Optimal Home Pricing Prediction

Wesley Stevens
Jacob Barazoto

April 16, 2019

Abstract

Real estate is one of the largest markets in the economy. Previously, brokers have identified optimal house prices on a case-by-case basis. Using machine learning techniques, we propose a model that predicts the selling price of a home in the western United States with 92% accuracy.

1 Problem Statement and Motivation

The housing market is a large and profitable industry; Many people want to know if they're getting a good deal for their new home and many more want to know the value of their home before contacting a broker. We want to automate the process of finding a marketable house by building a predictive model to determine its optimal selling price. In doing so, we will also identify what features are most important in determining the selling price. Although a similar service is currently provided by some real estate websites (such as Zillow.com), only an estimate of the selling price is available. Their processes of identifying important features are not publicly known, nor is the process of getting the price estimate. Through building our own model to obtain these important features, we shall expose this process by gathering, engineering, and applying machine learning algorithms to current real estate data.

2 Ethics

There seem to be no inherent ethical problems with providing the optimal price of a house and determining the most important features that influence selling price. Any potentially negative impacts our research could have (i.e. market manipulation) have already been introduced by other pricing services. We merely offer detailed information on how those prices are determined. Whilst gathering data, we discovered that there were website scraping restrictions regarding what information we could use; We strictly followed these rules to stay within bounds of ethical practices.

3 Data

After we researched real estate websites that contained current and reliable data on recently sold houses, we chose to gather data from "estately.com". Their scant scraping restrictions allowed us to legally obtain all the data we needed to build our models. We were confident that the data on this website was credible since they offer real-time updates on houses in the market and rely on licensed real estate agents for that data.

We decided to scrape data from states where sufficient, usable data was provided. California, Arizona, Colorado, and Nevada all provided enough data, so we used these states to build our models. All the other states with sufficient data were not geographically suitable for a model based in the western United States.

Our data initially contained information on ~100,000 listings in these states, with 22 features per listing. After we cleaned and feature engineered what we gathered, 73,000 total listings remained, with 37 features per listing. We determined it was necessary to drop certain features and listings in order to minimize bias and improve accuracy. Many of the added features were simply one-hot encoded.

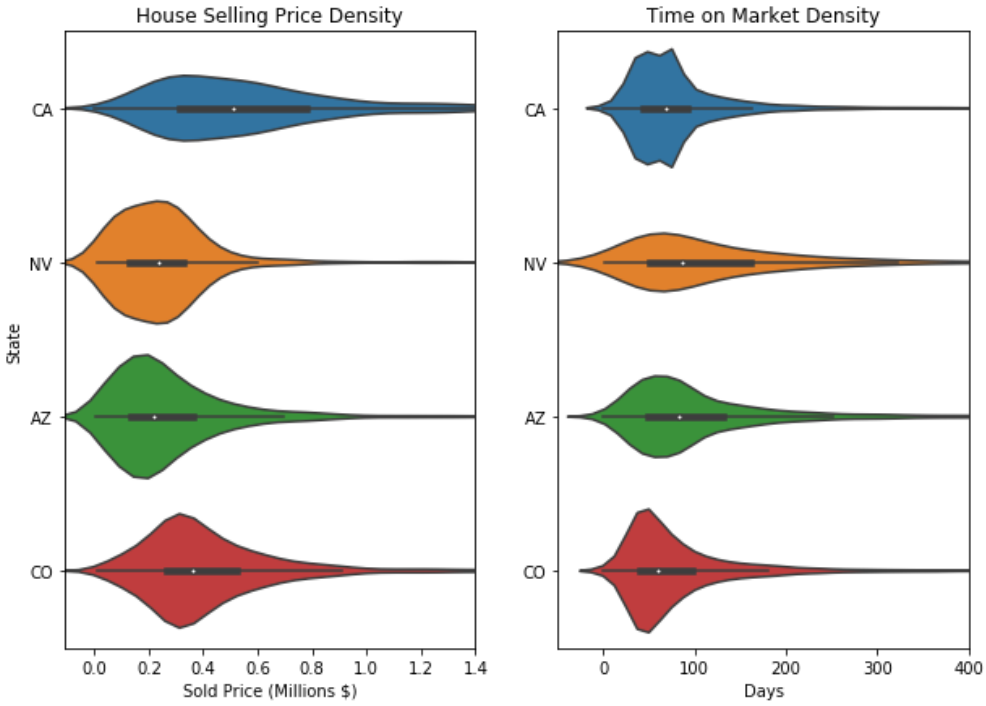
Below, we provide a sample listing from our data. The house size is in square feet, the lot size is in acres, the sold price is in millions, and the elementary school rating is out of ten. We also one-hot encoded by state, the month it was posted, the month it was sold, and if it was a foreclosure sale.

