Francis L. Bermillo
bermillf@oregonstate.edu

Implementation Assignment 1: Gradient Descent
CS534 Machine Learning

Part 0 (20 pts): Understanding your data + preprocessing

(a) Remove the ID feature. Why do you think it is a bad idea to use this feature in learning?
- **ID does not have any correlation with the pricing so including this feature won't help much in predicting and would only add to the model complexity which may lead to overfitting.**

(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?
- **Computing the age of the house by taking the difference between the year and yr_built can be useful to predicting the price of the house.**

(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 latter two are ordinal), please report the percentage of examples for each category.
- Numerical Table

features	mean	std	min	max
bedrooms	3.4	0.9	1.0	33.0
bathrooms	2.1	0.8	0.5	7.8
sqft_living	2080.2	911.3	370.0	9890.0
sqft_lot	15089.2	41203.9	572.0	1651359.0
floors	1.5	0.5	1.0	3.5
view	0.2	0.8	0.0	4.0
sqft_above	1793.1	830.9	370.0	8860.0
sqft_basement	287.1	435.0	0.0	2720.0
yr_built	1971.1	29.5	1900.0	2015.0
yr_renovated	81.2	394.4	0.0	2015.0
lat	47.6	0.1	47.2	47.8
long	-122.2	0.1	-122.5	-121.3
sqft_living15	1994.3	691.9	460.0	6110.0
sqft_lot15	12746.3	28241.2	660.0	871200.0

Francis L. Bermillo
bermillf@oregonstate.edu
Implementation Assignment 1: Gradient Descent
CS534 Machine Learning

- Categorical Tables

Month	Houses sold (%)
1	4.54
2	5.75
3	8.46
4	10.08
5	11.23
6	10.3
7	10.37
8	8.74
9	8.38
10	8.87
11	6.44
12	6.84

Year	Houses sold (%)
2014	68.15
2015	31.85

Waterfront	Houses w/ waterfront (%)
0	99.3
1	0.7

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

Day	Houses sold (%)
1	2.47
2	3.47
3	3.03
4	3.24
5	3.53
6	3.08
7	3.2
8	3.44
9	3.67
10	3.32
11	2.95
12	3.42
13	3.4
14	3.13
15	2.88
16	3.52
17	3.45
18	3.38
19	3.04
20	3.77
21	3.16
22	3.27
23	4.4
24	3.88
25	3.3
26	3.21
27	3.32
28	2.87
29	2.98
30	3.03
31	1.19

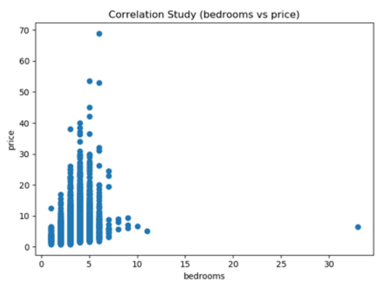
Grade	Houses (%)
1	0
2	0
3	0
4	0.11
5	1.05
6	9.33
7	41.3
8	28.38
9	11.82
10	5.47
11	2.1
12	0.39
13	0.05

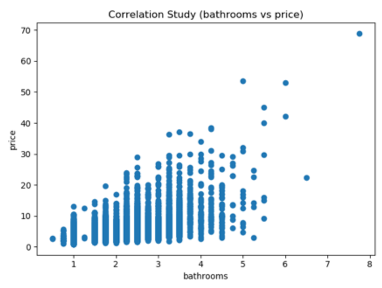
Zipcode	Houses (%)
98001	1.61
98002	0.88
98003	1.27
98004	1.49
98005	0.75
98006	2.35
98007	0.64
98008	1.09
98010	0.42
98011	0.92
98014	0.52
98019	0.83
98022	1.12
98023	2.35
98024	0.31
98027	2.03
98028	1.35
98029	1.6
98030	1.14
98031	1.32
98032	0.48
98033	1.87
98034	2.56
98038	2.67
98039	0.24
98040	1.27
98042	2.6
98045	1.13
98052	2.67
98053	1.86
98055	1.15
98056	1.72
98058	2.1
98059	2.31
98065	1.43
98070	0.53
98072	1.34
98074	2.11
98075	1.77
98077	0.94
98092	1.8
98102	0.52
98103	2.82
98105	1.12
98106	1.5
98107	1.28
98108	0.77
98109	0.47
98112	1.24
98115	2.76
98116	1.52
98117	2.46
98118	2.2
98119	0.79
98122	1.38
98125	1.87
98126	1.72
98133	2.28
98136	1.26
98144	1.55
98146	1.21
98148	0.27
98155	2.07
98166	1.22
98168	1.32
98177	1.09
98178	1.15
98188	0.66
98198	1.38
98199	1.58

