to Consumer Theory. *Journal of Political Economy*, *74*(2), 132–157. https://doi.org/10.1086/259131

Lange, S., & Riedmiller, M. (2010). Deep auto-encoder neural networks in reinforcement learning. In *The 2010 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). https://doi.org/10.1109/IJCNN.2010.5596468

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature; London*, *521*(7553), 436–444.

Limsombunchao, V. (2004). House price prediction: Hedonic price model vs.artificial neural network. *Proceedings of the Tenth Annual Conference of the New Zealand Agricultural and Resource Economics Society (Inc)*. Retrieved from https://researcharchive.lincoln.ac.nz/handle/10182/5198

Mullainathan, S., & Spiess, J. (2017). Machine Learning: An Applied Econometric Approach. *Journal of Economic Perspectives*, *31*(2), 87–106. https://doi.org/10.1257/jep.31.2.87

Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks (pp. 1717–1724). Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Retrieved from https://www.cv-foundation.org/openaccess/content_cvpr_2014/html/Oquab_Learning_and_Transferring_2014_CVPR_paper.html

Reed, R. (2016). The relationship between house prices and demographic variables. *International Journal of Housing Markets and Analysis*, *9*(4), 520–537.

Rosen, S. (1974). Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition. *Journal of Political Economy*, *82*(1), 34–55.

Shin, H.-C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., … Summers, R. M. (2016). Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Transactions on Medical Imaging*, *35*(5), 1285–1298. https://doi.org/10.1109/TMI.2016.2528162

Shlens, J. (2014). A Tutorial on Principal Component Analysis. *CoRR*, *abs/1404.1100*. Retrieved from http://arxiv.org/abs/1404.1100

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv:1409.1556 [Cs]*. Retrieved from http://arxiv.org/abs/1409.1556

Sommervoll, D. E., & Sommervoll, Å. (2018). *Learning from man or machine: Spatial aggregation and house price prediction* (CLTS Working Papers No. 4/18).

Norwegian University of Life Sciences, Centre for Land Tenure Studies. Retrieved from https://ideas.repec.org/p/hhs/nlsclt/2018_004.html

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.*, *15*(1), 1929–1958.

Trunk, G. V. (1979). A Problem of Dimensionality: A Simple Example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-1*(3), 306–307. https://doi.org/10.1109/TPAMI.1979.4766926

Weessies, K. (n.d.). LibGuides: Finding Census Tract Data: About Census Tracts. Retrieved January 6, 2019, from //libguides.lib.msu.edu/c.php?g=96120&p=625755

Wold, S., Ruhe, A., Wold, H., & Dunn, I. (1984). The Collinearity Problem in Linear Regression. The Partial Least Squares (PLS) Approach to Generalized Inverses. *SIAM Journal on Scientific and Statistical Computing*, *5*(3), 735–743. https://doi.org/10.1137/0905052

Zhang, K., Wu, Q., Liu, A., & Meng, X. (2018). Can Deep Learning Identify Tomato Leaf Disease? *Advances in Multimedia*, *2018*, 10.

Zillow. (2018). What is a Zestimate? Zillow’s Zestimate Accuracy. Retrieved July 25, 2018, from https://www.zillow.com:443/zestimate/