Predicting Real Estate Prices Using Natural Language Processing and Machine Learning Algorithms

Yipeng Lai [2] , Lauren Low [1] , Dayana Meza [1] , Emma Scott [1] , Elaine Ye [1] , Yanwan Zhu [1]

1 Smith College, 1 Chapin Way, Northampton, MA 01063
2 StreetEasy, 1250 Broadway, New York, NY 10001

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur eget porta erat. Morbi consectetur est vel gravida pretium. Suspendisse ut dui eu ante cursus gravida non sed sem. Nullam sapien tellus, commodo id velit id, eleifend volutpat quam. Phasellus mauris velit, dapibus finibus elementum vel, pulvinar non tellus. Nunc pellentesque pretium diam, quis maximus dolor faucibus id. Nunc convallis sodales ante, ut ullamcorper est egestas vitae. Nam sit amet enim ultrices, ultrices elit pulvinar, volutpat risus.

Author summary

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur eget porta erat. Morbi consectetur est vel gravida pretium. Suspendisse ut dui eu ante cursus gravida non sed sem. Nullam sapien tellus, commodo id velit id, eleifend volutpat quam. Phasellus mauris velit, dapibus finibus elementum vel, pulvinar non tellus. Nunc pellentesque pretium diam, quis maximus dolor faucibus id. Nunc convallis sodales ante, ut ullamcorper est egestas vitae. Nam sit amet enim ultrices, ultrices elit pulvinar, volutpat risus.

Introduction

Real estate is one of, if not the biggest, investments an individual makes over the course of their lifetime. Traditionally, home valuation has been conducted by professional property appraisers – those licensed and hired to give an opinion of a home's value, based on prices of neighboring homes, property analysis and judgment. In recent years though, there has been an uptick in developing models and algorithmic resources to aid housing price estimates. Companies like Zillow, Trulia, and others have developed machine learning models to automate the job of an appraiser. These organizations take into account housing attributes including property size, number of bedrooms, number of bathrooms, geographic location, state of the economy, among other variables to predict home prices.

StreetEasy, launched in 2006 and acquired by Zillow in 2013, is reshaping the ways people buy, sell and rent property in New York City and New Jersey. StreetEasy's existing predictive tool uses metrics on bedrooms, bathrooms, property size, and geographic features to estimate particular listing prices, similar to Zillow's Zestimate. One feature the current machine learning algorithm neglects is the descriptive text of

given listings. A primary objective of this project is to convert text-based listing descriptions into input variables for the purpose of improving predictive model accuracy. In particular, we want to research whether listing descriptions can be utilized to create meaningful features to improve the accuracy of predicting home prices. Current literature supports leveraging machine learning algorithms with natural language processing to improve prediction accuracy [1,2].

Research Questions and Objectives

Some research questions include:

- Can we use listing attributes to predict home prices?
- Can we convert the description text into meaningful features?
- Can we use text-based features in addition to the existing features to predict home prices?

Many questions already have preliminary answers. In StreetEasy's current model, home prices are predicted using variables such as number of bedrooms, number of bathrooms, property size, time to subway, year the property was built and property neighborhood/geographic coordinates. To predict home prices more accurately, we include the listing description text as tokens in the machine learning model. Using bag-of-words method and word embeddings to extract keywords, our analysis has yielded some key phrases such as "stainless steel," "washer dryer," "windows," "storage," "central park" and "closet space."

In asking and answering these questions our goal is to improve upon StreetEasy's existing machine learning algorithm by combining current models with text-based analysis to predict home prices. By creating new variables from the listing descriptions – binary variables for the term "stainless steel," say – we aim to improve prediction accuracy.

Methods

Data and Variables

The data provided by our project partner includes two datasets on sale listings and amenities. StreetEasy collects information from agents by having them directly enter listing information on the website or through a feed from their brokerage. StreetEasy verifies the listing and property information based on records from New York City's Department of Finance and Department of Buildings [3].

Our main data set consists of 59,661 sale listings of 52639 unique properties listed and/or sold on Streeteasy in 2019. It has 31 variables including home price, property attributes (e.g., the number of bedrooms, the number of bathrooms, size in square feet), geospatial information (e.g. state, zipcode, longitude, latitude), and a text description of each listing.

Data Preprocessing

Exploratory data analyses show that the distributions of the variables are skewed and 13 of them contain NA values for up to 11242 observations. We performed data cleansing, reduction, and imputation to pre-process our data before running the analyses. First, we select the 95% quantile of the non-zero values as a reasonable range of the listing prices. We also filter out the outliers with unusually large values for property size and number of bedrooms and bathrooms.

In addition, we use zip codes to fill in an estimate for observations without valid latitude/longitude information and created more geographical variables such as city, county, and state using the `zipcodeR` package that pulls information from U.S. Census data [4]. We perform these two steps because of the varying quality of geospatial information in the original data set. First, some of the listings contain four-digit zip codes and erroneous longitude and latitude values such as $(0, 0)$. Secondly, the original location information about areas and neighborhoods was entered manually by agents, so the location variables do not have uniform criteria and may differ in the level of specificity. Although the new geographic coordinates might not have street-level precision, by adding a leading zero to four-digit zip codes and incorporating associated features from the `zipcodeR` package, we are able to obtain geographic information that is consistent across listings.

In order to prevent data leakage for our machine learning models, we remove the listings that appear more than once in the data set. The duplicate listings are results of agents posting the same listing multiple times with updated information. In order to eliminate duplicates, we first group the listings by `property_id` and keep the ones with the fewest NA values for all variables within each group. Next, we select the rows with the longest listing descriptions and the largest number of bedrooms within each group. Last, we use the `distinct()` function to keep one unique listing corresponding to each `property_id`.

