Machine Learning Engineer Nanodegree

Capstone Project

Niclas Geiger
July 23rd, 2017

I. Definition

Project Overview

Affordable housing becomes an increasing issue. The information of areas with cheaper prices and more expensive areas is often changing and hard to get. Therefore, machine learning can help improve this issue with predicting the prices for certain areas and houses. This will not only be beneficial for willing buyers but also for people looking to sell their house as they can see the best price for this area and commodity in seconds.

Problem Statement

To make a good prediction for housing prizes you need to look at several variables. The housing price depends on the area you live in, the age of the house, the size and much more. All these attributes have to be looked at to make a precise prediction. Predicting the price will most certainly look for similarities in other houses in the area to make an estimation. This is a regression problem using clustering to predict prices for new entries.

Dataset and Inputs

I will use the House Sales in Kings Country Dataset from kaggle. This dataset includes 21613 home sales in between May 2014 and May 2015 in Kings Country which also includes Seattle. Every entry has 19 features which include housing area, latitude & longitude and other important data.
The output data will be the predicted prize for the entered data. Data will be used for cross validating the algorithm. This means the data will be split into different training, validation and testing data for several optimization-runs.
All the data is labelled with the final selling price.

Metrics

For evaluating the result there are several metrics. One is the r2-score, which describes the amount of variance which can be predicted by the model. This usually ranks between 0 and 1 with 0 being the worst.
Another metric for regression is Mean Squared Error. This metric takes the difference in between observed and predicted Values, squares it and then takes the mean of it. The same without squaring is done for the Mean Absolute Error.

Both error Metrics use the difference in between actual data and prediction to score. Regarding house prices we are more interested in the variance in between price predictions thus the chosen metric will be R-squared.

II. Analysis

Data Exploration

	price	Sqft Living
Count	21,613	21,613
Mean	540,088.1	2079.9
Std.	367,127.2	918.44
Min	75000	290
25%	321,950	1427
50%	450,000	1910
75%	645,000	25550
Max	7,700,000	13540

Price

The first values I will have a closer look at will be the price.
We have prices ranging from roughly $75,000 to $7,700,000. 50% of the prices is around $450,000.
The standard deviation is $360,000. The mean of the prices is $540,000.
If we look at the quarters we have:
- first 25% span over $235,000
- second 25% span over $130,000
- third 25% span over $195,000
- top 25% span $7,125,000

This clearly shows that the first 3/4 of all houses lie in a pretty similar range and are at least somewhat evenly distributed in their quarters. The top 25% are really far out regarding their prices which might be caused by a lot of expensive houses mixed with some extremely expensive houses.

Living Sqft.

The second value I will examine is the living space (in square feet). This value spans in between 290 to 13,540 sqft.
The standard deviation is 918 and the mean around 2080 sqft. This points that most of the houses should be in between 1000 to 3000 sqft. When looking at the quarter distribution you get the following picture:
- first 25% is 1137 sqft. in distance
- second 25% spans 483 sqft.
- third 25% is 640 sqft.
- the last 25% is 9990 sqft.

This huge difference proves the previous estimation as at least 75% of the houses are in between 290 to 2550 sqft. in living space. This also kind of resembles the discoveries of looking at the house prices.

Exploratory Visualization

Relationship between price and living space

I will print a plot to visualize the relationship between prices and living space. Usually the available space of a house is one of the biggest factors for the selling price.

