Implementation assignment 1:

Behnam Saeedi, Kamilla Aslami, Mostafa Estaji
CS534: Machine Learning
Due Oct 12th 11:59pm
Fall 2018

CONTENTS

1 INTRODUCTION

Purpose of this assignment is to get familiar with machine learning linear regression techniques. This assignment helped us understand critical concepts such as gradient, sums of squares error, learning rate, λ and many other crucial concepts in ML. The team of three worked hard on this project in an equal share in order to achieve the results described in the assignment. Most of the assignment was done by all three members working in a group environment achieving this equal contribution metric. (33% by each member). This document will cover the questions in an orderly fashion followed by a select number of graphs and tables that would help with illustration of our points. Please note that the graphs are all produced by pyplotlib and might not include the full behavior of the lines in order to help us see other lines and their behavior in our graphs. Furthermore, we have scaled some of these lines for the purpose of graphing them in order to make them more human readable and easier to work with. for more detain on these and how to run the code with different option please look into "README.md" file. Finally, the results of our computation for predicting will be stored in "Predicted_y.csv" in the "resources folder". For your convenience we ran the code and stored a backup version of "Predicted_y.csv" in the parent folder of "resources", next to "hw1.py".

2 PART 0

Pre-processing and simple analysis.Perform the following pre-processing of the your data.

2.1 A

Remove the ID feature. Why do you think it is a bad idea to use this feature in learning?

2.1.1 Answer

We remove the ID mainly because the ID is usually assigned on basis that is irrelevant to the nature of the data and considered to be a noise. Removing the ID helps to prevent our regression to take the ID value into account and throw our estimation off.

2.2 B

Split the date feature into three separate numerical features: month, day, and year. Can you think of better ways of using this date feature?

2.2.1 Answer

Perhaps in order to simplify the data we can find the earliest date and set that as 0 and every other day could be the number of days from that day. This way we reduce 3 columns into 1 while not losing the effect of the data. For the purpose of application, data is fine as is.

2.3 C

Build a table that reports the statistics for each feature. For numerical features, please report the mean,the standard deviation, and the range. For categorical features such as waterfront, grade, condition (the later two are ordinal), please report the percentage of examples for each category.

2.3.1 Answer

Table 1, 2 3 and 4 illustrate the requested statistics. The same results are also produced in the beginning of our python script. (due to spacing issues with latex, these tables might be dropped on the next page).

2.4 D

Based on the meaning of the features as well as the statistics, which set of features do you expect to be useful for this task? Why?

2.4.1 Answer

Number of bedrooms, bathroom, square footage, waterfront, condition year built and year renovated perhaps are the most influential aspects of this pricing estimation.

2.5 E

Normalize all features to the range between 0 and 1 using the training data. Note that when you apply the learned model from the normalized data to test data, you should make sure that you are using the same normalizing procedure as used in training.

2.5.1 Answer

All features were normalized as follows: $\frac{X-Mean}{Deviation}$

3 PART 1

Explore different learning rate for batch gradient descent.For this part, you will work with the pre-processed and normalized data and fix λ to 0 and consider at least the following values for the learning rate: $10^0, 10^1, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7$**.**

3.0.1 Answer

The following two figures are the result of plotting those learning rates in 1000 and 10000 iteration models:

Table 1
Total count, mean, Standard Deviation and Range of values

	bedrooms	bathrooms	sqft_living	sqft_lot	floors	sqft_above	sqft_basement	yr_built	sqft_living15	sqft_lot15	price
Total	33752	21188.75	20802232	150892014	15037	17930993	2871239	19711249	19943261	127463234	53852.96817
Mean	3.375	2.119	2080.223	15089.201	1.504	1793.099	287.124	1971.125	1994.326	12746.323	5.386
SD	0.943	0.765	911.334	41203.89	0.543	830.87	435.005	29.481	691.900	28241.243	3.574
Min	1	0.5	370	572	1	370	0	1900	460	660	0.82
Max	33	7.75	9890	1651359	3.5	8860	2720	2015	6110	871200	68.9

	waterfront
No	99.3%
Yes	0.7%

Table 2
Percentage of waterfront houses compared to non waterfront houses

	condition
1	0.13%
2	0.76%
3	65.3%
4	25.69%
5	8.12%

Table 3
Percentage of each condition for the housing

	grade
4	0.11%
5	1.05%
6	9.33%
7	41.3%
8	28.38%
9	11.82%
10	5.47%
11	2.1%
12	0.39%
13	0.05%

