AI5213/EC4213/ET5402/CT5303/AI4021: Machine learning and deep learning Spring 2022

Report for Coding Assignment #1

(due on Tue May. 10, 2022. 11 :59PM)

Instructor: Jeany Son
Student name (GIST ID#): Sejin Park (20175068)

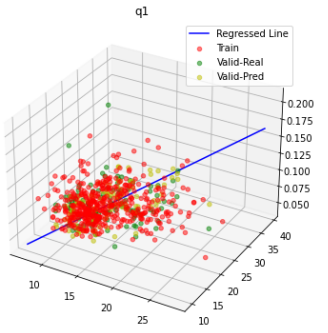
REPORT0

First of all the procedure, dataset have to initialize for regression and classification.
I divide the dataset into 8 :2 consisting training and validation sets. And, To exclude the association of the data order, I shuffle the data using random.shuffle function. then dataset become shuffle when dataset is gotten in the procedure. And I made a each two file for data, linear regression, logistic regression and perceptron to visualization and get the Accuracy. Files for visualization can show graph more easily because only the first two features are taken from the dataset. And applying the normalization to improve the data.

REPORT1

Report the error and draw the regressed line using the first two features. Discuss any ideas to reduce the errors (e.g., new feature transforms or using kernels or etc.)

First, Using the Vanilla linear regression, We can regressed the Breath cancer dataset.
To reduce the errors, apply the normalization for data range, then I can reduce 20% error. for more reducing, I tried the feature size and shape transformation but it is not efficiency. I have tried many methods, such as data processing, but we no longer find a way to reduce errors significantly. However, I guess using kernels could reduce errors even more.
Using the First two feature, I get the error and regressed line like this.
(error is change because of data random shuffling when it execute)




```
vanilla linear regression err epoch0->
8.473056657851583e-13
vanilla linear regression err epoch1->
1.7094293066484116e-12
vanilla linear regression err epoch2->
1.8436889826403894e-12
vanilla linear regression err epoch3->
1.2486181644208722e-12
vanilla linear regression err epoch4->
1.0152238726056003e-12
vanilla linear regression err epoch5->
8.291850921883365e-13
vanilla linear regression err epoch6->
1.278372890989492e-12
vanilla linear regression err epoch7->
9.459299691312808e-13
vanilla linear regression err epoch8->
1.4600223300668135e-12
vanilla linear regression err epoch9->
6.983663555900383e-13
```

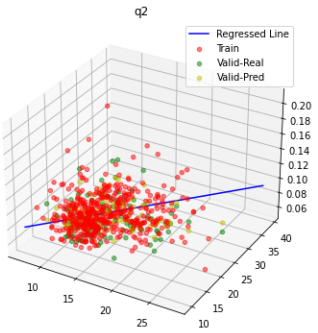
```
vanilla linear regression err epoch995->
2.6480388971673437e-12
vanilla linear regression err epoch996->
1.1817831937007348e-12
vanilla linear regression err epoch997->
1.535712012725482e-12
vanilla linear regression err epoch998->
9.830395627457316e-13
vanilla linear regression err epoch999->
7.345768561613568e-13
Error mean is 1.4624327828512444e-12
```

For 1000 epoch, Error mean is about 1.46e-12

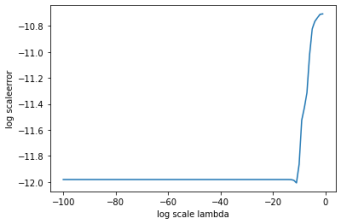
REPORT2

Report the error and draw the regressed line using the first two features. Sweep lambda from 0.0 to 5.0 (or some other reasonable values) with a reasonable sized step (e.g., 0.5), plot a graph (x-axis : lambda, y-axis : error) and discuss the effect of the lambda (especially comparing with vanilla linear when lambda=0.)