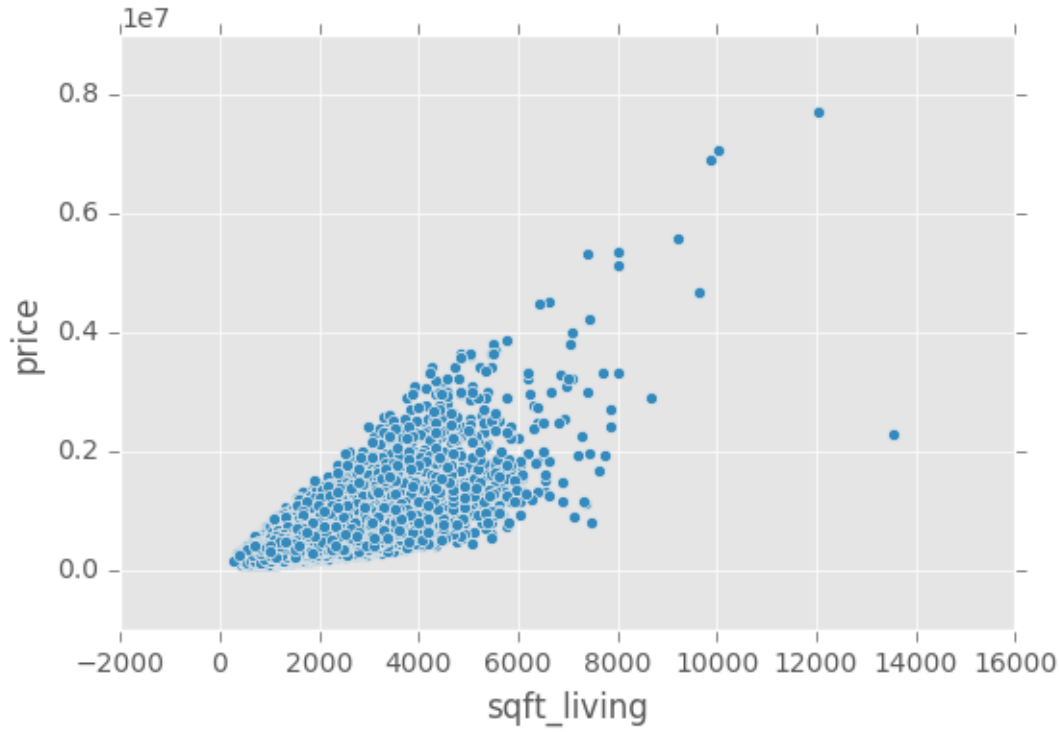


Figure 1 Relationship between living space and price

As one can see here, this kind of hints to a rather linear relationship between both values.

Relationship between prices and location

Another interesting factor for the prices I found was the location of the house. This knowledge is derived from my domain knowledge but can also be visualized by using a heat map over the prices on the geolocation of homes.

Feature Relevance

To try to find out which features might be redundant, we will compare the regression score for each feature. This will give the score we get for predicting a feature just by all the other features. If a feature has a regressor score > 0.95 then this is highly likely to be redundant. If a feature has a regressor score of < 0 then it is not really well described by the rest of the data and might therefore be an important one.

Feature	**Regressor score**
Id	-0.13338979727
Bedrooms	-0.160352462696
Bathrooms	0.50795150982
Sqft living	0.996110554063
Sqft Lot	0.269109294583
Floors	0.592701773203
Waterfront	0.293462831528

View	-0.0166225785212
Condition	-0.333728697699
Grade	0.588520208236
Sqft above	0.98419834569
Sqft basement	0.973475041586
Year built	0.639089125308
Year renovated	-0.417516672101
Latitude	0.990196948347
Longitude	0.95619613868
Sqft Living 15	0.568577429132
Sqft Lot 15	0.182621375198

The result shows that sqft_living, sqft_above, sqft_basement, zipcode, latitude and longitude can be very good described by the rest of the data when you leave them out. I think this might be explained, that all the square foot parameters describe the house and can therefore be explained by the number of rooms and other square foot values. zipcode can explain the latitude and longitude and vice versa. Bedrooms, view, condition and year renovated are all features which are hard to be described but just looking at all other features.

sqft above and basement are both already included in sqft_living which is just the sum of it. To remove the above and basement feature I will check if the ratio of above to basement really has an impact on the price.

Detecting Outliers

The second step to improve our learning behaviour is to find outliers and then remove them from the data set if needed. To detect outliers, I will compare us Tukey's Method to compare the values of each feature for an entry to the difference between the 25th and 75th percentile for that feature. If the difference is more than 3 times higher, we consider an entry to be an extreme outlier.
To investigate if the outliers are errors in selecting data or just expectional data points I made a graph for all features in correlation to the price to see which feature was handled as an outlier, extreme outlier or normal data point.

Looking at the graphs I found out, that the rules for outlier detection are not applicable to some features.