Table 4
Percentage of each grade

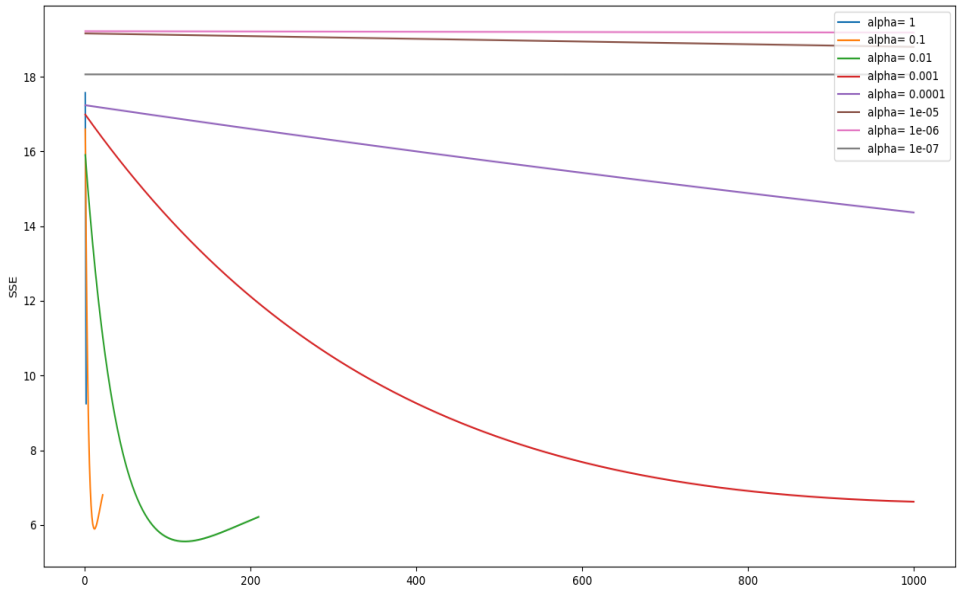

Figure 1. SSE as a function of iterations with different values for α

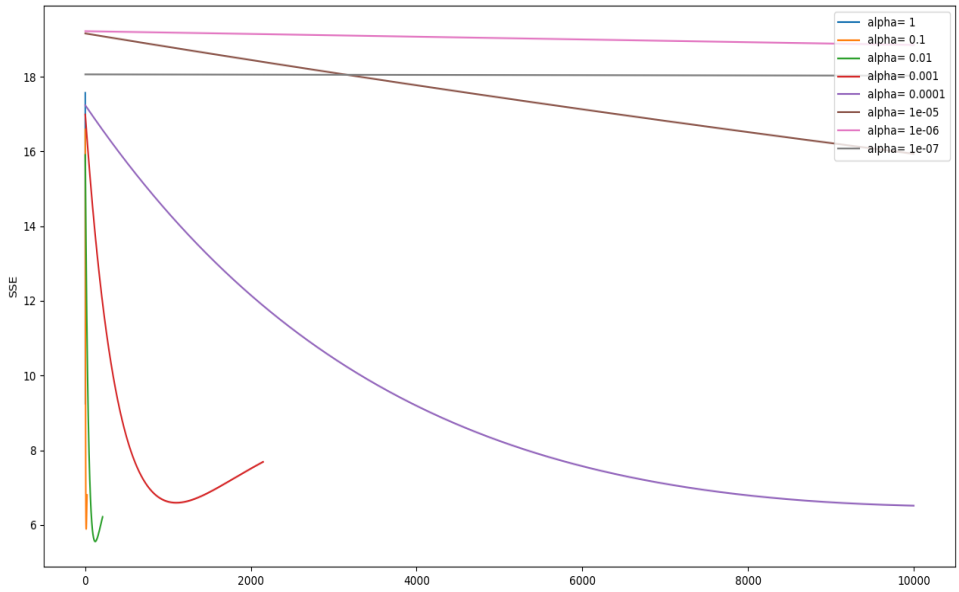

Figure 2. SSE as a function of iterations with different values for α with high performance flag

