CS534 — Implementation Assignment 1

Chih-Hsiang Wang
Jui-Hung Lu
Webster Cheng

Contribution	Chih-Hsiang Wang	Jui-Hung Lu	Webster Cheng
Percentage (%)	33.3	33.3	33.4

General Introduction:

This report analyzes the data set of house sold in Seattle. By the training data, we can use gradient descent to achieve some models with different weights on different features. Theses weights can be further applied on validation data to find the smallest SSE of price. The techniques we use are data analysis before modeling, choosing different learning rate, experiment with different regularization coefficient, and discussion about normalization. Hope after the effort on each part, we can get the desirable weights to predict the test data with small SSE.

Part 0 : Preprocessing and simple analysis. Perform the following preprocessing of the your data.

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

Because the ID feature has no impact on the price, it is just a series of numbers to label data.

(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?

We calculated the days from selling the house to 5/31/2015, trying to figure out the relation between price and the date when the house was sold. It is better to merge three separate numerical features together because they can be presented as a feature showing date to avoid the different units of year, month, and day.

(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.

training	mean	std	range
dummy	1	0	0
date	1311.1734	113.193376	390
bedrooms	3.3752	0.943246	32
bathrooms	2.118875	0.765128	7.25
sqft_living	2080.2232	911.334358	9520
sqft_lot	1.51E+04	4.12E+04	1650787
floors	1.5037	0.542647	2.5

view	0.2294	0.755932	4
sqft_above	1793.0993	830.865434	8490
sqft_basement	287.1239	435.005264	2720
yr_built	1971.1249	29.480594	115
yr_renovated	81.2267	394.379804	2015
zipcode	98078.2931	53.518391	198
lat	47.559814	0.138651	0.6217
long	-122.21329	0.141405	1.195
sqft_living15	1994.3261	691.900301	5650
sqft_lot15	12746.3234	28241.243	840540

training	waterfront	Percentage	conditional	Percentage	grade	Percentage
	1	0.70%	1	0.13%	4	0.11%
	0	99.30%	2	0.76%	5	1.05%
			3	65.30%	6	9.33%
			4	25.69%	7	41.30%
			5	8.12%	8	28.38%
					9	11.82%
					10	5.47%
					11	2.10%
					12	0.39%
					13	0.05%

validation	mean	std	range
dummy	1	0	0
date	1307.6466	113.862443	377
bedrooms	3.36591	0.905331	8
bathrooms	2.111354	0.763625	7.25
sqft_living	2073.00125	906.843038	13150

sqft_lot	1.46E+04	3.84E+04	1023459
floors	1.487315	0.536898	2.5
view	0.229945	0.766417	4
sqft_above	1784.97231	817.074895	9020
sqft_basement	288.028944	441.959342	4820
yr_built	1971.06754	29.172653	115
yr_renovated	88.08451	409.997947	2015
zipcode	98077.3121	52.97083	198
lat	47.560523	0.139036	0.6154
long	-122.21362	0.140987	1.196
sqft_living15	1977.85939	669.918412	5540
sqft_lot15	12812.61	27162.2707	434077

validation	waterfront		conditional		grade	
	1	0.79%	1	0.14%	3	0.02%
	0	99.21%	2	0.93%	4	0.16%
			3	65.10%	5	1.25%
			4	26.01%	6	9.15%
			5	7.81%	7	41.16%
					8	28.81%
					9	12.52%
					10	5.02%
					11	1.41%
					12	0.45%
					13	0.04%

(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?

[Condition, Grade]
These two features are the rating by people that it may closely correlate with price. However, the rating may be too subjective to cause the bias toward result.

[Date, Yr_built, Yr_renovated]
These features relate to the information about age of the house which may have close relationship with the price. However, one American said people here do not care about the age of houses. As a result, we keep these features in doubt for now.

[Bedrooms, Bathrooms, Sqft_living, Sqft_above, Sqft_lot, Floors, Sqft_living15]
These features present how big the house is, however, features may be dependent to each other that some of them may be not very useful in the task.

[Waterfront]
Waterfront seems to be useful to evaluate the price, but the percentage of the house with waterfront is too small for modeling. Therefore, we are not sure about this feature.

(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.