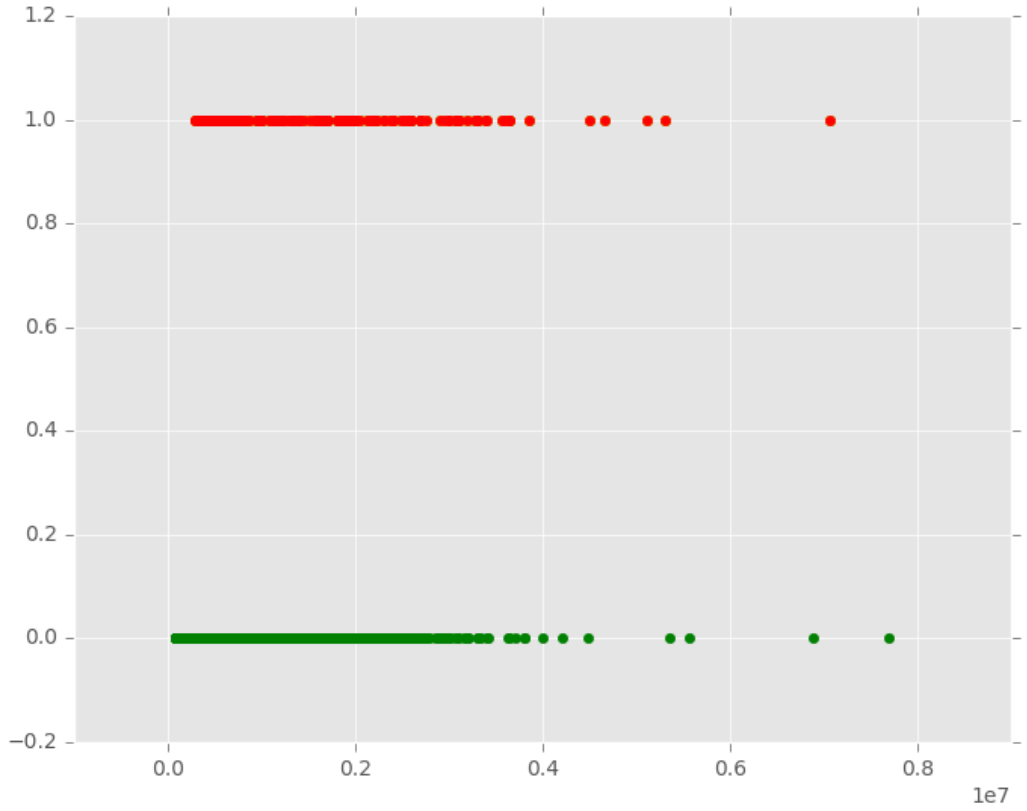


Figure 2 waterfront (Y) to price (X) outlier detection

Waterfront as a binary feature would have meant to drop every data point which has waterfront equals 1 from the dataset as an extreme outlier, which is not logical.
The same goes for year_renovated, view, grade, bathrooms and bedrooms.
One exception I found when looking at the outlier graphs was the one data point for bedrooms.
There is one outlier which has 33 bedrooms for a rather cheap price and just 1620 sqft living space. This definitely seems like an error to me and would most likely disturb the model. Therefore, I dropped this data point.
Another case can be made for exceptional values. For example, looking at the sqft_living graph the outlier detection produced, one can see that everything is split pretty evenly. One data point on the top, leftish corner is kind of standing out. This data point is still viable in my opinion, as this seems to be only due to the variance, but still is inside this cone the data is forming.
Thus, I will only remove this one off data point as everything else seems like just out of the ordinary but still viable data points.