As you can see The two plots illustrate the behavior of each learning rate as the algorithm iterates through. The first graph illustrates the learning rates on a 1000 threshold and the second set runs the algorithm on a 10000 threshold. Please note, for the plots, learning rates that are large do increase very rapidly which you can see in graph towards the end of the line. we stop plotting them as they start to rise. Some of these results were rather surprising to us since we did not expect certain values to behave in the way they did. for example for learning rates we expected that some learning rates will rise rapidly, though we did not expect this rapid of a change.

3.1 A

Which learning rate or learning rates did you observe to be good for this particular data set? What learning rates make the gradient decent explode? Report your observations together with some example curves showing the training SSE as a function of training iterations and its convergence or non-convergence behaviors.

3.1.1 Answer

Based on our results $10^{-}3$ as learning rate seemed to give us the best results. These results are illustrated in figures 1,2, and 3. As you can see, learning rates equal to 1, 0.1, 0.01, 0.001 start increasing and will make the gradient decent explode. This was interesting to visually see since it behaves rather unexpectedly.

3.2 B

For each learning rate worked for you, Report the SSE on the training data and the validation data respectively and the number of iterations needed to achieve the convergence condition for training.What do you observe?

3.2.1 Answer

Convergence condition for learning rate $\alpha = 10^{-3}$ is achieved after 2127 iterations. Observed SSE value on the training data is 7.223, and SSE on the validation data is 14.583. The error on the validation set is higher than the error on the training set because we train the data on the more significant number of examples. The validation set is twice smaller, and this explains the higher error on validation.

3.3 C

Use the validation data to pick the best converged solution, and report the learned weights for each feature. Which feature are the most important in deciding the house prices according to the learned weights? Compare them to your pre-analysis results (Part 0 (d)).

3.3.1 Answer

Table 5 shows the learned weights for each feature. According to the weights, the most important features are: *bathrooms*, *sqft living*, *floors*, *view*, *grade*, *sqft above*, *sqft basement*, *lat* and *sqft living15*. These results do not precisely correspond to our pre-analysis, but nevertheless, our expectations regarding the number of bathrooms and square footage have been met.

4 PART 2

Experiments with different λ values. For this part, you will test the effect of the regularization parameter on your linear regressor. Please exclude the bias term from regularization. It is often the case that we dont really what the right λ value should be and we will need to consider a range of different λ values. For this project, consider at least the following values for λ: $0, 10^3, 10^2, 10^1, 1, 10, 100$. Feel free to explore other choices of λ using a broader or finer search grid. Report the SSE on the training data and the validation data respectively for each value of λ. Report the weights you learned for different values of λ. What do you observe? Your discussion of the results should clearly answer the following questions:

4.1 A and B

What trend do you observe from the training SSE as we change λ value? What trend do you observe from the validation SSE?

4.1.1 Answer

After experimenting with a range of λ values we generated the following graph in order to demonstrate the results in two following figures:

	weight
intercept	4.82373230589
day	0.171850826813
month	0.299666561604
year	0.0121241192194
bedrooms	0.295571441644
bathrooms	0.55618472642
sqft living	0.642102962067
sqft lot	0.252657183785
floors	0.508001683794
waterfront	0.359891451085
view	0.670561196143
condition	0.295503807259
grade	0.76362181762
sqft above	0.600847198847
sqft basement	0.536477994207
yr built	0.224954921172
yr renovated	0.381322944887
zipcode	0.144001049841
lat	0.713333209614
long	0.231714513465
sqft living15	0.679677413178
sqft lot15	0.254638829381