Because nearly 20% of the observations do not have values for the `size_sqft` variable, we plan to use the `mice` package to impute missing property size values by regression (`norm.predict`) [5]. The predictors of the imputation consist of the number of bedrooms, the number of bathrooms, unit type, floor count, city, and state.

After data cleaning, we eventually have 41327 listings in the data set. The number of bedrooms ranges from 0 to 36 with a mean of 2.58 (SD = 2.11). The number of bathrooms ranges from 0 to 20 with a mean of 1.13 (SD = 1.13). On average, the listings are located on the 10th floor, and were built in 1946. The distribution of `size` is skewed, with a mean of 1496 square feet but a large standard deviation (SD = 1120.27). Similarly, `price` also has a skewed distribution: the mean price is 1,125,424 dollars, but the standard deviation is over 870,000. In our analysis, we will log transform `size_sqrt` and `price` when necessary.

Table 1. Descriptive statistics for continuous variables

A table from the psych package in R

Variable	vars	n	mean	sd	medin	min	max	range
bedrooms	1	41327	2.58	2.11	2	0	36	36
bathrooms	2	41327	1.85	1.13	2	0	20	20
size_sqft	3	31955	1496.34	1120.27	1190	100	36000	35900
floor_count	4	41327	9.94	14.30	4	0	96	96
year_built	5	35888	1946.03	161.83	1955	0	2021	2021
price	6	41327	1125434.27	874393.42	829000	40000	4599000	4559000

Figure 1 shows the distributions of the log price of listings by state, and the vertical lines indicate the mean log price for each state. After log-transformation, the price variable is more normally distributed. In general, homes in New York State are more expensive than those in New Jersey.

Data Analysis Plans

In order to determine whether adding text features as predictors will improve the accuracy of predicting housing price, we first create models without text features and then compare their model evaluation metrics with those with extracted text features. We select Root Mean Squared Log Error (RMSLE) as our evaluation metrics for the

Fig 1. Log-transformed Distribution of Home Prices

linear regression model; the smaller the RMSLE, the more accurate the model is in 102
predicting the outcome. To avoid overfitting and ensure the generalizability of our 103
models, we plan to perform k-fold cross validation on each of our models and take the 104
mean RMSLE as the metrics for comparison across models. 105

To establish a baseline, we create a null model that only has mean log price as the 106
predictor. We expect all other models we build to have better prediction performances 107
than the null model. 108

Models without text features. We are working on building a linear regression model 109
using log price as outcome and a decision tree/ random forest model that automatically 110
predicts which categories of price range a listing falls into. For the linear regression 111
model, we plan to use stepwise selection to select the top 5 most significant predictors of 112
housing price, build the model using those variables, conduct 5-fold cross validation, 113
and then compute the mean RMSLE. 114

Models with Text Features 115

To convert the listing descriptions into meaningful text-based features, the words are (1) 116
tokenized: the words inside of the listing descriptions are split into word or words (i.e., 117
multiple words tokenzied together are called 'n-grams') known as tokens to create a 118
one-token-per-row for text analysis; (2) vetted for stop words: removing words that 119
have little to no meaningful information for analysis (e.g., 'the,' 'as,' 'a,' 'of'); (3) 120
stemmed: finding the base or root of words (e.g., we want to categorize 'windows' the 121
same as 'window'), and lastly (4) analyzed for extracting keywords and n-grams. 122
N-grams generate words that commonly occur together and improve accuracy of 123
sentiment analysis models [6,7]. These steps allow us to infer the value from the listing 124
descriptions using regular expressions or models. A second method to featurize the text 125
is the bag-of-words (BoW) approach. This model enables us to measure the presence of 126
each word throughout all listing descriptions inside the database. In more technical 127
terms, we measure the frequency of each token. A supplementary approach to the two 128
methods indicated above is pre-trained word embeddings or train word embeddings. To 129
contextualize the semantic relationship of terms, word embeddings examine the pairwise 130
correlation between words. Assessing co-occurrences is critical in creating a highly 131

predictive model. ¹³²

References

1.

Chen P-H, Zafar H, Galperin-Aizenberg M, Cook T. Integrating natural language processing and machine learning algorithms to categorize oncologic response in radiology reports. Journal of digital imaging. Springer; 2018;31: 178–184.

2.

Ivanov O, Wolf L, Brecher D, Masek K, Lewis E, Liu S, et al. Improving emergency department ESI acuity assignment using machine learning and clinical natural language processing [Internet]. 2020. Available: `http://arxiv.org/abs/2004.05184`

3.

How a listing makes its way to StreetEasy [Internet]. StreetEasy. NYC Agent Resource Center; 2019. Available: `https://streeteasy.com/agent-resources/working-with-streeteasy/how-a-listing-makes-its-way-to-streeteasy/`

4.

Rozzi G. zipcodeR: Data & functions for working with US ZIP codes [Internet]. 2020. Available: `https://CRAN.R-project.org/package=zipcodeR`

5.

van Buuren S, Groothuis-Oudshoorn K. mice: Multivariate imputation by chained equations in r. Journal of Statistical Software. 2011;45: 1–67. Available: `https://www.jstatsoft.org/v45/i03/`

6.

Kruczek J, Kruczek P, Kuta M. Are n-gram categories helpful in text classification? In: Krzhizhanovskaya VV, Závodszky G, Lees MH, Dongarra JJ, Sloot PMA, Brissos S, et al., editors. Computational science – ICCS 2020. Cham: Springer International Publishing; 2020. pp. 524–537.

7.

Pitler E, Bergsma S, Lin D, Church K. Using web-scale n-grams to improve base NP parsing performance. 2010;