Using the First two feature, I get the error and regressed line like this.(error is change because of data random shuffling when it execute). I set the lambda = 1e-7 firstly. Error mean is about 4.01e-12
Let's discuss effect of the lambda. using the lambda we can perform the ridge regression. ridge regression is one method to regularization.
Let's Sweep the value of the lambda and observe the effect of the lambda.
I set the range 1e-100 0 first. And divide 100 step, I increase lambda 10 times each steps.




```
vanilla linear regression err epoch996->
2.859577753741483e-12
vanilla linear regression err epoch997->
3.200949053015164e-12
vanilla linear regression err epoch998->
3.237085835513879e-12
vanilla linear regression err epoch999->
4.565871224193481e-12
Error mean is 4.015684619801564e-12
```

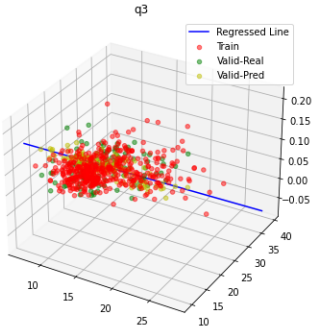


Error is constant when less than about 1e-10, have least value about 1e-10, and increase when larger than 1e-10.(graph is log10 scale) We can think this lambda value 1e-10 is most effective when using Closed form solution ridge regression.

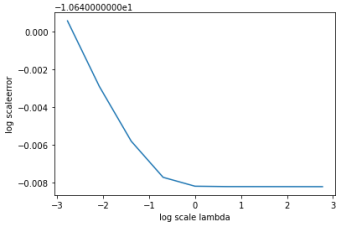
REPORT3

Report the error and draw the regressed line using the first two features. Sweep eta from 0.01 to 10.0 (or some other reasonable values) with a reasonable sized step (0.01, 0.1, etc), plot a graph (x-axis : eta, y-axis : error) and discuss the effect of the eta.

Using the First two feature, I get the error and regressed line like this.(error is change because of data random shuffling when it execute. So, execute 20 times, measure the mean Error) I set the lambda to 1e-10 what we observe in the Report2 and eta to 1.0 firstly. And Let's Sweep the eta and observe the effect of the eta. I set the range of the eta pow(2,-4) to pow(2,5). And, I divide range to 10 step, and each step eta multiply by 2. Error mean is about 1.87e-1112




```
1.640886144216979e-11
vanilla linear regression err epoch15->
1.6446572172229308e-11
vanilla linear regression err epoch16->
1.90646171382782e-11
vanilla linear regression err epoch17->
2.2015906056746862e-11
vanilla linear regression err epoch18->
2.0981127715476058e-11
vanilla linear regression err epoch19->
1.9280582743247585e-11
Error mean is 1.8772166155955912e-11
```



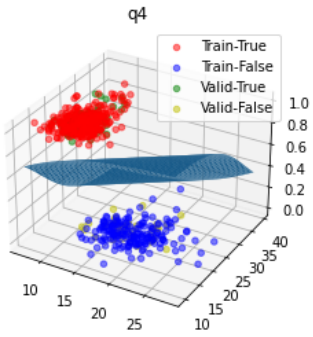
Let's discuss effect of the eta.
When the eta is smaller than 1(that graph is log10 scale) error is decrease, Start from the 1, error is constant. then we can choose eta larger than 1. however, if we choose too larger value as a eta, it cause the overflow. So we choose eta carefully. (I choose the 1 as a eta value)

REPORT4

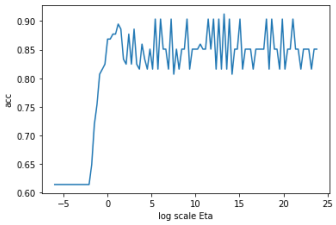
Report the error of your logistic regression model and draw the decision boundary using the first two features. Sweep eta from 0.01 to 10.0 (or some other reasonable values) with a reasonable sized step (0.01, 0.1, etc), plot a graph (x-axis : eta, y-axis : acc) and discuss the effect of the eta.