State_CO	0.0
State_NV	0.0
Posted_Feb	0.0
Posted_Mar	1.0
Posted_Apr	0.0
Posted_May	0.0
Posted_June	0.0
Posted_July	0.0
Posted_Aug	0.0
Posted_Sept	0.0
Posted_Oct	0.0
Posted_Nov	0.0

Posted_Dec	0.000000
Sold_Feb	0.000000
Sold_Mar	0.000000
Sold_Apr	0.000000
Sold_May	1.000000
Sold_June	0.000000
Sold_July	0.000000
Sold_Aug	0.000000
Sold_Sept	0.000000
Sold_Oct	0.000000
Sold_Nov	0.000000
Sold_Dec	0.000000
Marketability	0.002325

Bedrooms	4.00
Bathrooms	2.00
HouseSize(sqft)	2246.00
LotSize(acre)	2.00
YearBuilt	1990.00
Stories	1.00
SoldPrice	0.74
UtilityCosts	167.00
ElementarySchoolRating	4.00
DaysOnMarket	59.00
Foreclosed_True	0.00
State_CA	1.00

To familiarize ourselves with our data, we plotted the selling price distribution of each home according to the state the house was in. We can see this in the graph below.



4 Methods

After we talked with a real estate broker, we discovered that once a listing was posted, it ideally would get views after the first week and offers within a month. The average time to close a deal was 2.5 months after the house was originally posted. So we included an indicator to quantify this behavior's effect on the house's selling price. This indicator (named "Marketability" in our data) is a normalized error function with a punishing coefficient. The punishing coefficient is determined by how long the house was on the market. The longer the house is on the market, the more it punishes. It takes as input the time a house is on the market and returns a value between 0 and 1.

We constructed training, testing, and validation sets using an 80/20 random split for the training and validation sets. We then did an 80/20 randomized cross validation split on the training set to get the final training and testing sets. Using cross validation in this manner provides a better estimate of our model's accuracy. Since our data is continuous, it made sense to build our model using regression algorithms–not classifiers. To construct the best model, we compared the accuracy of following algorithms: Support Vector Machine regression, K-Nearest Neighbors, Kernel Ridge regression, an Extreme Gradient Boosting (XGBoost) regressor, and a Gradient Boosting Trees regressor. We discuss these algorithms in more detail in the following section. We also tuned several hyperparameters via a cross validation grid search to boost accuracy. The accuracy's listed in the next section are scores based on the validation set, after we trained and tested the models.

5 Results

5.1 Support Vector Machine Regression

SVM regression is a logical first choice since it is effective in high dimensional spaces while offering efficient computation. However, despite loosely fine-tuning hyperparameters, we only achieved 6% accuracy. This means that our data is not currently linearly separable. The R^2 score was -0.685, which verifies this conclusion.

5.2 Kernel Ridge Regression

We suspected that using the kernel trick to map our non-linear data into a linear space might work, so we next tried Kernel Ridge regression. Kernel Ridge regression projects high-dimensional data to smaller sub-spaces allowing for more efficient computation. Unfortunately, we only achieved a 16% accuracy score. This was probably caused by data loss through projection.

5.3 K-Nearest Neighbors

K-Nearest Neighbors regression offers cluster-based learning on both discrete and continuous labels. Other benefits include low temporal complexity on high-dimensional spaces. We achieved a 50% accuracy rate with this algorithm after several guesses of k. This was worse than we expected, but better than SVM regression.

