Colton Avila

AI534

15 October 2021

IA1 – Linear Regression

Part 1a)

Learning Rate	MSE	Iterations
10	Nan	150
10**0	Nan	303
10**-1	0.0009	3353
10**-2	0.0255	5000
10**-3	0.0743	5000
10**-4	4.0856	5000
10**-5	12.029	5000
10**-6	14.901	5000

Table 1. Learning rates and the MSE at 5000 iterations convergence criteria.

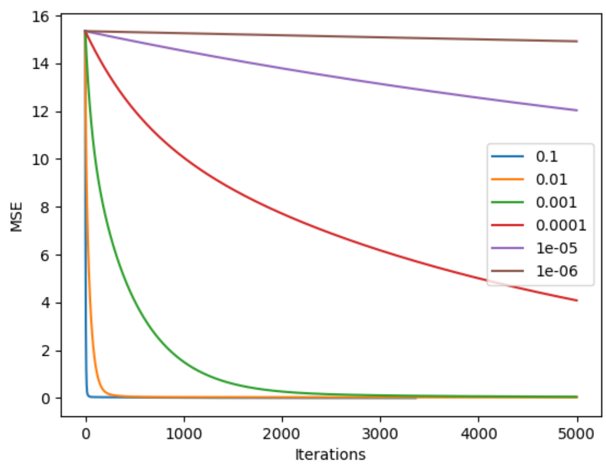

Figure 1. MSE vs the number of batch gradient descent iterations for each learning rate.

The learning rates highlighted in green on Table 1 converged for this training dataset. The learning rates 10 and 10**0 caused the gradient descent to diverge. The high learning rates did not converge in the 5000 allowed iterations.

b)

Learning Rate	MSE of Validation Set
10**-1	4.541
10**-2	4.670
10**-3	4.802

Table 2. Learning Rates and the MSE of the validation set.

The best learning rate for minimizing the MSE is 10**-1. We should choose the learning rate that converged the fastest for learning rates with similar values. This allows for larger datasets to converge faster as calculation speeds decrease with increased training set sizes.

c)

Feature	Weight
bedrooms	-0.281
bathrooms	0.339
sqft_living	0.763
sqft_lot	0.058
floors	0.0181
waterfront	3.919
view	0.449
condition	0.200
grade	1.114
sqft_above	0.757
sqft_basement	0.155
yr_built	-0.884
zipcode	-0.263
lat	0.837
long	-0.304
sqft_living15	0.144
sqft_lot15	-0.099
month	0.055
day	-0.050
year	0.173
bias	5.335
age_since_renovated	-0.103

*Table 3. Learned Weights for each feature for learning rate 10**-1.*

We see “sqft_living”,” waterfront”,”grade”,”sqft_above”, and ”lat” having high weight values.

Part 2a)

Learning Rate	MSE	Iterations
10	Nan	28
10**0	Nan	30
10**-1	Nan	33
10**-2	Nan	37
10**-3	Nan	42
10**-4	Nan	49
10**-5	Nan	58
10**-6	Nan	71

10**-10	1.125e+68	5000
10**-11	5296.03	5000
10**-12	6160.18	5000
10**-13	7148.40	5000
10**-14	392489.0	5000

Table 4. Learning rates vs MSE for non-normalized data.