Table 5
Learned weights for each feature

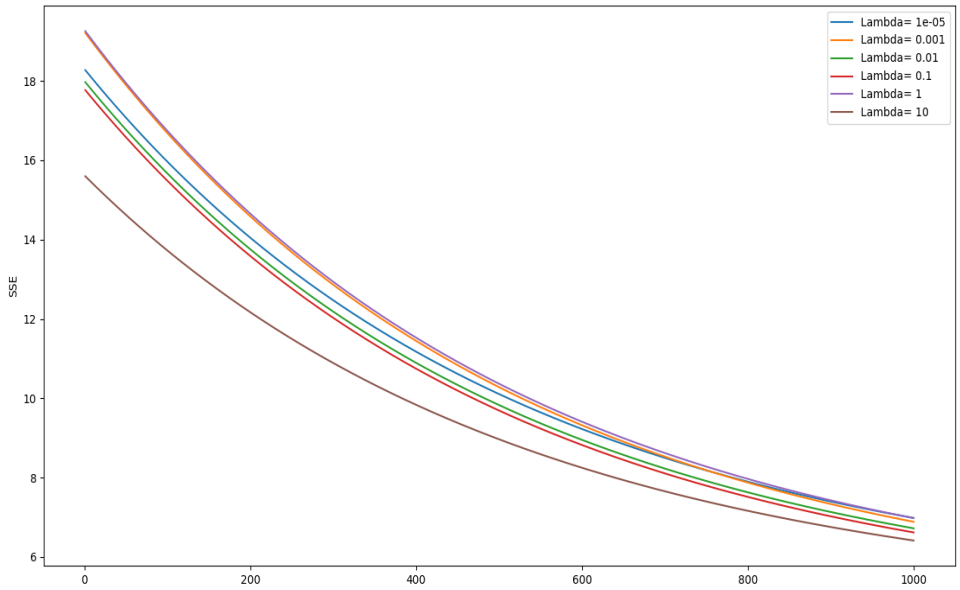

Figure 3. SSE as a function of Iterations with different λ, trade off between variance and bias

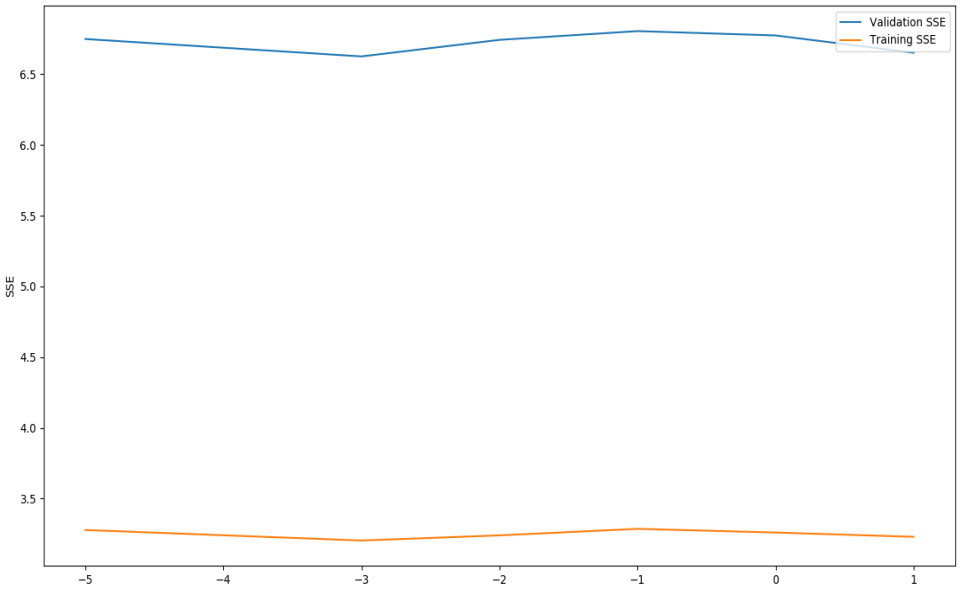

Figure 4. SSE as a function of λ

We can see that indeed higher bias reduces the variance. Based on our results the change in λ value did affect the SSE. This change is consistent in both training (shown in orange on the bottom) and validation sets (shown in blew on the top of training set).