The normalization technique we use is (data-min(x)) / (max(x)-min(x)), which can transform data from each feature into value between 0 and 1.
(data is the specific number,
min(x) is the smallest value in feature x,
max(x) is the largest value in feature x)

Before normalization, the value of SSE may easily exceed the limitation if we do not set learning rate small enough, causing the result of calculated weight set as followed,
[inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf, inf]
After normalization, we can get the converged SSE even with larger learning rate.

Part 1 (30 pts). Explore different learning rate for batch gradient descent. For this part, you will work with the preprocessed 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}.

(a) Which learning rate or learning rates did you observe to be good for this particular dataset? 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.

Since gradient descent will explode when leaning rate equals to 1, 10**-1, 10**-2, 10**-3, 10**-4, we do not care about the SSE but only list out the iteration times till it explodes (SSE will increase after each iteration).

Learning rate	norm	Count	Smallest SSE (training)	Smallest SSE (validation)	Convergence
10**0	0.5	34	NA	NA	No

10**-1	0.5	44	NA	NA	No
10**-2	0.5	61	NA	NA	No
10**-3	0.5	103	NA	NA	No
10**-4	0.5	461	NA	NA	No
10**-5	0.5	146651	19520.4004291605	11541.6317051127	Yes
10**-6	5	227026	19470.1651512964	11407.2390195216	Yes
10**-7	50	537035	19523.51346	11407.24165	Yes

Based on the running time and the SSE applied on training data set by using different learning rate and weights, we found that when learning rate is 10**-6, the result is better in consideration of a tradeoff between performance and running time.

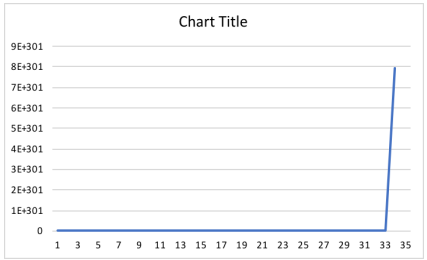
By observation from plots, the SSE of validation data for learning = 10**-5 will increase after certain iteration. As a result, we set the norm to be larger when we run the gradient descent for learning rate 10**-6 and 10**-7, which prevent the overfitting and excess running time.

(The example curves can be found in part(b))

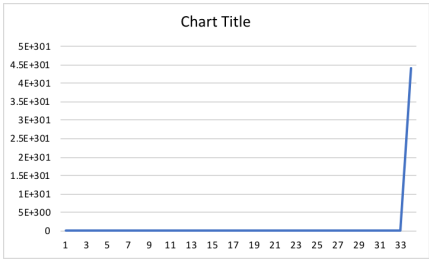
(b) For each learning rate worked for you, Report the SSE on the training data and the validation data respectively and the number of iterations needed to achieve the convergence condition for training. What do you observe?

(x axis: number of iteration, y axis: SSE)