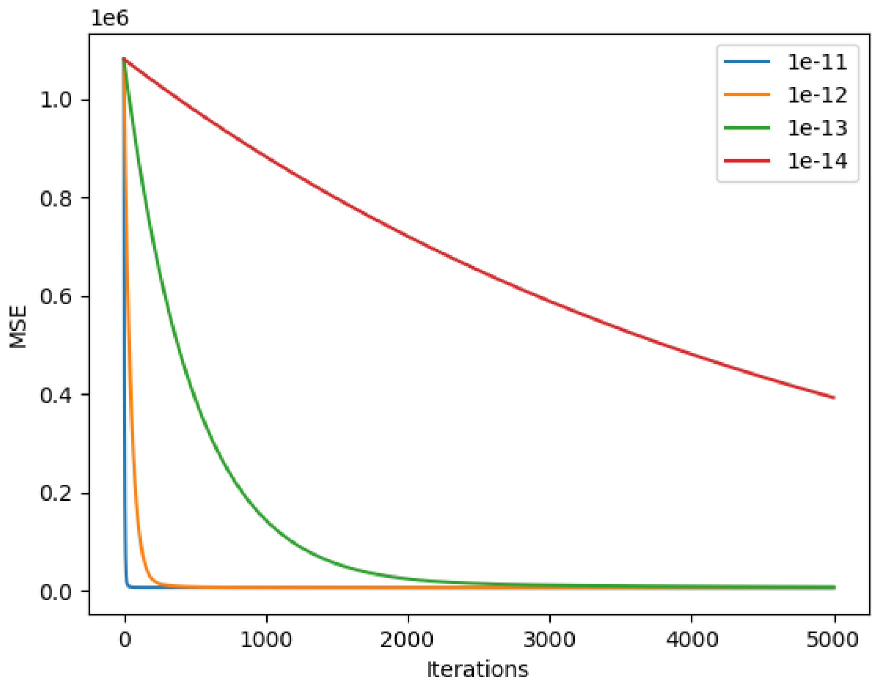

Figure 2. MSE values vs. the number of batch gradient descent iterations for non-normalized data

We don't see the model converge for any of the learning rate previously tried. We also see that the model diverges at 10**-10. The optimal learning rates for the non-normalized data are 10**-11,10**-12, and 10**-13. The learning rate 10**-14 does not converge in the 5000 iterations allowed. The normalized data is easier to train as the learning rates can be larger. The values reach convergence quickly.

b)

Learning Rate	MSE of Validation Set
10**-11	12.555
10**-12	14.13
10**-13	14.31

Table 5. MSE values of validation set using the weights from corresponding learning rates.

Feature	Weights (Non-Normalized)	Weight (Normalized)
bedrooms	9.23E-08	-0.281
bathrooms	1.28E-07	0.339

sqft_living	0.000201	0.763
sqft_lot	4.01E-06	0.058
floors	4.74E-08	0.0181
waterfront	5.49E-09	3.919
view	9.71E-08	0.449
condition	1.46E-08	0.200
grade	2.56E-07	1.114
sqft_above	0.00016	0.757
sqft_basement	4.1E-05	0.155
yr_built	1.11E-06	-0.884
zipcode	4.38E-05	-0.263
lat	3.68E-08	0.837
long	-5.6E-08	-0.304
sqft_living15	0.000127	0.144
sqft_lot15	2.72E-06	-0.099
month	3.46E-09	0.055
day	-9.5E-08	-0.050
year	9.06E-07	0.173
bias_t	4.5E-10	5.335
age_since_renovated	-6.3E-07	-0.103

Table 6. Non-normalized features and their respective weights. Rows highlighted in blue were identified as important features in the normalized data.

Comparing the weights between the non-normalized data with a learning rate of 10**-11 and the normalized data with a learning rate of 10**-1, we see a few differences in which variables are the most important. We see that while grade and waterfront were important in the normalized run, they are quite insignificant in the non-normalized run. The explanation for this may be due to large differences in the scale of input values. Since the values in normalized run are between 0 and 1, we see the differences between 1 and 2 and 100 and 200 as the same. In the non-normalized runs, the smaller values would see much smaller amounts of change. We still see some values are higher consistantly across both runs. This implies that weights can be a measure of feature importance as long as the data is normalized but may not be the best measure.

