Real Estate Price Prediction Using Machine Learning

Mahavir Parmar   
*College of Business & Economics,  
California State University, East Bay.* Hayward, United States  
mparmar5@horizon.csueastbay.edu

Professor: Chongqi Wu (Project Guide)  
*College of Business & Economics,  
California State University, East Bay.* Hayward, United States  
chongqi.wu@csueastbay.edu

*Abstract*— The real estate industry is the least transparent in our environment. Housing prices fluctuate daily and are sometimes inflated rather than based on valuation. This study project's major focus is on predicting house values using real-world factors. Our goal is to base our evaluations on each basic parameter that is considered when establishing the pricing. We use a variety of regression techniques in this pathway, and our outcomes are not solely determined by one methodology, based on the accuracy of each technique we will choose the best model. The best method produces the least amount of error and the highest level of accuracy when compared to using other algorithms.

Keywords- linear regression, machine learning, prediction, parameters, boosted regression, forest regression, SVR

# Introduction

Data is at the center of technological advancements, and predictive models can now achieve any result. This method makes considerable use of machine learning. Machine Learning is supplying a valid dataset and then making predictions based on it. The machine learns how important a given event is to the overall system based on its pre-loaded data and predicts the outcome appropriately. Predicting stock prices, predicting the possibility of an earthquake, predicting company sales, and so on are only a few of the modern applications of this technique [6].  I chose California as our major research area and are forecasting real-time house values for a variety of neighborhoods in and around the city. I  used factors such as "square feet area," "number of bedrooms," "number of bathrooms," "walkscore," "bike score," "competitive score," and "zipcode." In order to provide reliable findings for all scenarios, I have used a dataset from Redfin real estate website. A total of 1400 observation and 11 features are scraped which can be observed in the figure below. I used the following algorithms in various combinations, and the weight for each method is dependent on the models MAE and R2 We conclude that a succession of algorithms, rather than a single algorithm, produces better outcomes [1] after analyzing for various test runs.

Table

Description automatically generated

Dataset Scraped from Redfin website

# Related Work

## The real estate sector has evolved into a competitive and opaque market. In this market, the data mining method gives developers an advantage by processing data, projecting future trends, and supporting them in making better knowledge-driven decisions. Main goal is to create a model that estimates a customer's property cost based on his or her preferences. Our methodology examines a set of parameters chosen by the customer in order to determine the best pricing for their needs and interests. For prediction, it employs traditional techniques such as linear regression, Ridge regression, Lasso regression, forest regression, SVR and attempts to provide an interpretation of the data produced. Furthermore, high gradient boosting is employed to improve the algorithm's accuracy. It aids in determining the strength of the association between the dependent variable and the other changing independent variable, referred to as the label attribute and regular attribute, respectively.

# Data cleaning

Outliers are data points that are markedly different from the rest of the observations. In general, we eliminate any outliers from a dataset before feeding it to the model, but this isn't always the best way to get a better-fitted model or statistically meaningful results. What if the outlier value is part of the dataset we're looking at? hence, Outlier detection is one of the crucial steps towards model building and should be done with enough consideration as it changes from scope to scope. For this particular project I have restricted the dataset with properties having 5 bedrooms and properties whose price are less than 1.5 million as exotic properties are very rare in special areas and it will bias the model. Removed duplicates as scraping from web might lead to some duplicates and also since all the data is captures as strings we have changed it to numerical since it’s a regression problem of predicting continuous variable based on independent continuous variable. Except for Zip code as it is technically a numeric but logically it is discrete variable. Removing nulls, specials characters, extracting zipcode from address are also parts of data cleaning.

Chart

Description automatically generated

Heatmap after removing Nulls

Scraped the dataset I had around 1400 data point and after data cleaning we have around 842 data points from which I have 75% of the data points for training and the remaining for validation. Before moving onto train/test split, I had to normalize data on the same scale as price bedroom walk score are all on different scale. Normalization was performed on all continuous variables as none of the features were normally distributed.

Chart, scatter chart

Description automatically generated

Price vs Square feet(before removing Outliers)

Chart, histogram

Description automatically generated

Price distribution is not normal

Chart, histogram

Description automatically generated

Since we have shriked our scope of study to 5 bedroom apartments only, we can see the there are more aaprtments listed in this scope. Number of listings decrease as the number of bedrooms are increasing.

Chart, line chart

Description automatically generated

One Hot Encoding is something we have to do since we have a discrete variable zip code which needs attention in regression models. After grouping zipcodes and performing One Hot Encoding on this grouped feature we get the 91 discrete variables, hence adding 91 more features in the dataset.

Table

Description automatically generated

# Train Test Dataset preparation

For further analysis, the data has been separated into two categories: Train and Test. The ideal train-test ratio is 75:25, which means we use 75% of the observations for training and 25% for testing.

# Proposed System

## Our dataset contains a number of critical factors, and data mining is at the heart of our system. First cleaned up the full dataset and trimmed the numbers that were outliers. I chose five distinct machine learning methods and tested our system with various combinations to provide the most reliable results possible [9]. Here I have used a discrete variable zipcode to capture localities with high price per square feet area. I have considered nearby 60 zipcodes and grouped them in one of the zipcodes from those 60 zipcodes. Later I have done one hot encoding on those zipcode to get a better idea of effect of each zipcode on the data point. I used a unique approach to improve accuracy, and from the results its revealed that the actual real estate value is also influenced by adjacent local facilities such as a train station, supermarket, school, hospital, temple, and parks.