Using the First two feature, let get the accuracy and decision boundary.(accuracy is change because of data random shuffling when it execute, so, the average accuracy was calculated by repeating it 20 times.) I set the eta is 1 firstly. Acc average is 0.895

Sweep the value of the eta and observe the effect of the eta. I set the range pow(2,-20) to pow(2,80) and divide the range to the 100 steps. Eta value starting from the pow(2,-20) increase by multiply 2 each steps. When Eta value is




```
Logistic Regression(GA) Acc epoch16->
0.8947368421052632
Logistic Regression(GA) Acc epoch17->
0.9035087719298246
Logistic Regression(GA) Acc epoch18->
0.8947368421052632
Logistic Regression(GA) Acc epoch19->
0.9035087719298246
Acc mean is 0.8951754385964911
```

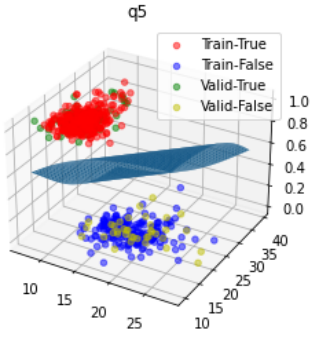


smaller than about 1 2, Accuracy is increased following the Eta increasing. However, when Eta is larger than 1 2, Accuracy is vibration. So we choose the Eta value 1.

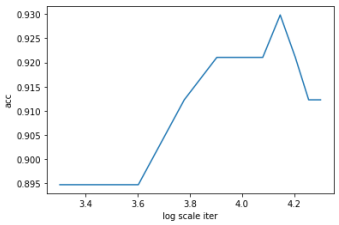
REPORT5

Report the error of your logistic regression model and draw the decision boundary using the first two features. Using the etayou found from REPORT4, run with different numbers of iterations, plot a graph (x-axis : eta, y-axis : acc) and discuss the effect of the number of iterations.

Using the First two feature, let's get the accuracy and decision boundary.(error is change because of data random shuffling when it execute, so, the average was calculated by repeating it 20 times.) I set the eta to 1 what is find in Report4 and iteration to 1000. Then Average accuracy is 0.8894
Let's discuss effect of the iteration.




```
0.8771929824561403
Logistic Regression(SGA) Acc epoch16->
0.8333333333333334
Logistic Regression(SGA) Acc epoch17->
0.8947368421052632
Logistic Regression(SGA) Acc epoch18->
0.8859649122807017
Logistic Regression(SGA) Acc epoch19->
0.9210526315789473
Acc mean is 0.8894736842105264
```



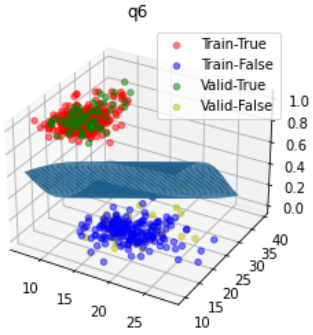
first, I set the range of iteration 2000 to the 200000. range is divided by 10step. between the each step, max iteration value increase 2000. Accuracy is increased following iteration increasing when before the about 14000. Starting from the about 14000, Accuracy is decreased following the iteration increasing. It may be cause by overflow and over fitting.

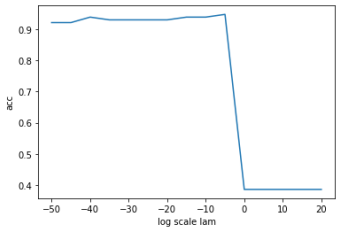
REPORT6

Report the error of your regularized logistic regression model and draw the decision boundary using the first two features. Sweep lambda from 0.0 to 5.0 (or some other reasonable values) with a reasonable sized step (e.g., 0.5), plot a graph (x-axis : lambda, y-axis : acc) and discuss the effect of the lambda (especially comparing with vanilla logistic when lambda=0.)

Using the First two feature, let's get the accuracy and decision boundary.(error is change because of data random shuffling when it execute, so, the average was calculated by repeating it 20 times.) I set the eta to 1 what is find in Report4 and iteration to 14000 what is find in Report5.