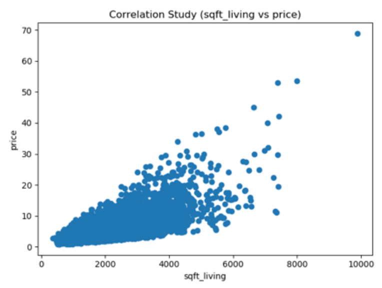
(d) Based on the statistics and your understanding of these features/housing prices, which set of features do you expect to be useful for this task? Why?

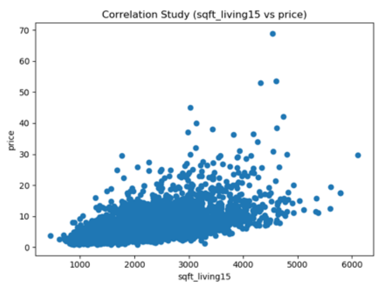
- **After conducting a correlation study between the features and the price, these features (bedrooms, bathrooms, sqft_living, grade, sqft_above, sqft_living15) all have positive correlations compared to the house price as seen below so these will be useful for predicting the house price.**

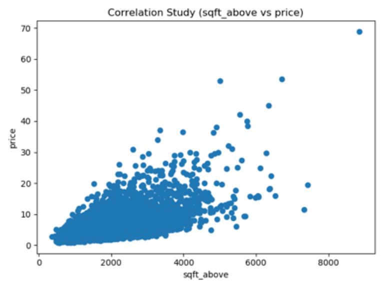
Francis L. Bermillo
bermillf@oregonstate.edu
Implementation Assignment 1: Gradient Descent
CS534 Machine Learning

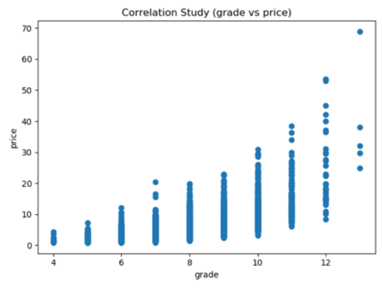












Part 1 (40 pts). Complete the implementation and explore different learning rates for batch gradient descent.

(a) Which learning rate or learning rates did you observe to be good for this particular dataset? What learning rates (if any) make gradient descent diverge? Report your observations together with some example curves showing the training MSE as a function of training iterations and its convergence or non-convergence behaviors.

- **It can be observed from the plots below that learning rates larger than 1E-2 seemed to diverge and caused exploding gradients (Figure 1) while learning rates less than or equal to 1E-2 performed well in this dataset and were able to bring the loss down and converge around ~10 (Figure 2). This makes sense since having large learning rates make large adjustments which will quickly make it diverge from a solution.**

Francis L. Bermillo
bermillf@oregonstate.edu
Implementation Assignment 1: Gradient Descent
CS534 Machine Learning

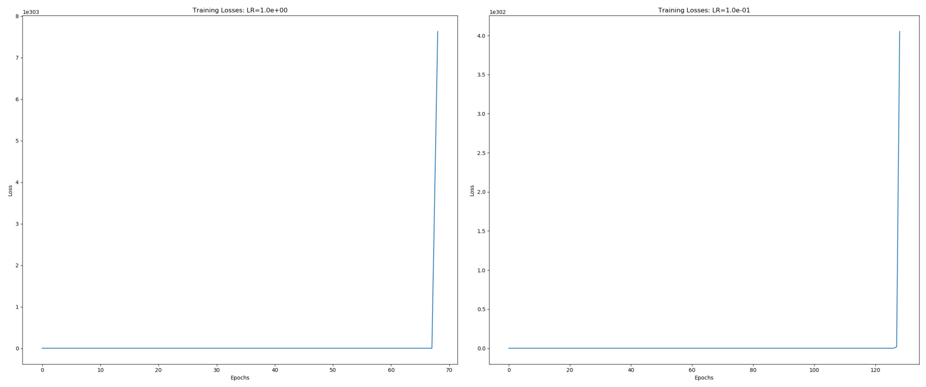


Figure 1: Diverging loss (left plot has a learning rate 1 and right plot has 1E-1)