Learning rate = 1, norm = 0.5

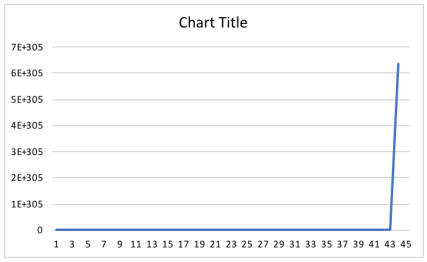


Training

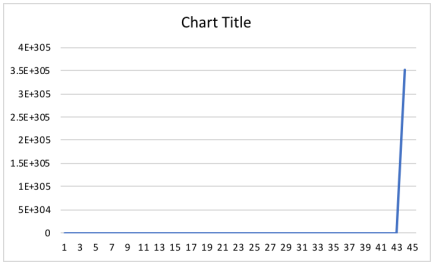


Validation

Learning rate = 10**-1, norm = 0.5

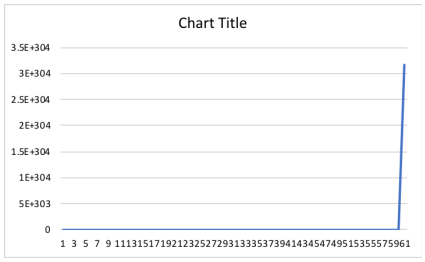


Training

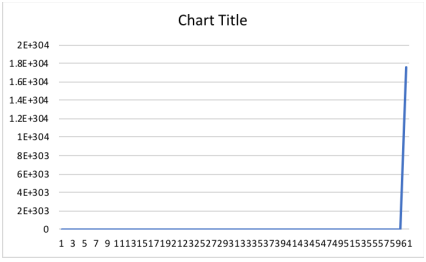


Validation

Learning rate = 10**-2, norm = 0.5

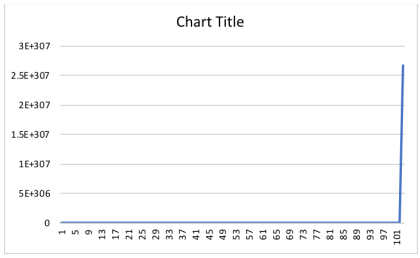


Training

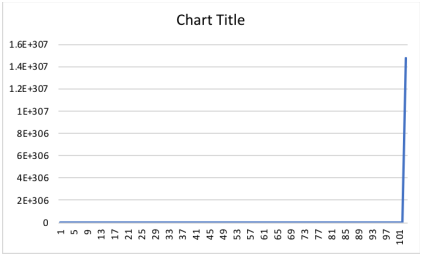


Validation

Learning rate = 10**-3, norm = 0.5

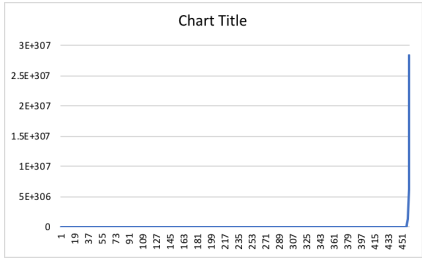


Training

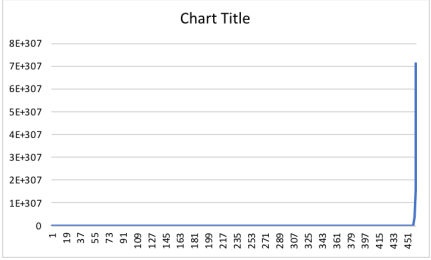


Validation

Learning rate = 10**-4, norm = 0.5

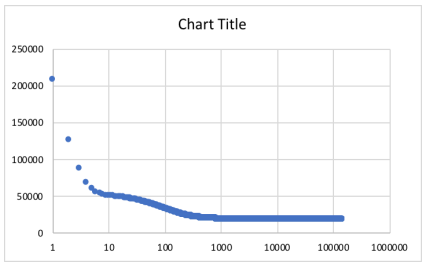


Training

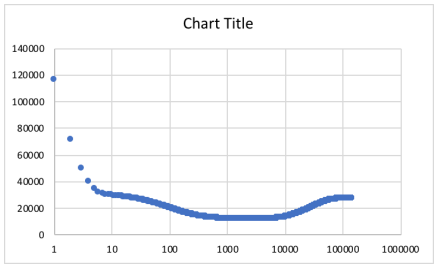


Validation

Learning rate = 10**-5, norm = 0.5

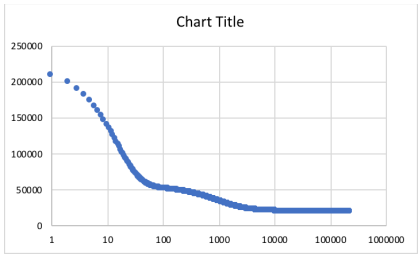


Training

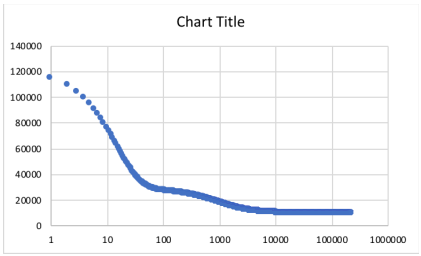


Validation

Learning rate = 10**-6, norm = 5

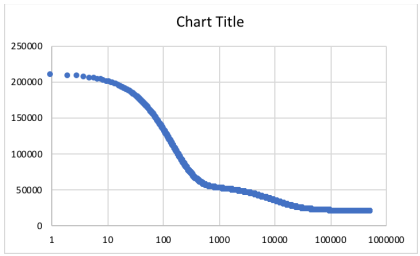


Training

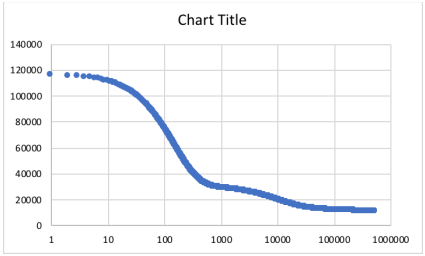


Validation

Learning rate = 10**-7, norm = 50



Training



Validation

(Number of iterations is listed in (a))
The smaller the learning rate, the smoother the line will be. However, the time till convergence increases at the same time, causing it difficult for us to wait until the convergence condition when $\epsilon < 0.5$ (It is the reason we set the number of norm larger when learning rate is 10**-6 and 10**-7). We also discovered that smaller learning rate tends to result in the model with lower SSE while running the validation dataset.

(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)).

The best converged solution we pick:

Learning rate	10**-6
Norm	5
Gradient iteration	227026
Regularization coefficient	0
SSE (training)	19470.1651512964
SSE (validation)	11407.2390195216

Name of features	Weight
dummy	-1.53769775
id	ignore
date	-0.46554161
bedrooms	-3.46240794
bathrooms	3.47822681
sqft_living	7.40660941
sqft_lot	0.49654003
floors	0.11647305
waterfront	4.67634092
view	2.31073478
condition	1.24892816
grade	8.81785241
sqft_above	7.86654823
sqft_basement	1.36909086
yr_built	-2.96392046
yr_renovated	0.28240647
zipcode	-1.02900581

lat	3.7571975
long	-2.66722725
sqft_living15	1.3214321