4.2 C

Provide an explanation for the observed behaviors.

4.2.1 Answer

It is important to remember what regularization does. The λ is our bias value which we select based on some prior intuition. We know that the better the λ value, the narrower our distribution around the mean would be. In this case, we can see that some λ values (the larger) yield a smaller SSE in the training set. On the validation set, we can see that SSE in general is much higher (higher variance), However the trend which it is following is more or less similar to the training set's trend, shifted. This decrease in random error however, comes at a cost of large systematic error. In general we are expecting to see that an increase in λ yields a decrease in SSE. (trade off between variance and bias). This effect is seen in the figures above.

4.3 D

What features get turned off for $\lambda = 10, 10^{-2}$ and 0?

4.3.1 Answer

According to Table 6, *day* feature turns off for $\lambda = 10, 10^{-2}$. Also, *sqft_lot15* feature has small weights for $\lambda = 0, 10^{-2}$.

Feature	Lambda=1	Lambda=0.1	Lambda=0.01	Lambda=10	Lambda=0
Intercept	5.38507374	5.38507956	5.38509063	5.38505966	5.38507439
day	0.23890709	0.232891	-0.0874269	-0.00500858	0.1948497
month	0.49843578	0.33814908	0.49608728	0.49112280	0.24021173
year	0.38220061	0.31196398	0.35532883	0.40051902	0.27855505
bedrooms	0.98315649	0.29938558	1.1511544	0.75459655	0.32647954
bathrooms	1.443121	1.70551006	1.60836781	1.72566156	2.06636104
sqft_living	2.29945239	1.7083078	2.49382848	2.13362332	2.03048249
sqft_lot	0.62536965	0.21099809	0.65194555	0.47674363	0.46416995
floors	1.56591837	1.31801055	1.54243174	1.55102899	1.21709207
waterfront	1.13907568	0.49065774	0.55079338	0.84725493	1.22473953
view	2.26100597	2.27705956	2.00551265	2.09051384	2.06844313
condition	0.8882315	0.4847742	0.56168667	1.25795735	0.9267548
grade	2.82464814	2.51524477	2.3235433	2.75139598	2.065943
sqft_above	1.88987469	1.54004243	1.47218628	1.98573825	1.97939303
sqft_basement	1.49926863	1.68338923	1.4303019	1.28447954	1.64182951
yr_built	0.05632265	0.31591756	0.03063231	0.33705693	0.2762873
yr_renovated	0.52092687	0.65917007	0.54379385	1.03647209	1.16739275
zipcode	0.05380372	0.17571772	0.21278629	0.53530588	0.26147735
lat	2.23049395	2.08745909	2.13060618	1.89443114	2.04702414
long	0.16184029	0.40771319	0.20216403	0.73245362	0.35326619
sqft_living15	2.11830308	1.75143423	2.15462798	1.76208887	2.31574608
sqft_lot15	0.47649387	0.9102228	0.15361872	0.66421228	0.11533289

Table 6
Learned weights for each feature with different lambdas

5 PART 3

Training with non-normalized data Use the pre-processed data but skip the normalization. Consider at least the following values for learning rate: $1, 0, 10^3, 10^6, 10^9, 10^{15}$. For each value , train up to 10000 iterations (Fix the number of iterations for this part). If training is clearly diverging, you can terminate early. Plot the training SSE and validation SSE respectively as a function of the number of iterations. What do you observe? Specify the learning rate value (if any) that prevents the gradient descent from exploding? Compare between using the normalized and the non-normalized versions of the data. Which one is easier to train and why?

