Predicting Home Prices in Bellevue, WA

Christian Avalos

[1] University of Washington Bothell, 18115 Campus Way NE, Bothell, WA 98011, USA

Abstract. The goal of this machine learning project is to predict the price of a home listed in Bellevue, WA. This prediction is made using the king county pricing data set obtained from Kaggle. Different regression models are explored to see which model produces the best prediction percentage.

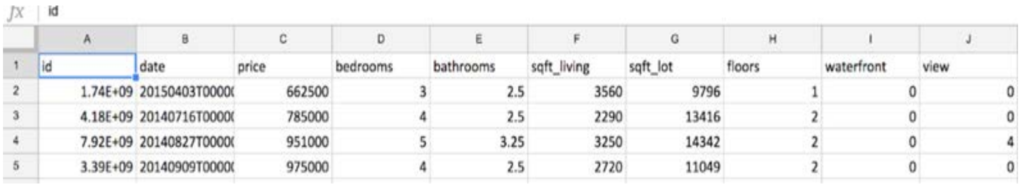
Keywords: Regression, Machine Learning, Housing, Bellevue

1 Introduction

If you are from the Seattle area you have probably noticed or heard about how much home prices have been rising. As people are constantly buying and selling homes, a very common question is asked, "How much will this home sell for" to answer this question with a great prediction would be extremely valuable.

This is exactly the kind of problem I am aim to solve in my machine learning project. I have further narrowed the question down to "Predicting home prices in Bellevue, Washington". Why Bellevue you might ask?

Let's begin with my original dataset label 'House Sales in King County' provided by the excellent data science website Kaggle. It originally contained over 22 thousand data objects (houses) in King County with 21 attributes for each house. To perform a more effective analysis, I filtered this data set to just homes from the city of Bellevue. This was this first step my preprocessing my data (more on that to come).



	A	B	C	D	E	F	G	H	I	J
1	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view
2	1.74E+09	20150403T0000(	662500	3	2.5	3560	9796	1	0	0
3	4.18E+09	20140716T0000(	785000	4	2.5	2290	13416	2	0	0
4	7.92E+09	20140827T0000(	951000	5	3.25	3250	14342	2	0	4
5	3.39E+09	20140909T0000(	975000	4	2.5	2720	11049	2	0	0

Figure 1.1: A snapshot of the original dataset

2 Method

A general outline of the method I used to go about this excrement is shown step-by-step in the diagram labeled Figure 2.1 on the next page.

https://www.datasciencecentral.com/profiles/blogs/deep-learning-pictures

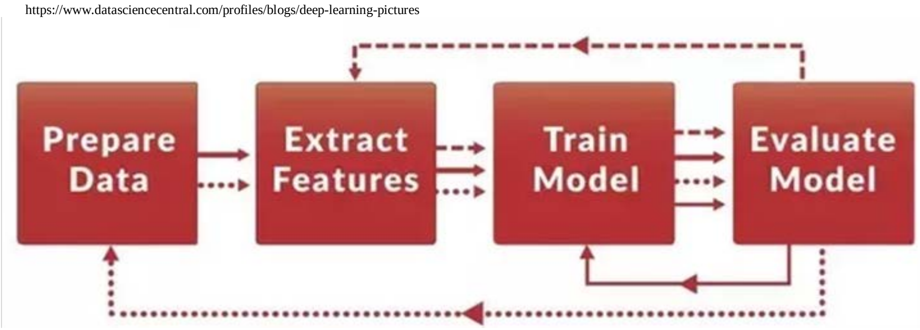


Figure 2.1: A flowchart style example of the research method used

A more detailed summary of the process in words is listed below
-Preprocess
-Play with the data, see how separate features interact with the house prices
-Find a relationship
-With this relationship choose a model
-Implement the model
-Analyze the model (Can it be improved, if so how)
-Implement improvements
-Analyze results (‘Is this method effective?’)