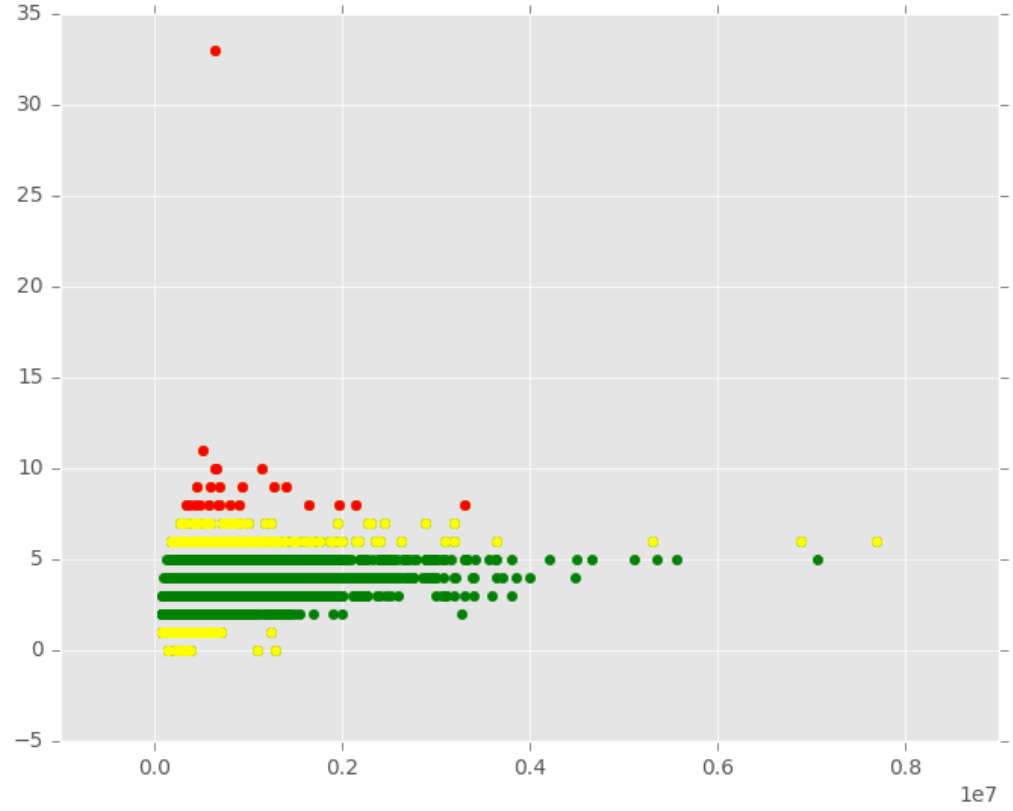


Figure 3 Bedrooms (Y) and price (X) outlier detection

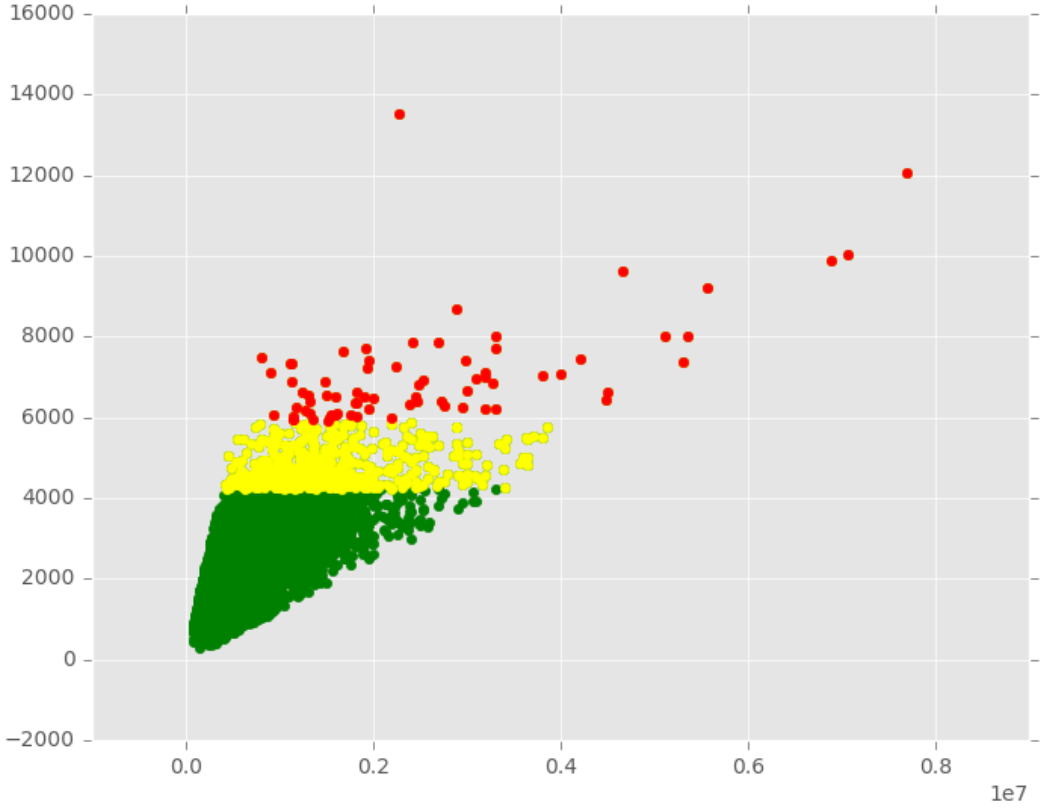


Figure 4 Sqft Living (Y) and price (X) - outlier detection

Calculating Feature Relevance

The next step involved finding feature relevance by calculating how much a feature could be described by using all other features. This was done by using a Decisiontree-Regressor and dropping the feature from the set. Then we split the set into 75% train and 25% test set using train_test_split. The score was the result of predicting the tested feature just using the Regressor. The results were as followed:

Redundant features(>0) : ['sqft_living', 'sqft_above', 'sqft_basement', 'zipcode', 'lat', 'long']

Significant features(<0) : ['bedrooms', 'view', 'condition', 'yr_renovated']