sqft_lot15	-1.60280724

The descending list of abs(weight) is as followed

Name of features	Weight
grade	8.81785241
sqft_above	7.86654823
sqft_living	7.40660941
waterfront	4.67634092
lat	3.7571975
bathrooms	3.47822681
bedrooms	3.46240794
yr_built	2.96392046
long	2.66722725
view	2.31073478
sqft_lot15	1.60280724
dummy	1.53769775
sqft_basement	1.36909086
sqft_living15	1.3214321
condition	1.24892816
zipcode	1.02900581
sqft_lot	0.49654003
date	0.46554161
yr_renovated	0.28240647
floors	0.11647305

[Grade] is the biggest weight which corresponds to our anticipation. But [Condition] has little impact on the price, which means it is not a reliable feature.

[Sqft_above] and [Sqft_living] are second and third largest weights in the model. [Bedrooms] and [Bathrooms] also take an important role in predicting the price. However, [Floor], [Sqft_lot], and

[Sqft_living15] are not very crucial in the model. It is because of our shortage of knowledge about housing market that we could not predict which feature is more important.

To our surprise, [latitude] and [longitude] are also important in the model. Out of curiosity, we searched the location and found the houses are located in Seattle. Thanks to the fact that one of our members came from Seattle, he explained well on how prosperous it is in different region based on different latitude and longitude.

Only [Yr_built] takes an important part in the model. [Data] and [Yr_renovated] are contrary to prediction because our misunderstanding about the market.

Part2 (30 pts). 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 don't 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:

Learning rate	λ	Count	Smallest SSE (training)	Final SSE (validation)	Convergence
10**-5	0	145983	19464.289448543874	11706.6916800997	Yes
10**-5	10**-3	145902	19464.29043790864	11706.6062921203	Yes
10**-5	10**-2	145174	19464.299787898963	11705.8412162586	Yes
10**-5	10**-1	138271	19464.433647478807	11698.5427511518	Yes
10**-5	10**0	93722	19484.8350037916	11647.6298699706	Yes
10**-5	10**1	22988	19592.53334327402	11564.8106956063	Yes
10**-5	10**2	4813	21912.5860104077	13046.3512142477	Yes

(a) What trend do you observe from the training SSE as we change λ value?