5.1 Answer

For the purpose of this question a flag option for high performance was introduced. This flag is accessible by adding the letter h as a command line argument to the python code. Passing this flag alongside p, produced the following graphs:

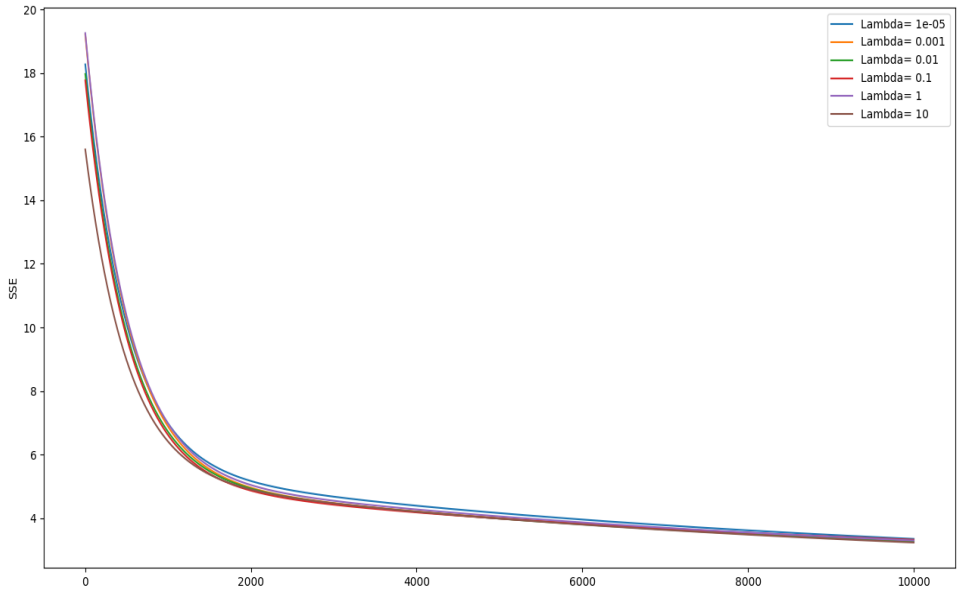

Figure 5. SSE as a function of Iterations with different λ, trade off between variance and bias

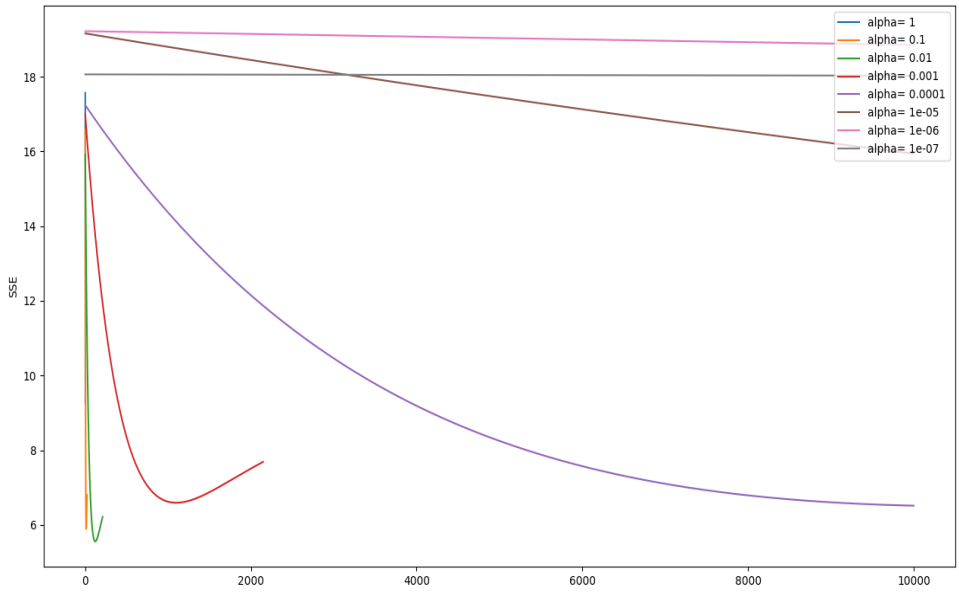

Figure 6. SSE as a function of iterations with different values for α

As you can see there is an improvment over the SSE

6 RESULTS

In this section we included the plot of the results produced by the learning algorithm.