* Linear regression

The simplest basic method for prediction is linear regression. It uses two variables: the predictor variable and the variable that is the most important of the two, whether the predictor variable or su. These regression estimations are used to illustrate how one dependent variable interacts with one or more independent variables. The formula is used to describe the regression equation with one dependent and one independent variable.

b = y + x\*a, where b represents the estimated dependent variable score, y represents the constant, x represents the regression coefficient, and a represents the independent variable score.

Graphical user interface, text, application, email

Description automatically generatedFrom the results it is clearly visible that this model is unreliable(very high MAE). Hence we move onto other models to achieve reliable results as compares to this one.

* Forest Regression

Bagging of trees is a technique used in forest regression. The main goal is to arrange the various trees in a pleasing manner. The Variance in the Trees is then reduced by averaging them. A huge number of decision trees are built using this method [3]. The random forest training algorithm uses the bootstrap aggregating, or bagging, strategy to train tree learners[7].

Bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples: Given a training set X = x1,..., xn with responses Y = y1,..., yn, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples:

For b = 1, ..., B:

1. Take n training instances from X, Y and replace them with Xb, Yb.

2. Use Xb, Yb to train a classification or regression tree.

Predictions for unseen samples a' can be created after training by summing the predictions from all of the separate regression trees on a':

A picture containing text, clock

Description automatically generated

Additionally, the standard deviation of the predictions from all of the separate regression trees on a' can be used to evaluate the prediction's uncertainty.

A picture containing text, clock

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Comparatively Forest Regressor has achieved very significant results than Linear Regression, for which this could be considered for modeling prices of house.

* Extreme Gradient Boosted Algorithm

Boosted regression is a form of learning technique that uses decision trees to make predictions by combining a number of weak prediction models [10].

This Boosting algorithm starts with a real-world value y and looks for an approximation F(x) in the form of a weighted sum of hi(x) from weak learners in class H:

A picture containing text, watch

Description automatically generated

Diagram

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

As expected, further achieving higher results as compared to Forest regressor. This model is has higher R2 for which it can

* Support Vector Regression (SVR)

In most linear regression models, the objective is to minimize the sum of squared errors. Take Ordinary Least Squares (OLS) for example. The objective function for OLS with one predictor (feature) is as follows:

Diagram, text, schematic

Description automatically generated with medium confidence

Lasso & Ridge, are extensions of this simple equation, with an additional penalty parameter that aims to minimize complexity and/or reduce the number of features used in the final model. Regardless, the aim as with many models is to reduce the error of the test set. However, what if the only concern about reducing error to a certain degree? Large errors withing an acceptable range is something that SVR gives flexibility about. SVR gives us the flexibility to define how much error is acceptable in our model and will find an appropriate line (or hyperplane in higher dimensions) to fit the data.

Graphical user interface, text, application, email

Description automatically generated

Amongst all models, SVR seems the most promising model. It has the least Mean MAE and maximum R2 which clearly means that it explains 64% of the variability in the dataset.

# Results table

1. Table Type Styles

| Sr.No | Model evaluation | | |
| --- | --- | --- | --- |
| Regression techniques used | Mean MAE | R2 |
| 1 | Lasso | 0.189 | -0.0011 |
| 2 | Ridge | 0.113 | 0.6319 |
| 3 | XGBoost | 0.109 | 0.6321 |
| 4 | SVR | 0.104 | 0.6394 |
| 5 | Forest Regressor | 0.104 | 0.6199 |
|  |  |  |  |

After comparing the above different techniques based on Mean absolute error and also R2, we can say that SVR and XGBoost as well are reliable for this particular case dataset, since they have a least MAE and maximum R2

# Conclusion

A technique has been developed that aims to deliver an accurate prediction of house prices. Linear Regression, Forest Regression, Boosted Regression, SVR are all used to their full potential by the system. The method implemented will provide precise results and eliminate the risk of buying in the wrong residence. Additional customer-beneficial features can be added to the system without interfering with its primary functionality. The addition of larger cities to the database could be a key future update, allowing our users to look at more residences, gain greater accuracy, and so make better decisions.

# Limitations

Couple of possible limitation are as follows:

* Collected 1400 rows of data for this project; however, if we can collect vast amounts of data in the future, we will be able to find many trends in the real estate market. Even though Redfin was the sole source we used to gather data for this project, we could have gotten it from zillow, apartmentlist, redfin, and a variety of other sources.
* Data on various property kinds, middle class family who might think of properties not more than 3 bedrooms. Exotic properties could be a problem and this can lead to the discovery of many additional real estate market inferences. We only collected data from 7 counties in California for this research; however, we may broaden the scope and collect data from all of the California
* **Dataset Size, source, and variety of data:** People keep on changing the price of house and the listing of the house might be updated or deleted which will change the model training and test set base. Every time we scrape we might get different responses because of the above problem. Website hitting rate might also kill connection in between if scraping large amount of data. Hence for a large scale implementation hidden and paid scraping techniques like selenium could be used which actually helps in maintaining the connection and fetches large amounts of data.

# References

[1] <https://www.redfin.com/>

[2] <https://www.geeksforgeeks.org/>

[3] <https://stackoverflow.com/>

[4] <https://towardsdatascience.com/>

[5] <https://machinelearningmastery.com/>

[6] <https://scikit-learn.org/stable/index.html>

[7] <https://seaborn.pydata.org/tutorial/regression.html>

[6] Anaconda Jupyter Notebook