Let's discuss effect of the lambda. lambda is feature of the Regularization. So let see the effective of this lambda. I set the range of the lambda 1e-50 to 1e+14. I divide this range to 15 steps.(multiply by 1e5) Look at the last feature. The accuracy is increase smoothly following lambda increasing before the about lambda 1. After lambda 1, Accuracy is decrease remarkably. It may related to over fitting. if it is too larger than it can cause the over fitting issue.






```
Logistic Regression(SGA) Acc epoch16->
0.9035087719298246
Logistic Regression(SGA) Acc epoch17->
0.9035087719298246
Logistic Regression(SGA) Acc epoch18->
0.8596491228070176
Logistic Regression(SGA) Acc epoch19->
0.9035087719298246
Acc mean is 0.906578947368421
```

So we choose the lambda carefully. Let choose the 1e-7 as the lambda. Let measure the Accuracy when eta is 1, ite is 14000 and lambda is 1e-7. And what happen if lambda is 0 ? if lambda is zero than it is same to the vanilla logistic regression. However,these two values are sometimes the same and sometimes different. this is judged to be the influence of stochastic Gradient ascent

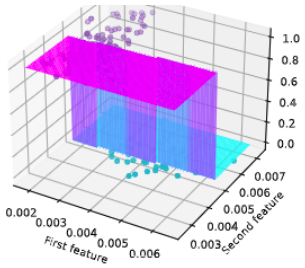
```
vanilla logistic regression
0.9035087719298246
Regularized Logistic Regression
0.9035087719298246
```

```
vanilla logistic regression
0.9298245614035088
Regularized Logistic Regression
0.9473684210526315
```

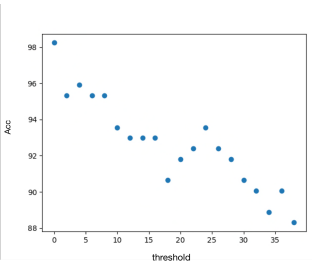
REPORT7

Report the error of your perceptron and draw the decision boundary using the first two features. Using different values for threshold, plot a graph (x-axis : thresh, y-axis : acc) and discuss the effect of the threshold. Discuss what happens when 'threshold=0' and why.

Using the First two feature, I get the accuracy and decision boundary. (accuracy is change because of data random shuffling when it execute, so I take the average of the accuracy trough the 20 times repetition.) Unlike what we saw earlier in logistic , perceptron can be seen that the boundary consists of layers, which is a characteristic of the boundary of perceptron.(like quantization) I set the feature eta to 1 lam to 0, iter to 1000, threshold to 0. then average accuracy is 92.36 like second figure.




```
Perceptron Acc epoch16->
0.9298245614035088
Perceptron Acc epoch17->
0.9473684210526315
Perceptron Acc epoch18->
0.9298245614035088
Perceptron Acc epoch19->
0.9122807017543859
Acc mean is 0.9236842105263159
```

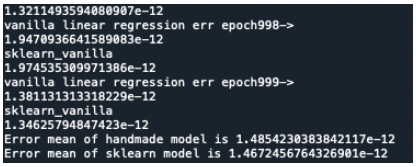


Sweep the value of the threshold and observe the effect of the threshold. I set the range of the threshold 0 to 38 and divide to 20 step. per steps, thresholds is increased by adding 2.
As show, Following the increasing the thershold, accuracy is decrease. it is because when threshold increasing, Accuracy decreases as it is often judged to be zero. variation of accuracy is maybe cause by stochastic procedure in perceptron predict. when the threshold 0, it has the best performance

REPORT8

Compare the error by your implementations of vanilla linear regression and OLS model in scikit-learn and discuss the reason for the difference. If they are identical, report and claim you're awesome!

Let's compare the handmade vanilla linear model and scikit-learn model. Let measure two model's error average during the 1000 repetition.