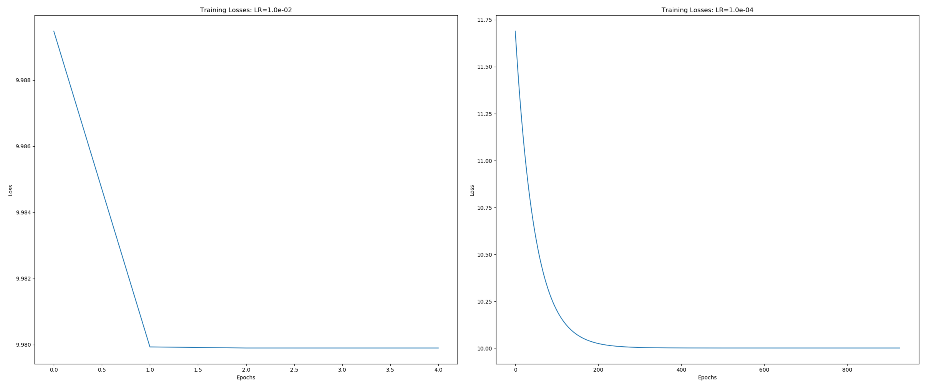


Figure 2: Converging solutions (Left plot has a learning rate of 1E-2 and right plot has learning rate of 1e-4

(b) For each learning rate that worked for you, Report the MSE 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? Between different convergent learning rates, how should we choose one if the MSE is nearly identical?

- ***Please note that the epsilon used for training in this section was set to 1E-3. Learning rates 1E-6 and 1E-7 were skipped because the training took too long.***
- **It can be observed that as the learning rates decrease, the number of iterations and loss increased. This could be due to not fully converging yet and if the training ran for an infinite amount of time, I would suspect that the losses would be much lower for the smaller learning rates.**
- **If one had to choose learning rates that had nearly identical MSE and time and resources is limited, then the larger learning rate should be chosen as it decreases the time the program has to run.**

Francis L. Bermillo
bermillf@oregonstate.edu
Implementation Assignment 1: Gradient Descent
CS534 Machine Learning

Learning Rate	Training Loss	Validation Loss	Iterations
1E-2	9.980	10.039	5
1E-3	9.930	9.937	91
1E-4	10.002	10.040	928
1E-5	10.028	10.021	9765

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

dummy: 1.166
month: 0.436
day: 1.153
year: 1.145
bedrooms: 1.003
bathrooms: 0.835
sqft_living: 1.078
sqft_lot: 0.497
floors: 1.053
waterfront: 0.822
view: 0.217
condition: 0.551
grade: 0.352
sqft_above: 1.186
sqft_basement: 0.682
yr_built: 0.701
yr_renovated: 0.844
lat: 0.573
long: 0.341
sqft_living15: 1.026
sqft_lot15: 0.394
zipcode_98001: 0.715
zipcode_98002: 0.428
zipcode_98003: 0.302
zipcode_98004: 1.066
zipcode_98005: 1.177
zipcode_98006: 1.165
zipcode_98007: 1.111
zipcode_98008: 0.978
zipcode_98010: 0.537
zipcode_98011: 0.285
zipcode_98014: 0.611
zipcode_98019: 0.436
zipcode_98022: 0.337
zipcode_98023: 0.258
zipcode_98024: 0.930
zipcode_98027: 0.216
zipcode_98028: 0.975
zipcode_98029: 0.351
zipcode_98030: 0.284
zipcode_98031: 0.294
zipcode_98032: 0.876
zipcode_98033: 0.449
zipcode_98034: 0.625
zipcode_98038: 0.761
zipcode_98039: 1.065
zipcode_98040: 0.931
zipcode_98042: 0.474
zipcode_98045: 0.336
zipcode_98052: 0.259
zipcode_98053: 0.506
zipcode_98055: 0.466
zipcode_98056: 0.660
zipcode_98058: 0.887
zipcode_98059: 0.900
zipcode_98065: 0.488
zipcode_98070: 0.584
zipcode_98072: 0.385
zipcode_98074: 0.993
zipcode_98075: 0.261
zipcode_98077: 0.901
zipcode_98092: 0.983
zipcode_98102: 0.981
zipcode_98103: 0.464
zipcode_98105: 0.578
zipcode_98106: 0.792
zipcode_98107: 0.477
zipcode_98108: 0.575
zipcode_98109: 0.401
zipcode_98112: 0.664
zipcode_98115: 0.249
zipcode_98116: 1.004
zipcode_98117: 0.281
zipcode_98118: 0.723
zipcode_98119: 0.511
zipcode_98122: 0.782
zipcode_98125: 1.164
zipcode_98126: 0.850
zipcode_98133: 0.239
zipcode_98136: 0.634
zipcode_98144: 0.714
zipcode_98146: 0.740
zipcode_98148: 0.885
zipcode_98155: 0.482
zipcode_98166: 0.333
zipcode_98168: 0.597
zipcode_98177: 1.160
zipcode_98178: 0.391
zipcode_98188: 1.108
zipcode_98198: 0.748
zipcode_98199: 0.661