According to the table above, we can observe the training data that SSE from λ=10**-3 has smaller value than λ=10**2. This result shows that smaller λ encourages a better fit of the data. Therefore, λ=0 have the smallest SSE value in this experiment.

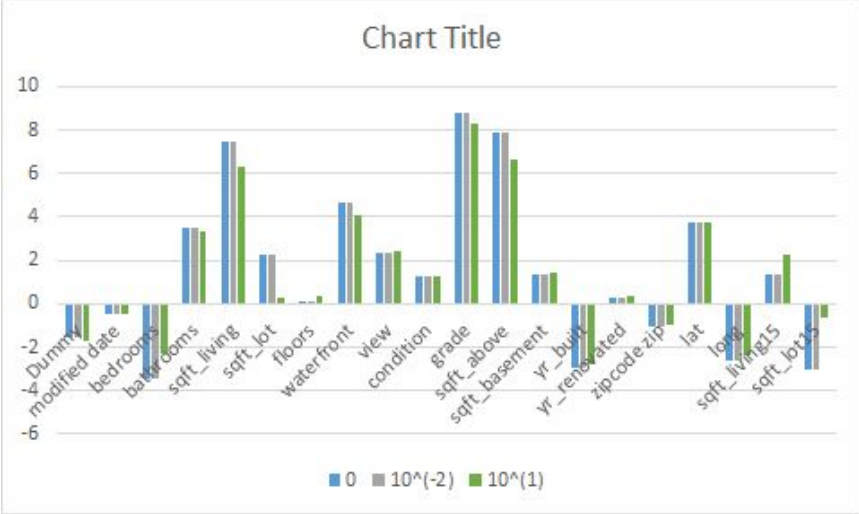
(b) What trend do you observe from the validation SSE?

According to the table above, we can observe that the smallest SSE happened with λ=10**1. This shows us that larger λ encourages simple model which not only fit in training data but also validation data. However, λ which is too big may increase SSE in testing and validation data as we can see in the example of λ=10**2.

(c) Provide an explanation for the observed behaviors.

The reason for this result is caused by the different regularization coefficient, which is also called penalty term. The linear regression uses higher λ value to increase the penalty value that will decrease the weight value changed every time. Moreover, the important feature will be more obvious in the testing data with large λ value. The regularization can resolve the problem of overfitting in the testing model. However, too large λ will modify weights too far that it may lead to underfitting.

(d) What features get turned off for λ= 10, 10−2 and 0?

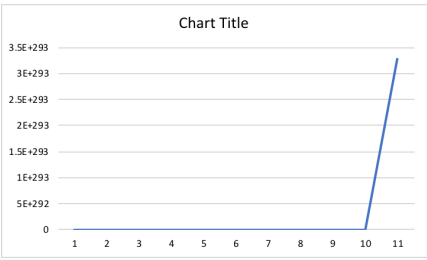


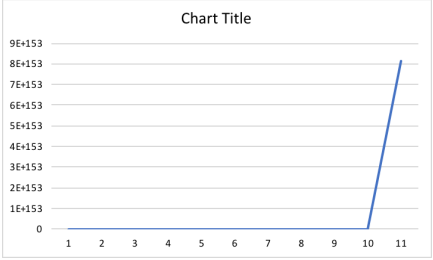
According to the chart above, we can observe that the feature *sqft_lot* and feature *sqft_lot15* are modified most for larger λ. These two features were almost turned into 0 by regularizer with larger λ. Moreover, we can observe that some features were only modified a little. However, the value in λ=10**-2 was overlarge value to regularize weight values by following the above question and data chart. Therefore, the features can be selected more appropriate to fit in this experiment.

Part 3 (10 pts). Training with non-normalized data Use the preprocessed 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?

