Price Estimation For Real Estates

Ali Batuhan ÜNDAR [1] **Muhammed Enes KOÇAK** [1] **Muhammed İkbal ARSLAN** [1]

Abstract

In this document, we will show how is our project is progressing. Our project is about predicting house prices from set of continuous data like size and room count alongside with some image data. For this purpose we will test 3 "shallow" learning algorithms, namely; K-Nearest Neighbour(KNN), Linear regression and Support Vector Machine(SVM). We will test our own collected and optimized data.

1. Introduction

House price estimation is one of the problems that defined machine learning. This estimation is very challenging because of the constant change in factors that affects the prices. But there is constant need for such estimation for all kinds of people: sellers, buyers, renters, tenant, agencies and many more. It has uses such as;

- It helps tenants and buyers help when deciding on an estate.
- It helps sellers and renters to decide on best possible prices.
- It, to some degree, prevents scams. An especially sensitive topic modern Turkey.

Even if current technology allows us to tap into nearly endless data online we still require a medium to describe such data. That's where machine learning helps us. It allows us to make accurate predictions/descriptions on data acquired from endless ocean of data. That is what we aim on this project.
Our project is about building such machine learning model that can estimate house prices from given variables extracted from house features. For this purpose we gather our data from internet using real estate agencies. But collecting your own data has it's own share of problems. Such problems multiplied if the said data involves graphics too. That is the problem we are currently facing. The data we collected requires some heavy processing before we can use them. Also we need to check if our data is balanced. We need to test if our data has sufficient length or feature size so we can create a generalized solution.
Our data is 5K houses strong. That includes average of 10 images per house. That is a lot of image data to classify and prepare.

2. Related Work

When we examined the related works, we saw that some research was done according to our project namely image classification and regression-based implementations for continuous data. We mainly narrate the producable and extensible code basely Caffe framework and AlexNet CNN architecture on multi featured images such as (1), (2). In the following lines we will contribute about how it performs with the data which we collect. Additionally, we find linear support vector regression solutions closely to ours such as (3). But when we apply this regression model to our dataset we take 60% test accuracy and which is the lowest value if we compared to other implementations. Also this (4) article talks about how interior and exterior photos of houses impact the price prediction made by computer. While we research the kNN effect, founded (5) is about how the surrounding neighboorhood affects the price of a real estate. Not exactly the same thing but helpful. Parameters is important too. For this reason we read some recently made studies such as (6) that affects the house prices and how much they affect. Very helpful when doing fine tuning. While analyzing the parameters, we realized that some locations is weightly affect the stabil results. In this way, (7) helped us a lot. But it only uses location and image data to make prediction. Meaning it make predictions mostly based on images. And they claim their method is pretty accurate too. Interesting...

3. The Approach

We have tried many algorithms on our collected data. We like to believe our problem is actually can be solved using linear regression algorithms like, well, linear regression and linear SVM. But, since our data also includes images we

[*]Equal contribution [1]Hacettepe University, Department of Computer Engineering. Correspondence to: <>.

BBM406 Fundamentals of Machine Learning, Fall 2018.

have limited option here. We have yet to try CNN algorithm. If we can get successful results using image data CNN will be our preferred algorithm. Otherwise we might ditch image processing part of our project and use continuous data only.
We finally got time to work on our image data set. We found a really good "place" classification algorithm [1]. We are currently classifying our image set into rooms in order to build a proper training set. Algorithms is pretty advanced but perhaps our image data lacks in quality, results weren't perfect. The accuracy for predictions, at best, is 70-80%.

4. Experimental Results

We tried 3 algorithms on our continuous data: KNN, Linear Regression and SVM. We analyzed our data and decided that our data is lacking, be it feature-wise or size-wise. Without any filtering or thresholding all of the three algorithms above gave poor result. That's when we started to do some processing on our data.
First and foremost, location data was a huge issue. Data was, obviously, in string format. We wanted to change this data to numerical value so that we can use it in our models and increase accuracy. For that reason we just enumerated them. We took all the distinct locations and gave them an integer number. But that hardly increased our accuracy. So we think that maybe we can enumerate places in some order. Like prices. So first we ordered places in the order of average prices and then enumerated them. That increased our KNN scores a bit.
Secondly we tried to normalize our data. That reduced KNN score a lot but somewhat improved our linear regression results.
Looking at the regression coefficients we tried to decide what the algorithms believed important and do some feature selection based on that. From that result we deducted that algorithm sometimes weighted some parameters a lot sometimes it did not. For example square meter of a house sometimes took large coefficients, while sometime it took smaller numbers. From that we saw that locations have great impact on how parameters are selected. For example in city "Izmir" we saw that house prices increased in parallel with number of rooms while in "Ankara" it increased more with size of house. That's why we decided to filter for a single city. That raised our lowest results on KNN and linear regression to 40% - 50% .
That was another issue with the data. Even with the filter it was still unbalanced. We shuffled data before each prediction and the accuracy varied greatly. While out lowest score was about 40% our highest score was 93% on KNN and 81% on linear regression. We thought about it a bit and decided that perhaps our location enumeration is flawed.
Issue is that while our enumeration increased linearly the prices vary much more complexly. There is no linear relation between neighbouring(places that have enumeration index close to each other) places. That was perhaps because our data was so little in size, 5.000ish data raw, about 2k after location filtering.
That's currently what we are checking. We wanna get some larger data set with more features. After we feed this data we can compare it with our previous results and correct our wrongs.