- **The most important features are highlighted above. Compared to my pre-analysis from Part 0d, the solution shares some similarities to my analysis which were bedrooms, sqft_living, sqft_above, sqft_living15. Some features that the solution deemed to be important are quite surprising, such as day, year, floors, and some zipcodes. These had no positive correlation with the price so there may be some correlation in the combination of these features significant enough to be weighted quite heavily. Others that I would've thought to be have heavier weighting, such as grade, wasn't deemed important which surprising knowing that this feature had the strongest correlation to the price. This could be due to the weight initialization in which I initialized it using a uniform randomization and used a seed of 0 to run the trainings. Maybe running with a different seed would change the outcome of which the best solution would be important**

Francis L. Bermillo
bermillf@oregonstate.edu
Implementation Assignment 1: Gradient Descent
CS534 Machine Learning

Part 2 (20 pts). Training with non-normalized data. Use the preprocessed data but skip the normalization. Consider at least the following values for learning rate: 100, 10, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}. 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 MSE and validation MSE 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?

(a) **None of the proposed learning rates worked in this part, all diverged fairly quickly during training. After sweeping through lower learning rates, I found that learning rates lower than 1E-14 work for non-normalized data but takes a long time to converge.**
(b) **Compared to normalized data, non-normalized data is definitely much harder to train due to exploding gradients. Normalizing data works really well and is easy to train by avoiding the dependence on features which helps prevent exploding gradients. This in turn lets the algorithm train on larger learning rates which converges faster to a tolerable loss.**

Part 3 (10 pts). Feature engineering (exploration). Similar to Part 0(b), list any modifications to the features provided that you think may be useful for making better predictions. Implement 3 (or more) ideas you have and report the results. This is an open question, so you are free to do your own exploration. For example, one simple extension is to try the two ways of representing the zip code feature (numerical or categorical) and use the opposite of what you have been for the assignment thus far. Also, you may try combinations of features (such as ratios, products, polynomial features, etc.). Please report what you have tried and what (if any) effect it has on the predictions.

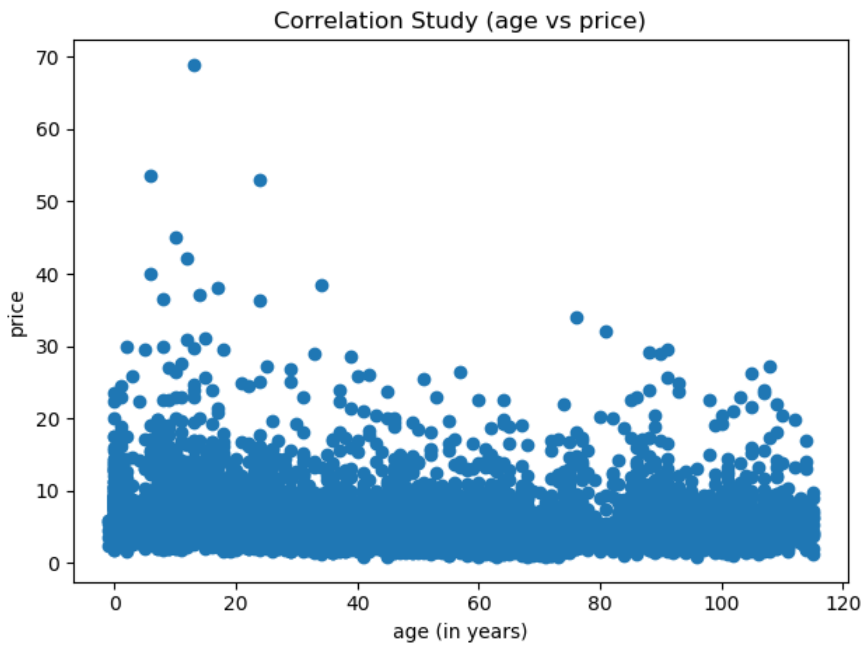
(a) **In this part, all experiments ran with a learning rate of 1E-4 and terminated when delta loss have exceeded an epsilon value of 1E-3. A control was included to compare results with the experiments. Below is a table showing the result of the experiments:**