6.1 Low performance

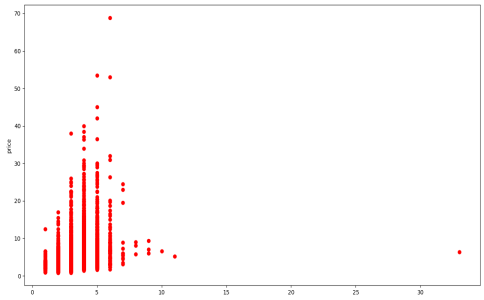

Figure 7. Price as a function of bedrooms for 1000 iteration threshold

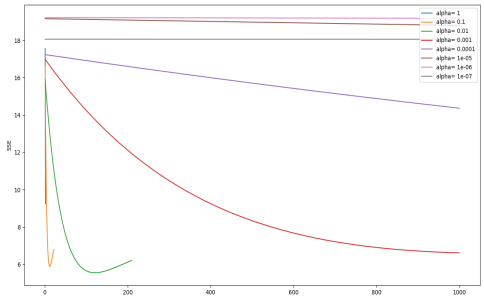

Figure 8. SSE as a function of iterations using a wide range of α for 1000 iteration threshold

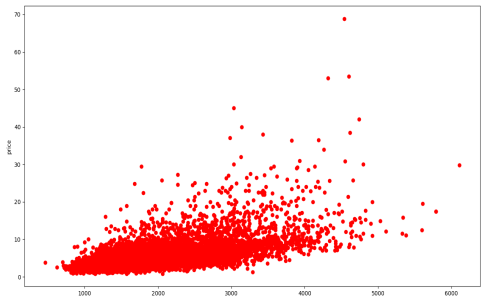

Figure 9. Price as a function of square footage for 1000 iteration threshold

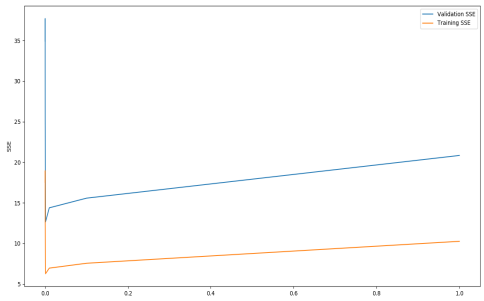

Figure 10. SSE as a function of α for both validation and training sets for 1000 iteration threshold

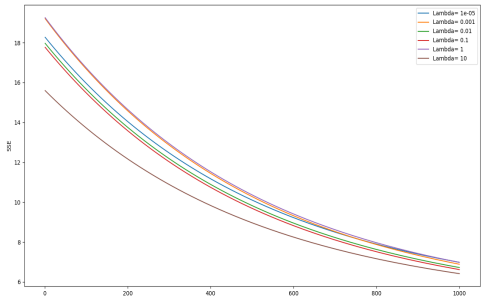

Figure 11. SSE as a function of Iterations with different λ, trade off between variance and bias for 1000 iteration threshold

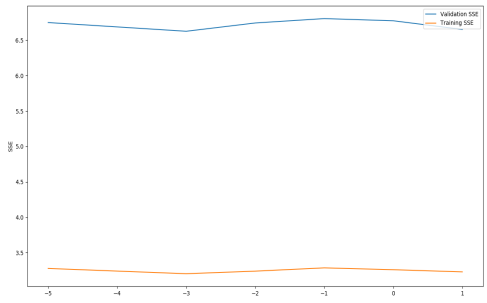

Figure 12. SSE as a function of λ for both training and validation sets for 1000 iteration threshold

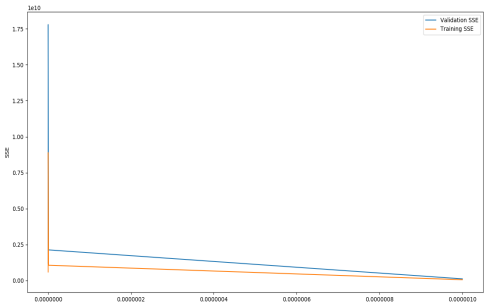

Figure 13. SSE as a function of α for both traininjg and validation sets for 1000 iteration threshold