3 Experiment and result

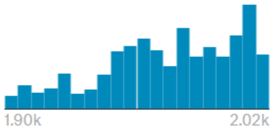
3.1 Pre-processing and cleaning the data

After narrowing down the original dataset, I was left with a smaller sample of around roughly two thousand houses. It was now time to trim some of the features. Manmade attributes of the home are not good predictors in the machine model sense and cannot be generated by machine, so these were removed. Additionally, the non-feature homeID was removed from the set. Lastly, the next step was to split the data into a train and testing set. I chose an 80:20 ratio as the splitting factor.

3.2 Pattern discovery

To begin with my effort to find a pattern I start plotting features. The python library matplotlib, has many great tools to do so. Going through a number of visualizations I was able to notice that there is a positive correlation between the data attributes and the price of the home. Additionally, mostly all attributes follow a normal distribution pattern. A linear correlation shows that this problem could be set up nicely for a regression model. This was exactly what I was hoping to find.

Figure 3.1 An example of correlation between year built and price



3.3 Choosing the Model

Since I noticed that this problem yields itself nicely to a regression model I decided to try a classic linear regression model on the data set. I was comfortable with this model and understood it well since it was introduced to us in quite some depth during the course. This method involves using the least squares method (OLS) to fit the data with the best fitting line.

Using the Linear Regression algorithm from the sklearn package I was able to see how well the model predicted the price of the home compared to test data, as you can see in figure 3, the test data (blue) strays little from the linear function fitting to the data, however the predicted does come a little further off of the line. I calculated this predication at (80.6) percent accuracy using an r^2 score also from the sklearn library. The model has a fairly decent score and looks good at first glance. However, by analyzing this method more closely you start to realize that the method might have a strong bias.

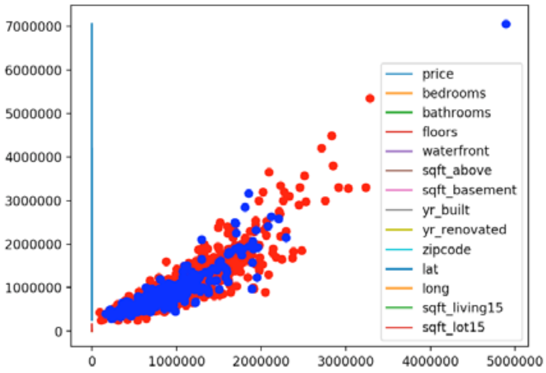
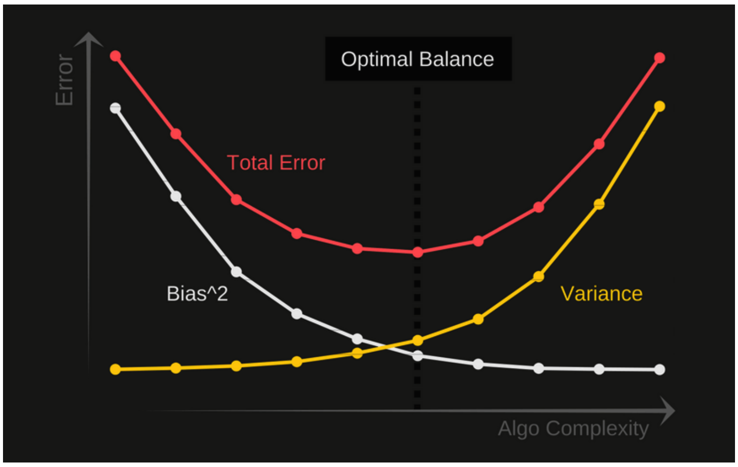


Figure 3.2 Shows the difference between the test and predicted values of the linear regression mode

3.4 Regularization

As mentioned above, the ordinary least squares method of multiple linear regression introduces some tradeoff. There are two things we should really examine, bias and variance. Simply speaking, bias is how biased out data is towards fitting the line and how well the data is predicted based on that line. Variance in this situation is how different data will vary to this line in each trail; or how different will are results be The well fitted straight line used in this method does provide low variance but also high bias. The graph below in figure 3.4 shows the tradeoff we should be striving for.



https://elitedatascience.com/bias-variance-tradeoff

Figure 3.3 Demonstrates the optimal balance between bias and variance that will minimize the total error

3.5 Ridge regression

One method of regression regularization is called ridge regression it works by adding a l^2 penalty to features with high coefficients. Its job is to essentially reduce larger features that might result in an overfitting of the line.