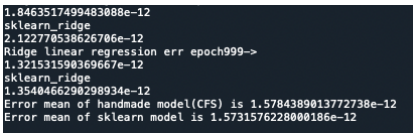


error is very similar to the scikit-learn model. Moreover, error is very small. it is maybe benefit to normalization of dataset and good structure of vanilla regression function

REPORT9

Compare the error by your implementations of ridge regression and ridge regression model in scikit-learn and discuss the reason for the difference. If they are identical, report and claim you're awesome !

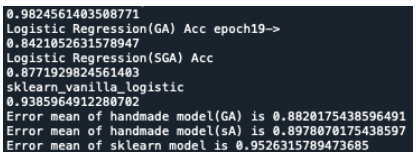
Next, Let's compare the handmade Ridge linear model and scikit-learn model. Let measure two model's error average during the 1000 repetition.Before the start, lambda is set to 1e-10 what we find out the in Report2 trough the sweeping



error is very similar to the scikit-learn model.(more vanilla linear regression model) Moreover, error is very small. it is maybe benefit to normalization of dataset and good structure of ridfe regression function. And tuning the parameter

REPORT10

Compare the error by your implementations of logistic regression and logistic regression model in scikit-learn and discuss the reason for the difference. If they are identical, report and claim you're awesome !
Compare the logistic regression model(REPORT4,5) and sklearn logistic model. Let measure the accuracy average during the 20 repetition. To start, eta set to 1 and iter set to 14000(according to result of Report 4,5). Average accuracy during the 20 repetition is like this

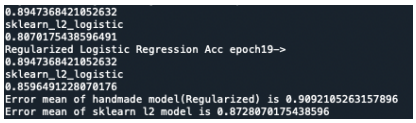


Compared to the sklearn model, the my 2 model has a 5% lower accuracy. To improve this difference, I tried to many way(Efforts were made to rebuild the model and touch the bias.) But I cannot improve this difference. And I tried to obtain better values through regularization and compare them, but there was no significant improvement. This difference cause by the difference between How sklearn and my model works. My model can adopt the eta value, sklearn cannot use the eta value when trainning the model The point is that the accuracy of the reguarized model is too lower than the accuracy of the non-reguarized model. Result of regularized model will show Report11, but it is strange that the regularized model is rather much less accurate. I thought it might be immature for me to use Sklearn, so I looked for a solution but couldn't find an answer.

REPORT11

Compare the error by your implementations of l2-regularized logistic regression and l2-regularized logistic regression model in scikit-learn and discuss the reason for the difference. If they are identical, report and claim you're awesome !

let's compare the L2 regularized logistic regression model(REPORT6) and sklearn L2 regularized logistic model. I measured the accuracy average during the 20 repetition. To start, eta set to 1, iter set to 14000 and lambda set to 1e-7(according to result of Report 4,5,6). Average accuracy during the 20 repetition is like this

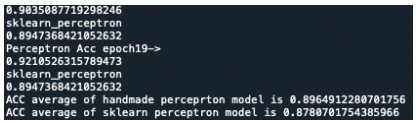


Although there's not much difference, Contrary to the results of Report10, this time the accuracy of the self-made model is higher than that of the sklearn model. It is not clear whether this is a better outcome. However, I tried to increase accuracy by accurately implementing L2 regularization. Stochastic was implemented as a shuffle. In addition, I tried to achieve higher accuracy by tuning the feature.

REPORT12

Compare the error by your implementations of perceptron and perceptron model in scikit-learn and discuss the reason for the difference. If they are identical, report and claim you're awesome!

Last of all, Let compare the my perceptron model and sklearn perceptron model. I set the threshold to 0 and eta 1 and iter to 1000



Similar to Report 11, Although there's not much difference,the accuracy of the self-made model is higher than that of the sklearn perceptron model. Likewise, I don't know if this is a good result. However I tied to build perceptron work well and has a high accuracy. Difference thing Between the my model and sklearn model is sklearn perceptron model can modify the eta, but my model cannot. And threshold can modify in my model, but sklearn model cannot(I tried to find this, I couldn't change it within the limits I found.).This difference thing maybe make the difference accuracy between the two model