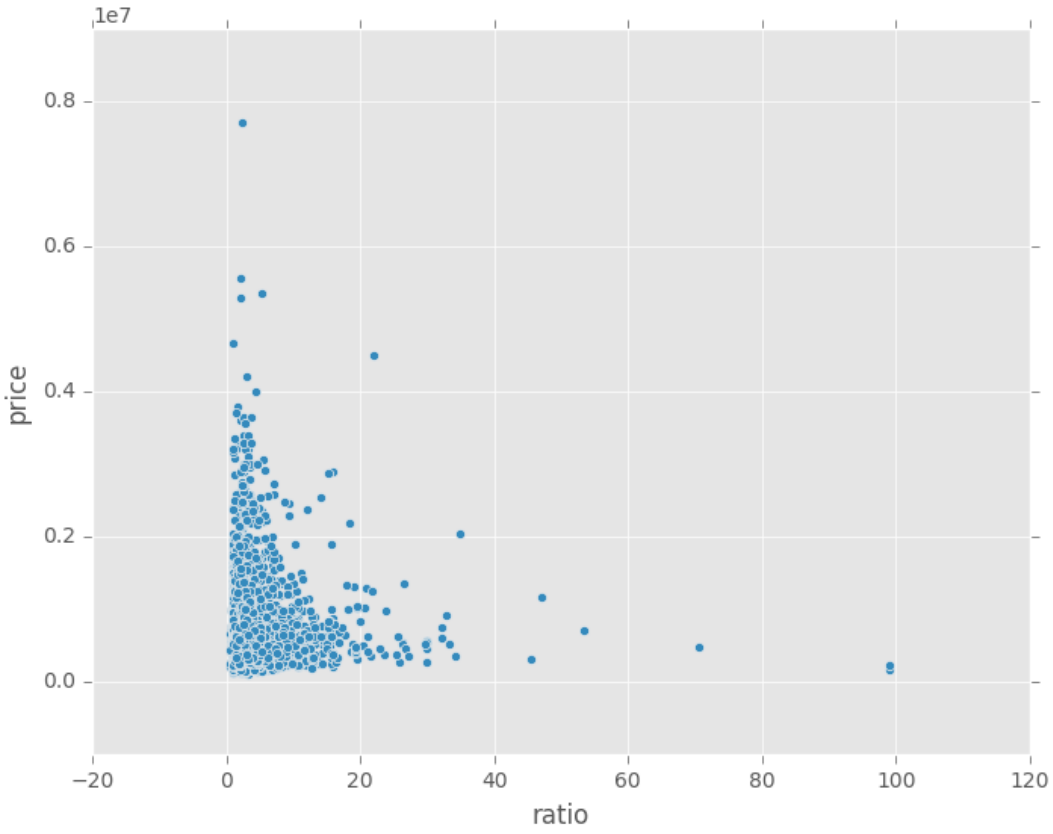


Figure 5 impact of above/basement-ratio to price

The result shows that sqft_living, sqft_above, sqft_basement, zipcode, latitude and longitude can be very good described by the rest of the data when you leave them out. I think this might be explained, that all the square foot params describe the house and can therefore be explained by the number of rooms and other square foot values. Bedrooms, view, condition and year renovated are all features which are hard to be described but just looking at all other features.
This result hinted that “sqft_above” (score:0.98) and sqft_basement” (score:0.99) could be really well described by using other features. Sqft above and basement are both already included in sqft_living which is just the sum of it. To remove the above and basement feature I will check if the ratio of above to basement really has an impact on the price. Therefore, I examined the impact of the ratio between those 2 to the price. No real correlation but some outliers could be found.

Feature Performance

To further investigate on feature performace, I will calculate the entropy (information gain) which each future holds for the data set. This Code was inspired by one of the sklearn examples(http://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html#sphx-glr-auto-examples-ensemble-plot-forest-importances-py) and I use the ExtraTreesClassifier to describe the importance of features.