(x axis: number of iteration, y axis: SSE)

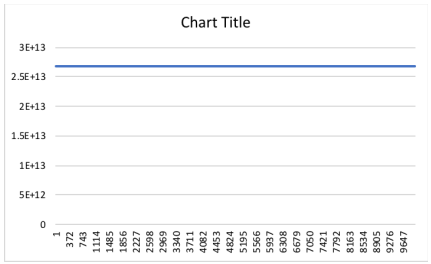
Learning rate = 1

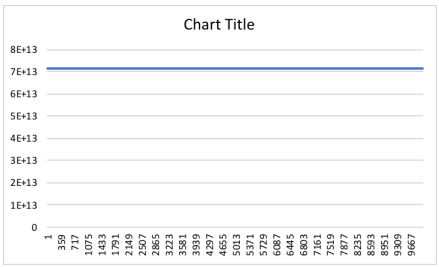




Training Validation

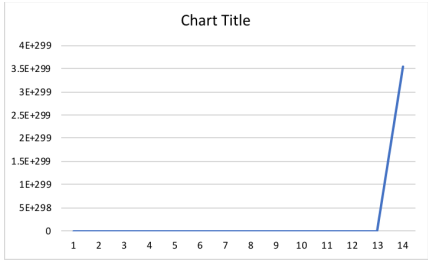
Learning rate = 0

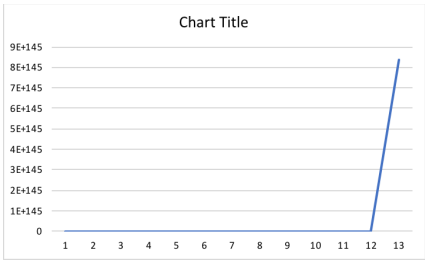




Training Validation

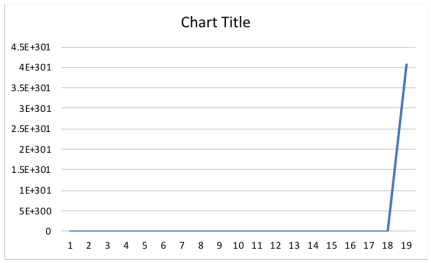
Learning rate = 10**-3

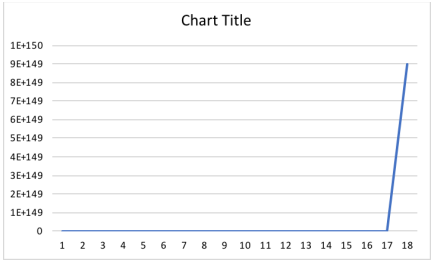




Training Validation

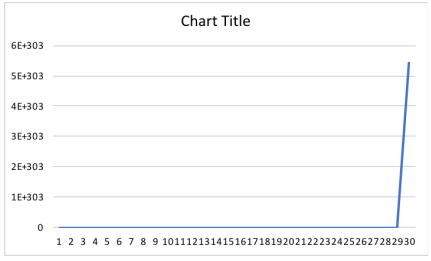
Learning rate = 10**-6

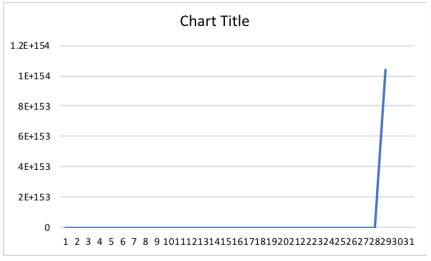




Training Validation

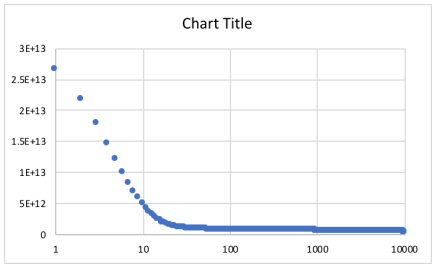
Learning rate = 10**-9



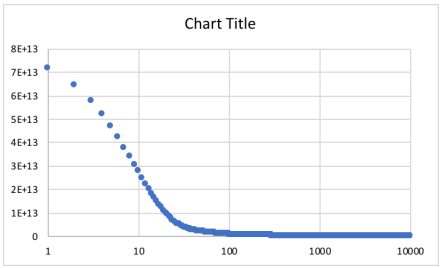


Training Validation

Learning rate = 10**-15



Training



Validation

Only when learning rate equals to 10**-15 does SSE converge (Though it does not meet the norm, it prevent the gradient descent from exploding). Without normalization, it means we need to use very small learning rate to fit the model and slow down the gradient process. Besides, the SSE is much larger than the model with normalization. As a result, dataset with normalization is easier to train for getting a smaller SSE on validation dataset.