5.4 XGBoost Regression

XGBoost is an extreme gradient boosting algorithm that uses numerous decision trees to create a strong model by improving weak learners. It uses gradient descent to choose the next predictor that will correct errors in the previous iteration. XGBoost also has a customizable regularization term to severely punish over-fitting. XGBoost also benefits from parallelization which greatly reduces runtime. Fitting an XGBoost model our data with fine-tuned hyper parameters gave us a score of 88.97% accuracy. This is both reliably accurate and well-fit since each decision tree uniquely predicts each feature while not over-fitting or overlapping. Thus we believe that this algorithm offers an accurate predictive model for optimal pricing.

5.5 Gradient Boosting Trees Regression

Gradient Boosting, like XGBoost, is a decision tree ensemble method that sequentially adds predictors to correct previous models. The method updates the model using gradient descent to generate a loss function whose minimum becomes the new prediction. Gradient Boosting also uses a regularization term in its loss function to avoid over-fitting. We obtained an accuracy score of 91.8% in this model. Since this model is robust to over-fitting, we believe that this algorithm offers an even more accurate predictive model for optimal pricing.

6 Analysis

The Gradient Boosting algorithm gave us the highest score on our validation set. So using the Gradient Boosting algorithm and a 95% confidence interval, we discovered that the features used the most in determining optimal pricing were the number of bathrooms, the size of the house, the year it was built, and the house's time on the market. Most of these are features we initially expected to influence selling price, but we were surprised to learn that features such as the number of bedrooms, the location, and the lot size did not. We believe this is because of how gradient boosting works with higher-dimensional data.

In addition to building the independent weak learner models, Gradient Boosting identifies underperforming features within the weak learner models and "boosts" them, or gives more weight to those possible outcomes. These slight changes help the model identify possible dependencies between features, thus improving model accuracy. With our data, it's reasonable to assume that some of our features are dependent on each other. We speculate that house size and lot size, as well as number of bathrooms and number of bedrooms are feature pairs dependent on one another. This offers a probable explanation as to why house size, number of bathrooms are important features in our model but their dependent counterparts are not. Our model appears to identify and account for these dependent features, which we believe is why Gradient Boosting regression is our most accurate model.

As a baseline, we chose to compare our model's results to Zillow's pricing model, the Zestimate®. Nationally, their model predicts (within 20% of actual price) the selling price with 85.1% accuracy. However, within certain states, they are able to improve that accuracy. In California, Arizona, Nevada, and Colorado, they obtained accuracies of 90.9%, 92.9%, 85.1%, and 94.2%, respectively. We were able to achieve an overall accuracy of 91.8% in our model of those states, which is a slight improvement on Zillow's Zestimate®, whose overall average was 90.78% on the states considered.

An even higher accuracy rating might be achieved using more features per listing, or introducing more listings from other parts of the country. The accuracy might also increase if we improve our model for punishing market time. The current model is a basic yet highly functional representation of how long a house should be on the market. Neural networks may also improve accuracy scores, but the marginal benefit might not be worth the runtime.

7 Conclusion

We were successful in creating a model that accurately predicts the optimal selling price of a house in the western United States. In addition, we were able to identify the most important features in that calculation. Compared to the pricing model that Zillow offers, on average we were able to improve accuracy for the states we considered. Furthermore, our prediction service is now available to the general public at "https://www.newhomes.me". Visit this site to see this algorithm in action or to check our prediction for the optimal selling price of your home.

References

[1] "Homes for Sale, MLS-Based Real Estate." Estately, www.estately.com/.

[2] "Homes for Sale, MLS-Based Real Estate." Estately, www.estately.com/robots.txt.

[3] "Machine Learning in Python." Scikit, scikit-learn.org/stable/.

[4] "Zestimate" Zillow, https://www.zillow.com/zestimate/.

[5] "XGBoost, a Top Machine Learning Method Explained" KdNuggets, https://www.kdnuggets.com/2017/10/xgboost-top-machine-learning-method-kaggle-explained.html.