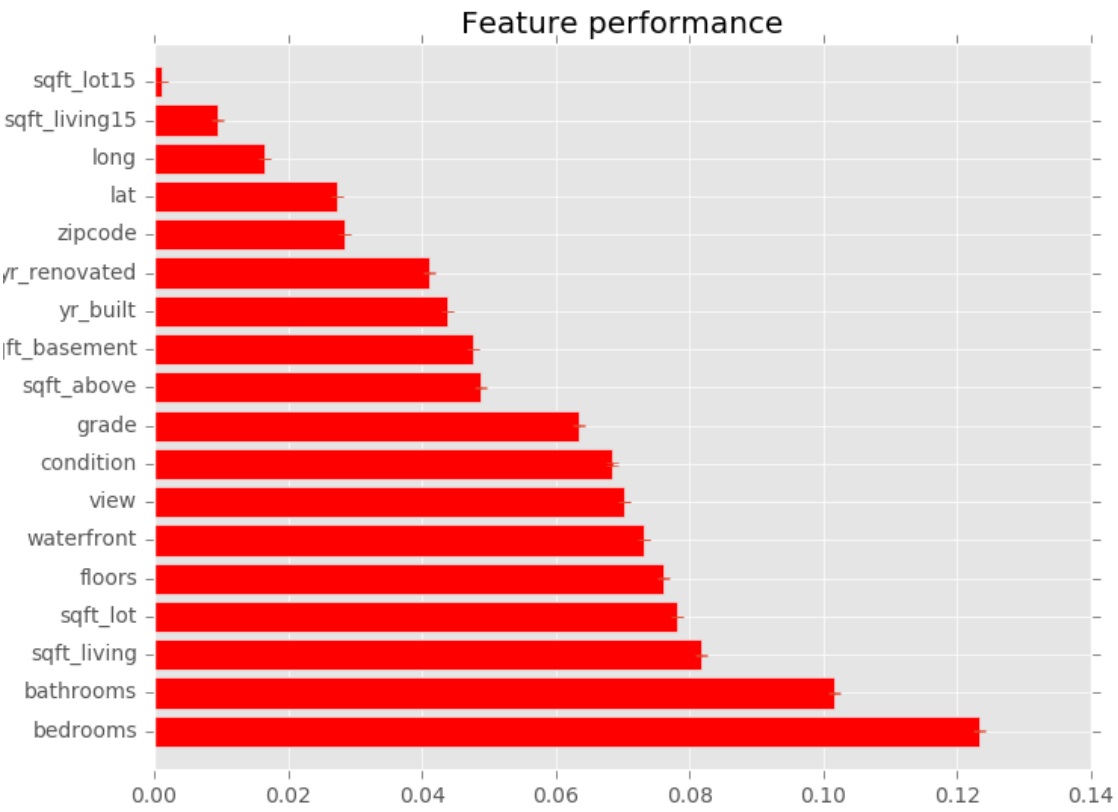


Figure 6 Feature Performance

The Entropy chart is led by bedrooms with a bigger margin. After that bathrooms and square foot living comes in. With a medium value, we have square foot lot, floor, waterfront and view. These are followed by condition, grade, square foot above and square foot basement. Year built, zipcode and latitude have the same low entropy. Longitude and Square Foot Living 15 have even less. The least entropy with nearly zero has square foot lot 15.

what does that mean?

We can definitely see that even though the observation via heatmap of location dependant prices was interesting but longitude, latitude and zipcode all have really bad entropy. The same goes for Square Foot Living 15 and Lot 15.

Algorithms and Techniques

My approach is to try three different algorithms for my regression problem. These algorithms are:

- Linear SVR
- Ridge Regression
- Regression Split with Clustering

They will be discussed in the following.

Linear SVR

Linear SVR is using Support Vector Machines for Regressions with a Linear Kernel. A Support Vector Machine creates a hyperplane which separates two clusters of points. This Hyperplane is computed to provide the biggest distance to every point of each cluster.
I am using the Linear Kernel as the data seems to be really linearly distributed.

Ridge Regression

Ridge Regression aims to find a vector (weight) for multiplication with a matrix which in turn reduces the difference to a target vector to the minimum. It is also known as weight decay.

Regression Split with Clustering

This technique is not standard and I will try it out myself to see, whether it can outperform one of the before presented standard methods or at least the benchmark model.
My Algorithm involved 2 Steps for training and for testing vice versa. The first step involved clustering the data and the second step was training several regressors, one for each cluster combination.

Clustering

The idea was to take correlated features into a cluster thus decreasing the feature space.
For Clustering the K-Means Algorithm will be used.
Two feature sets to cluster were chosen:

- Space Cluster: taking bedrooms, bathrooms and floors into one cluster
- Quality Cluster: taking waterfront, view, condition, grade, yr_built, yr_renovated

Space Cluster should sum up things regarding the space. Quality Cluster is about the quality of the house.
We find 5 different Clusters for the Space Cluster which are:

Cluster \ mean	condition	grade	yr_built	yr_renovated
Quality Cluster 0	3.8	7.0	1918	0
Quality Cluster 1	3.2	7.7	1939	1996
Quality Cluster 2	3.4	7.8	1981	0
Quality Cluster 3	3.0	8.3	2004	0
Quality Cluster 4	3.6	7.0	1955	0

When you look at the data, you see that the different 5 quality clusters mean:

- 0: An old aged house which is still in good condition and has average grades
- 1: An old house, which is in good condition with better grades
- 2: A medium aged house which was renovated recently and is thus in good condition with better grades
- 3: A new house with better grades and rather good condition
- 4: An older house with good grades in a rather good condition

	bedrooms	bathrooms	floors
Space Cluster 0	3.1	1.6	1.3
Space Cluster 1	3.4	2.3	1.5
Space Cluster 2	3.4	2.2	1.4
Space Cluster 3	3.5	2.6	2.0

When you look at the data, you see that the different 4 space clusters mean:

- 0: With an average of about 3 Bedrooms and 1.6 Bathroom spanning 1 floor, this is the typical medium sized house
- 1: Averaging nearly 3 Bedrooms, 2 Bathrooms and spanning 1.5 Floors this is the normal "family house"
- 2: This is pretty similar to the family house above but with a little less space
- 3: With an average of 3.5 Bedrooms and nearly 3 Bathrooms spanning 2 floors on average these are the most spacious houses.