5. Conclusions

Conclusion is, even with just using continuous data without image, we can get accurate predictions. But our concern is, now, how to combine image data and continuous data to train a model and make predictions.
After some discussion we came up with a hypothetical model we want to build. We wish to build a network composed of at least one CNN and one Neural network. That one CNN will only take single type of room as input; be it kitchen, bathroom, etc. That way we hope to bring a satisfying solution to our combining different types of data problem.
Another thing we will look into, as suggested, is the "relational reasoning". At our first glance we didn't really understand how this is related to our issue or how it will be any help. But after a little bit of extended research we hope to find a proper solution or at least proper reason to not use it.
Currently another issue of ours is lacking data set. While our data set is "hand picked" it's lacking quantity-wise. Right now we are at a crossroad, either we will continue to use our crafted data set with perhaps a bit more data gathering or we completely ditch our data and search for read-to-use data set. I do not wish latter to happen so we want to try both options, if time allows us, compare the results and determine based on that. If there is not a large difference or results favors our data set we will improve it further and go on with that. Otherwise we really might move on to a better, easy-to-use data set.
Talking about improved data sets there is of course another issue. Image data we have is less than perfect. And since we collect them raw, we had to create our own training set. The set we created is leaves much to be desired, there is a lot of noise. That is another crossroad. We want to use image data too, in our training. But at the same time lack of time and experience kind of worries us. We do not want to enter a road where we will fail in midway. That's why there is a case where we will leave image data completely and train solely on continuous data. But that is a worst case scenario, not something we wish for.
Our current progress, as slow as it might be, is pretty solid in my opinion. Sadly we have yet to find a unique solution for the problem. But there is still time and the perhaps the

[1] http://places2.csail.mit.edu/

"relational reasoning" is the perfect answer we seek. One can dream.

5.1. Figures

Below are some of the test results we got from few "shallow" learning algorithms. The graphs acquired by plotting expected and predicted values in tandem. We are expecting to see unison in change for both of the values on each graph with, perhaps, some margin of error.

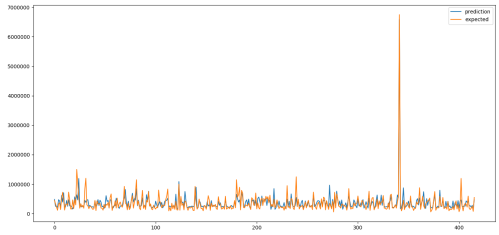

Figure 1. Linear regression results. After data processing, we acquired 80% accuracy on this model

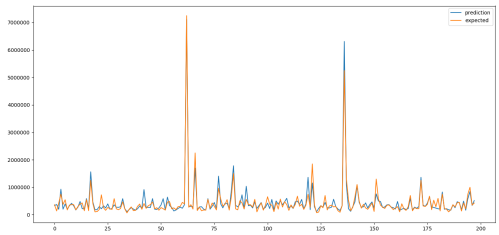

Figure 2. KNN results. After data processing, we acquired 93% accuracy on this model.

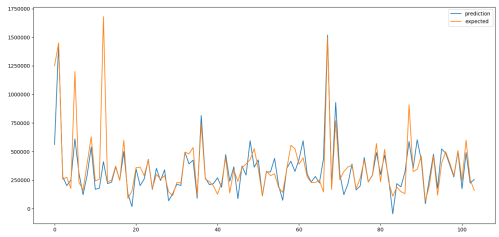

Figure 3. SVM results. After data processing, we acquired 60% accuracy on this model.

5.2. Tables

In this section, with the analysis we made, we wanted to present the results of some cities we have taken from datas with 3 different algorithms. Those result are testing results of the same data with three different algorithms that summarizes some fluctuations actually depending on the data instability.

Table 1. KNN, Linear Regression and Linear SVR results on various data sets.

DATA SET	KNN	LINEAR REGRESSION	LINEAR SVR
ANKARA	0.619244	0.565027	0.486675
İZMIR	0.728051	0.489293	-0.745511
İSTANBUL	0.297611	0.240697	0.122388
AYDIN	0.218803	0.269278	0.029357
BURSA	0.709825	0.758722	0.872432
ANTALYA	0.159593	0.046078	-0.028486
KONYA	-2.011201	-0.296143	-2.218469
ÇORUM	-2.603284	-0.826315	-2.181232
BALIKESIR	0.592623	0.666363	0.518283
MUĞLA	0.618474	0.615173	0.289220

After squared error processed, unusual negative results come up and in our opinion, it is about insufficiency of some specific city data.

References

[1] T.-J. Y. J. E. Vivienne Sze, Yu-Hsin Chen, "Efficient processing of deep neural networks: A tutorial and survey," pp. 1–23, 2017. https://arxiv.org/pdf/1703.09039.pdf.

[2] H. X.-Y. W. K. W. Li Yu, Chenlu Jiao, "Prediction on housing price based on deep learning," vol. 12, no. 2, pp. 90–98, 2018.

[3] A. J. M. R. I. B. Alejandro Baldominos, Iván Blanco and C. Afonso, "Identifying real estate opportunities using machine learning," pp. 11–15, 2018.

[4] S. B. Omid Poursaeed, Tomas Matera, "Vision-based real estate price estimation," 2017.

[5] B. L. Marco De Nadai, "The economic value of neighborhoods: Predicting real estate prices from the urban environment," 2018.

[6] Y. Gu, "What are the most important factors that influence the changes in london real estate prices? how to quantify them?," 2018.

[7] L. C. J. L. Quanzeng You, Ran Pang, "Image based appraisal of real estate properties," 2016.