3)

Feature	Weight (Removed Feature)	Weight
bedrooms	-0.28292	-0.28126
bathrooms	0.333516	0.339201
sqft_living	0.802654	0.762534
sqft_lot	0.051935	0.05817
floors	0.005973	0.019769
waterfront	3.89008	3.906922
view	0.463148	0.448712
condition	0.195792	0.199815
grade	1.159082	1.11661

sqft_above	0.796473	0.755205
sqft_basement	0.160831	0.155568
yr_built	-0.75494	-0.76025
yr_renovated	0.053907	0.055693
zipcode	-0.27291	-0.26349
lat	0.841084	0.836288
long	-0.28661	-0.30364
Sqft_living15	--	0.143705
sqft_lot15	-0.09563	-0.09943
month	0.055762	0.055657
day	-0.05021	-0.05042
year	0.171499	0.172449
bias_t	5.335418	5.335305
age_since_renovated	0.027981	0.019829

Table 7. Weights of features when "sqft_living15" was dropped.

We see that the "sqft_living" weight increased when we dropped "sqft_living15". If two features **x1** and **x2** are redundant, we would expect the weight **w1** when both **x1** and **x2** are used to be less than that in a model where only **x1** is used to learn. This happens because **x2** takes some of the weight from **x1** because they represent a similar thing. We see something similar in our model with a few features.

After dropping the feature "sqft_living15", we found the training set MSE to equal 0.0009 after 3357 iterations. Using the weights above we find that the validation set has an MSE of 4.55. This is ~0.02 larger than the MSE of the validation set using the weights that included "sqft_living15". Our features are not completely redundant, so when we remove "sqft_living15" we end up increasing the MSE. This makes sense since "sqft_living15" is not perfectly correlated / redundant with "sqft_living".

4)

The following section explains some feature engineering methods. We originally tried them on the test and training data not from Kaggle. We will try three different versions of normalization: min-max normalization, z-score normalization, and mean normalization.

$$MinMax(x) = \frac{x - \min(x)}{\max(x) - \min(x)}$$

$$MeanNormalization(x) = \frac{x - mean(x)}{\max(x) - \min(x)}$$

$$ZScore(x) = \frac{x - population\ mean}{population\ standard\ deviation}$$

Learning Rate	Min-Max MSE	Z-Score MSE	Mean MSE
10**-1	4.51	4.50	4.512
10**-2	4.76	4.63	4.75
10**-3	8.078	4.76	8.20

Table 8. Table of MSE for validation set using models trained on test data that was normalized differently

We found that for the training data set Z-score was still the best method of normalizing for reducing the MSE of the validation set.

To lower the impact that outliers had on the model, we tried removing values based on z-score as well as clipping outlier values to the boundary values.

Learning Rate	No Change to Outliers	Z-Score >3 Removed	5-95 Quantile Clipping
10**-1	4.541	5.11	4.44
10**-2	4.670	5.11	4.55
10**-3	4.802	5.30	4.7

Table 9. Comparison of outlier detection and removal methods

We found a higher MSE value on average for the z-score method and a lower MSE value for the 5-95 quantile method. This means that there were outliers in the training set that significantly impacted our results.

Our final test to modify the feature set was to remove other variables that had similar correlation to “sqft_living15” and check to see if any reduction lowered MSE in the validation set. Since “sqft_living15” had a correlation value of 0.76, we looked for other features with similar correlation values. We ran our tests with the learning rate of 0.15, and the quartile outlier changes.

Feature Removed	Correlation Value	MSE of Validation Set
None	---	4.44
Sqft_living15	0.762	4.47
Sqft_above	0.878	4.42
Grade	0.758	4.94
Sqft_lot15	0.774	4.44

Table 10. Testing the removal of other correlated features

We found that removing “sqft_above” reduced the MSE of the Validation set while the other raised our values.

For the Kaggle competition, we explored many of the feature engineering ideas above. My team’s name is Colton C Avila. We chose to use the 5-95 Quantile Clipping and drop the “sqft_above” variable and received a MSE of 3.717. We also tried using only the 5-95 Quantile clipping and received and MSE of 3.70. Our final and best method utilized the min-max normalization method with 5-95 Quantile clipping a received an MSE of 3.69. I learned that for some datasets you can drop features and the MSE will reduce like in Table 10, but for the full dataset this ended up hurting my final MSE.