Regression

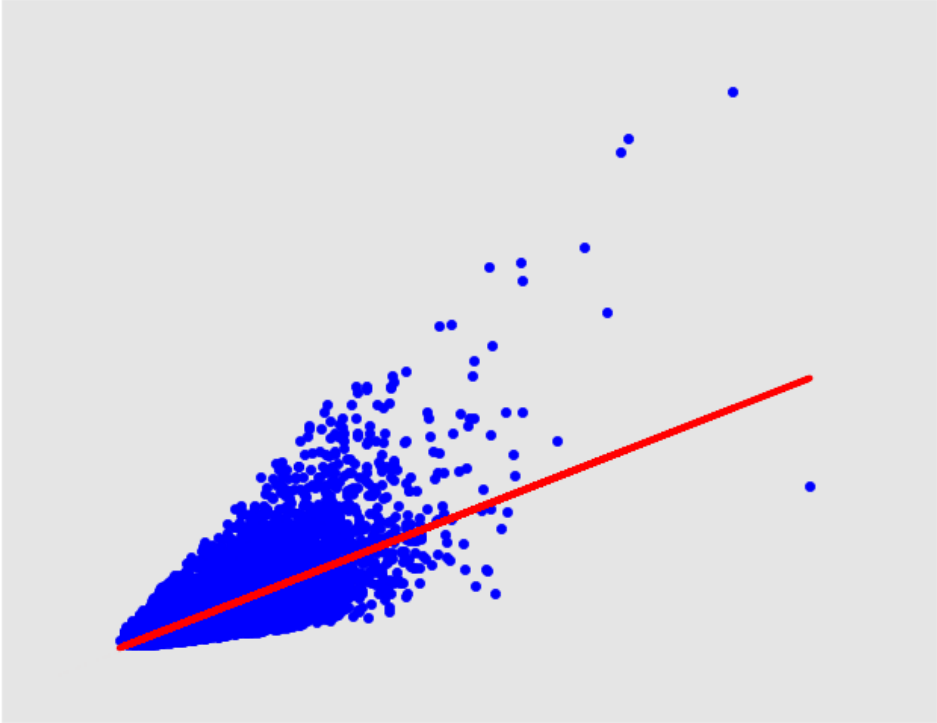
For every pair of quality and space clusters its own regressor was trained with the Linear Regression Algorithm. This was intended to have better linear Regression for better paired house. The Feature used for training was still sqft_living as X and the price as y.

Prediction

When predicting the prices for each entry the same approach was done. First clustering each value in the test set by their respective quality and space cluster. Then using the correspondent linear regressor on that value to predict its price.

Benchmark

My Benchmark model will be a trivial linear regression over the living space of all houses and their corresponding prices. First, I will split the data into training and test sets using cross validation. Then I will train the linear regression on the training set. To get a comparable number I will then take the r2-score of the prediction the linear regression does on the test data compared to the actual data.



For the Simple Regression, the R2 Score is: 0.4929.

III. Methodology

Data Preprocessing

The data is pre-processed detecting the outliers on the data set. All Extreme Outliers (more than 3 times difference between first and third quarter value) were removed from the set. Not all Features were being used in the algorithms:

- Linear SVR dropped: id, date, price, long, lat, zipcode, yr_renovated, sqft_above, sqft_basement
- Ridge Regression dropped: id, date, price, long, lat, zipcode, yr_renovated, sqft_above, sqft_basement
- Clustered Regression dropped: id, date, price, sqft_above, sqft_basement

Sqft_Above and sqft_basement were to be dropped as discussed already.

Id is dropped from all, as this can't really give any insight, because it is only meta information. Date also doesn't really pose any good value as there is no dependency on the price and the date the house was bought.
Price also needs to be dropped, as this is the y-value we use. After comparing the results long, lat, zipcode don't seem to give that much more insight and don't increase the prediction score. Therefore, they were both also dropped.

Implementation

Refinement
I first started out with only the clustered regression approach which ended up performing really weak. Several different Regressors and Clusters were tried, which held even worse results. Then I shifted to using more standard approaches which had way better results.

IV. Results

Model Evaluation and Validation
As mentioned before the benchmark model hit a r2-score of 0.4929.
Scoring for the other models was done the same way split the data into a random 10% to testing and the rest to training. The r2-scores were the following:

- r2-score for Clustered Regressors: -0.84
- r2-score for LinearSVR: 0.374198
- r2-score for Ridge Regression: 0.641903

So only the ridge regression is outperforming the simple linear Regression.

Justification
With a Difference of over 0.16 Ridge Regression is performing way better than the second best (linear regression) and is therefore the best model found.
I think beating the linear regression which is normally used for these kinds of problems by a bigger margin, the ridge regression is really better suited in this case.
The clustered regression was not performing well, but still I learned a lot about using several algorithms in a pipeline in sklearn. In this case it was about first clustering and then applying a regression to this. I tried to use different values for the clusters, adjusting the number of components but still this didn't improve the r2-score really.
I suspect that I didn't find the "right spot" with the clusters. My main idea was that I can find some clusters which decrease the spread of the relation in between sqft_living and price and therefore give a real improvement in the linear regression between those two features.