Feature	Training Loss	Validation Loss	Iterations
control	10.002	10.040	928
normalized zipcode	10.280	10.300	962
age	10.221	10.286	845
score	10.007	10.342	860
removing redundancies	10.111	10.220	1355

Francis L. Bermillo
bermillf@oregonstate.edu

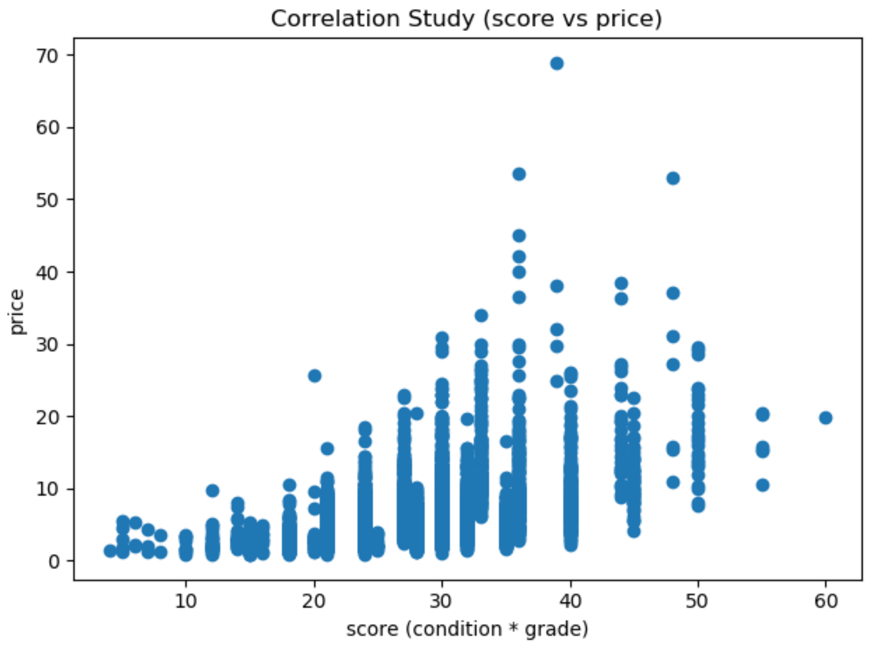
(b) Normalizing the zipcode seemed to have not made any difference and if anything made the solution a bit worse. This is probably due to the needed dependence of the solution on each zipcode to predict the house price in which normalizing it removes some of that dependency away.

(c) The age feature was calculated by taking the difference between the year the house was sold and the year it was built. It's a bit surprising that this didn't do as well since the best solution weighted year heavily, however this could be due to the age not having much of a correlation to the price of the house as seen from the scatter plot below:



(d) The score feature is calculated by taking the product of the condition and the grade. This feature seemed to do the best out of all the other experiments but has the worse validation loss, which is a sign of overfitting the data. Though I would have thought this would have been more useful than age due to the positive correlation with the price as seen from the plot below:

Francis L. Bermillo
bermillf@oregonstate.edu
Implementation Assignment 1: Gradient Descent
CS534 Machine Learning



(e) **Features that was thought to be redundant (sqft_living15, sqft_lot15, lat, long, condition) from the data were removed. The lat and long features were not necessarily redundant but can be represented by the zipcode instead to reduce the complexity of model and the same goes for condition, which could be represented by grade instead. Just like the age feature, this new addition didn't seem to have helped lower the loss in training nor in validation.**