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j}^{m} \beta_j^2$$

Figure 3.4 The formula for ridge regression with the penalty expressed as λ

To implement ridge regression, I imported the Ridge linear model from sklearn and passed in my training data into the set.

3.6 Lasso regression

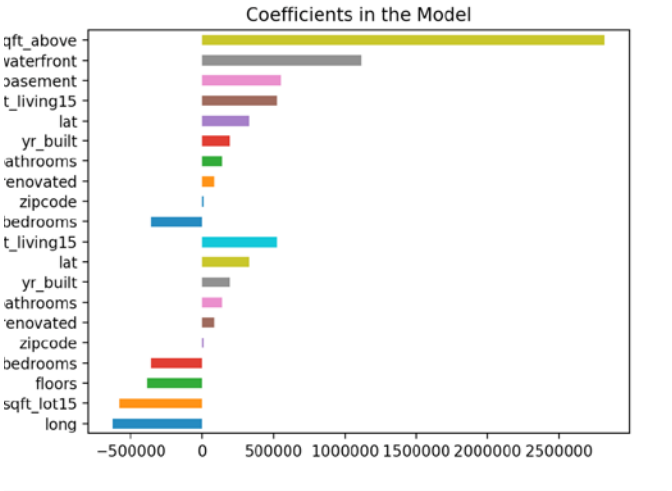
Lasso regression works very similarly to ridge regression in order to perform regularization. The difference in Lasso however is that it acts as a feature reduction tool as well as a regression function. This is due to the fact that it penalties the absolute values of the feature lengths.

Figure 3.5 The formula for Lasso regression with the penalty expressed as λ

$$\hat{\beta}^{\text{lasso}} = \arg\min_{\beta} \sum_{n=1}^{N} \frac{1}{2}(y_n - \beta x_n)^2 + \lambda \sum_{i=1}^{p} |\beta_i|$$

I implemented the Lasso regression technique to my dataset just as I had with ridge regression. However, this time I used a linear regression function named Lasso from sklearn.

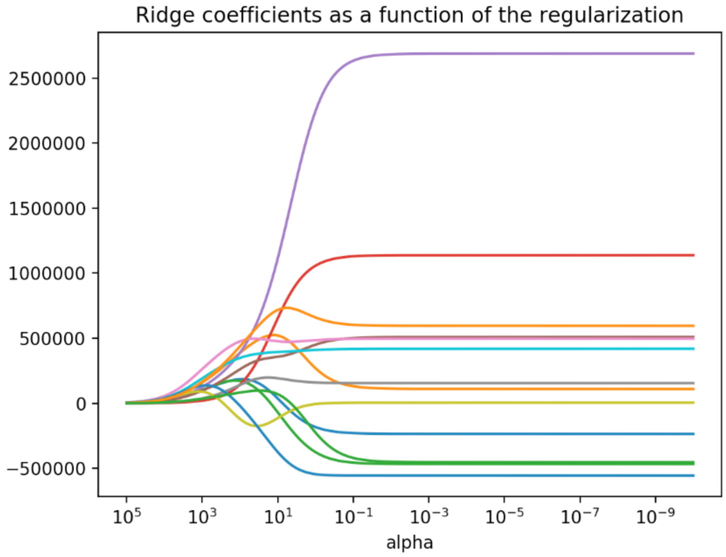
Figure 3.6 Shows the coefficients and their magnitudes (before data was scaled) that Lasso tries to reduce



3.7 Model analysis and results

I have already mentioned my results and how I analyzed and tried to improve my OLS method. Now I will focus on strictly the Lasso and Ridge regression methods. I realized the importance of normalization during the analysis of the models, and changed my dataset so that the features after the train test split were all on the same scaled (between zero and one) Additionally, I learned that Lasso and Ridge models the whole purpose is minimizing bias and variance using by using a penalty against coefficients. Therefore, when evaluating the results and tuning the model it is essential to look at the factor of penalization; or in other words the alpha of the algorithm. I tested some different alphas and came up alpha as 0.01 and 1.0 for ridge and lasso respectively. These ended up with me getting a final score of 78% with ridge and 77.8% with Lasso (using the r^2 method from accuracy function in sklearn)

Figure 3.7 Shows the effect of alpha on the ridge regression model. In my case, the larger the alpha meant the closer the coefficients of the features got to zero.