V. Conclusion

Summarization

First, I described the problem which is the prediction of housing prices. Then I explained, why this problem makes sense to be done using machine learning and why this problem can be improved on by doing so. I described the data set and decided on a metric to be used to evaluate the solution and compare them to a benchmark model. After that I analysed the data feature-wise and discussed some implications. After that I checked the dataset for outliers and discussed the found solutions. This led to one data set being dropped for being suspect to an error in collecting data. Then I checked all features for their relevance and performance.

After that all three of the algorithms and techniques used in this paper were introduced. The Techniques are Ridge Regression, Linear SVR and a clustered regression approach. The benchmark model was introduced which is a simple linear regression. This will be used to compare the results of the aforementioned algorithms.

The Data preprocessing was then explained. After that the results of running all four models were presented and compared. In result the Ridge Regression performed the best out of all models by a bigger margin.

Free-Form Visualization

As already mentioned the visualization presented before of the heat map regarding living cost was quite interesting. Unfortunately, this was not really usable with my approaches but might be worth some further investigation.

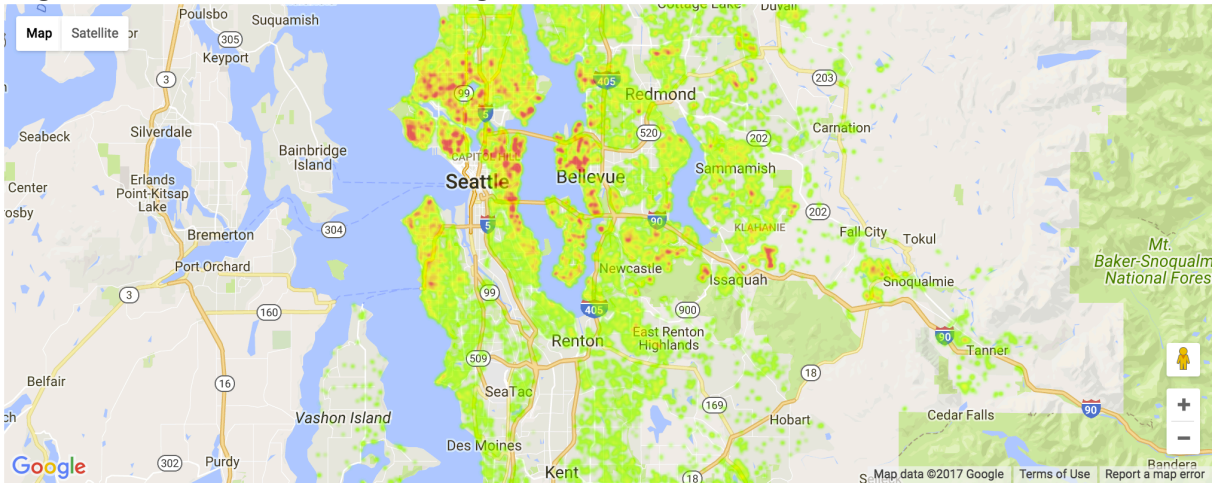


Figure 7 Heat map: Prices based on location

Reflection

I think I am quite happy with the final r2-score of the ridge regression. In hindsight, I feel that I learned mostly the best approach to machine learning is not always the most complicated one. Most of my time was spent on the clustered regression to be honest, which was the worst performing model by a big margin.

While talking to some of my colleague who are already working in this field I heard stories confirming my experience. In the real-life solution being used in industry it often boils down to standard solutions being optimized by their corresponding parameters instead of reinventing the wheel by yourself.

Inventing your own machine learning approaches by just “guessing” what is the right way to do it might oven lead into bad models and/or overfitting. For the next project, I will be doing I really learned that I will try some standard approach at first. Improve the parameters and

make the best out of it before moving onto more creative approaches. Coming from a more creative software background, this was a big learning for me.

Improvement

As mentioned before I believe the geo-data can be used way better. There is a definite correlation between location and prices. Speaking from experience, some areas of a town for example are a lot more expensive than other parts.
Also values like the quality of the house could be fitted way better into a prediction model. I tried to do this with the clustered regression approach but failed really hard on that.
Maybe there is a way to predict the overall quality of the house with a real correlation to the price and also take the location into account to have a really well-done prediction.
If there is a way a real estate agent does this for example out of "gut-feeling", there must be model which can reflect this. Improving these predictions is definitely a really interesting field, which I guess will see a lot of improvement in the upcoming years.

Notes

The whole python code used in the project is in the folder "code".
There is also a jupyter notebook ("housing.ipynb"), but this was only used as a sketch. The only thing which I was unable to redo in the normal python environment was the heatmap, which was not working outside of jupyter.

Graphs generated by the code can be found in "code/img"