6.2 High performance

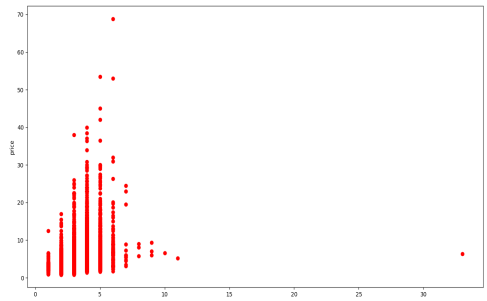

Figure 14. Price as a function of bedrooms for 10,000 iteration threshold

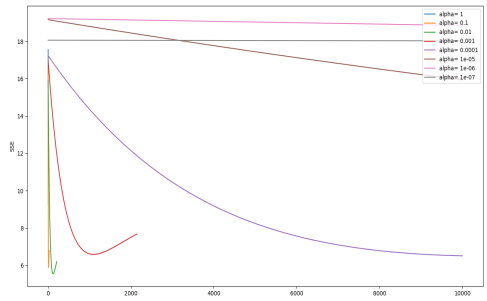

Figure 15. SSE as a function of iterations using a wide range of α for 10,000 iteration threshold

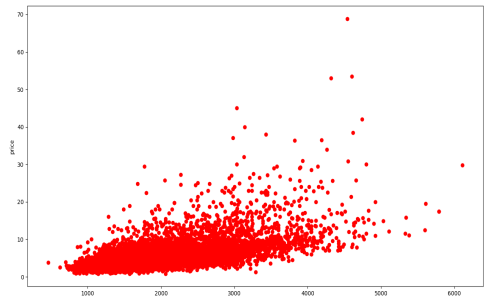

Figure 16. Price as a function of square footage for 10,000 iteration threshold

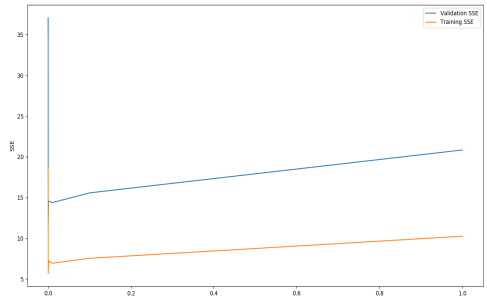

Figure 17. SSE as a function of α for both validation and training sets for 10,000 iteration threshold

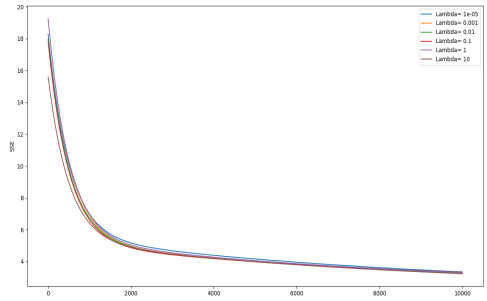

Figure 18. SSE as a function of Iterations with different λ, trade off between variance and bias for 10,000 iteration threshold

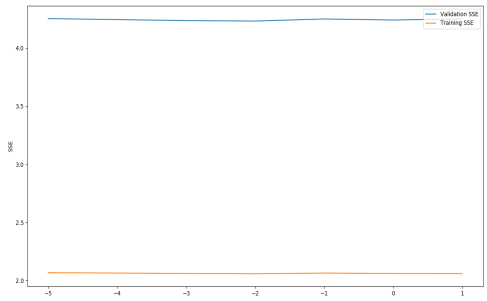

Figure 19. SSE as a function of λ for both training and validation sets for 10,000 iteration threshold

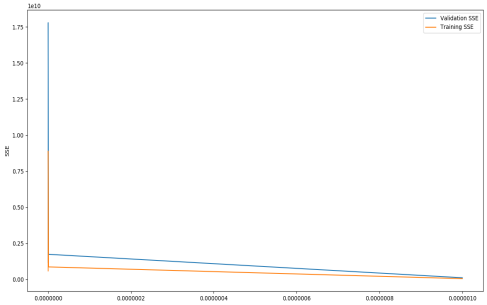

Figure 20. SSE as a function of α for both traininjg and validation sets for 10,000 iteration threshold