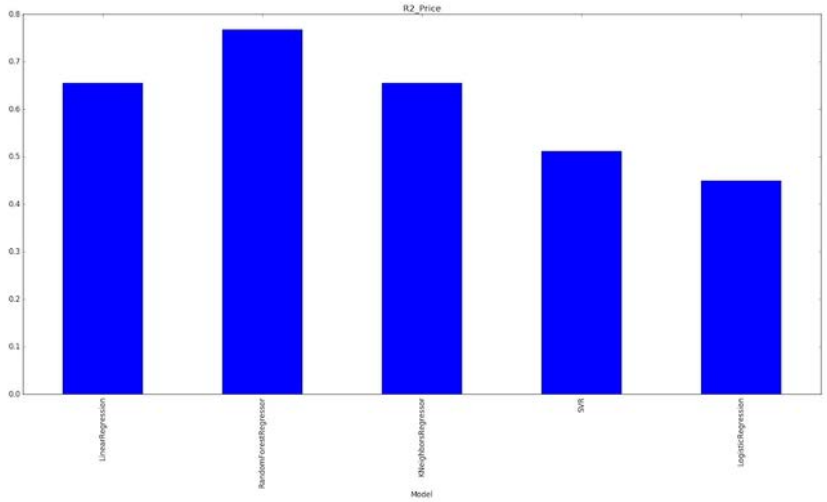


3.8 Previous related work

Luckily for me as a beginner with machine learning concepts, there are quite a few others who have tried predicting home prices with machine learning. Many different methods have been used to try and get the best prediction, however regression is the most common.

I will compare my work with the project posted on Yalantis.com by android developer Ilya Bershadskiy. Bershadskiy begins by quickly explaining the concepts of regression and machine learning in a nutshell. Very similar to the method I used, he gathers the housing data, checks for correlation and chooses a regression algorithm. The main difference in his project however is that he chooses a mix of regression algorithms and other models to implement his final results are indicated in figure 3.8 on the following page.

Figure 3.8 Bar graph displaying the variation in results (R^2 score) Bershadskiy obtained by implementing different models on a housing dataset to predict the price of a home.



4 Conclusion and Discussion

My experiment to predict home prices within the city of Bellevue ended with some good results. By using different regression models I was able to obtain about an 80% successful prediction rate based on the R^2 scoring method from three different algorithms (OLS: 77.7%, Lasso 77.8% and Ridge regression 78.1%) From these results it looks like Ridge regression would be the best model to determine home prices in Bellevue and would give about an 80% chance of predicting the right price.

Although my experiment went well and the prediction numbers were in a good range, I would still like to improve my results. I learned one crucial aspect to machine learning is that the data you feed into your model greatly effects your results and what you get out of it. If I were to do the experiment again I would have spent a lot more time in the preprocessing stage of my project. Making sure there are no outliers, trying a different normalization technique, and perhaps trying a PCA feature reduction on all of the features in the initial dataset. Additionally, if I had more time I would try some more models out and test them on my data.

Overall I am happy with my performance on this project. The large amount of work I put into it was very beneficial to my continued learning of machine learning and the results I obtained. I look forward to many more projects in the future.

5 References

Kaggle https://www.kaggle.com/harlfoxem/housesalespredictionAuthor, F., Author, S.: Ti

http://scikit-learn.org/stable/

https://blog.alexlenail.me/what-is-the-difference-between-ridge-regression-the-lasso-and-elasticnet-ec19c71c9028

https://elitedatascience.com/bias-variance-tradeoff

http://benalexkeen.com/feature-scaling-with-scikit-learn/

http://www.thefactmachine.com/ridge-regression/

https://drsimonj.svbtle.com/ridge-regression-with-glmnet