CS534

Implementation assignment 2

Name: Dayeon Oh

Student ID: 933-953-476

Part 1.

(a)

```
Training with reg_param 1
Time took : 187.1476171016693 with iteration count : 10000
Accuracy of training data with reg_param 1 is : 76.84012078413829
Accuracy of validation data with reg_param 1 is : 76.83998832268318
Training with reg_param 0.1
Time took : 187.63738322257996 with iteration count : 10000
Accuracy of training data with reg_param 0.1 is : 78.6774456391494
Accuracy of validation data with reg_param 0.1 is : 78.64997242855752
Training with reg_param 0.01
Time took : 188.5421221256256 with iteration count : 10000
Accuracy of training data with reg_param 0.01 is : 78.83401767027209
Accuracy of validation data with reg_param 0.01 is : 78.81864478250998
Training with reg_param 0.001
Time took : 127.27791380882263 with iteration count : 6744
Accuracy of training data with reg_param 0.001 is : 76.93598121135626
Accuracy of validation data with reg_param 0.001 is : 76.96324888903305
Training with reg_param 0.0001
Time took : 188.86287903785706 with iteration count : 10000
Accuracy of training data with reg_param 0.0001 is : 76.39916281893562
Accuracy of validation data with reg_param 0.0001 is : 76.38587044665736
Training with reg_param 1e-05
Time took : 143.44635367393494 with iteration count : 7588
Accuracy of training data with reg_param 1e-05 is : 76.67875573165472
Accuracy of validation data with reg_param 1e-05 is : 76.65185377404391
```

Figure 1. Algorithm 1 implementation with w = np.random.normal()

```
Training with reg_param 1
Time took : 186.63710713386536 with iteration count : 10000
Accuracy of training data with reg_param 1 is : 76.84012078413829
Accuracy of validation data with reg_param 1 is : 76.83998832268318
Training with reg_param 0.1
Time took : 187.1663990020752 with iteration count : 10000
Accuracy of training data with reg_param 0.1 is : 78.67584796536244
Accuracy of validation data with reg_param 0.1 is : 78.65645982678646
Training with reg_param 0.01
Time took : 187.0827076435089 with iteration count : 10000
Accuracy of training data with reg_param 0.01 is : 78.99058970139477
Accuracy of validation data with reg_param 0.01 is : 79.03597262317948
Training with reg_param 0.001
Time took : 186.98538875579834 with iteration count : 10000
Accuracy of training data with reg_param 0.001 is : 79.25740122381812
Accuracy of validation data with reg_param 0.001 is : 79.33114924259625
Training with reg_param 0.0001
Time took : 186.50583028793335 with iteration count : 10000
Accuracy of training data with reg_param 0.0001 is : 79.28456167819654
Accuracy of validation data with reg_param 0.0001 is : 79.40575432222907
Training with reg_param 1e-05
Time took : 186.45459699630737 with iteration count : 10000
Accuracy of training data with reg_param 1e-05 is : 79.28615935198351
Accuracy of validation data with reg_param 1e-05 is : 79.4025106231146
```

Figure 2. Algorithm 1 implementation with w = np.zeros()

Implemented the model based on the pseudo code of Algorithm 1. Figure1, for early stopping condition, I set the epsilon value to be 0.0002, and maximum iteration to be 10,000. Also, I set the initial weight vector with np.random.normal(0.0, 1.0, size = X.shape[1]). With the regularization parameter of 0.001 and 1e-05, the training ended with less than 10,000 iterations and others went through 10,000 iterations. In Figure 2, I set the set the initial weight vector with zero values. Also, I did not set the epsilon value and did the 10,000 iterations for all regularization parameter values. In this case, validation accuracy gets higher as the lambda value gets lower.

(b)

(1) Plot the training accuracy and validation accuracy of the learned model as the λ value varies.

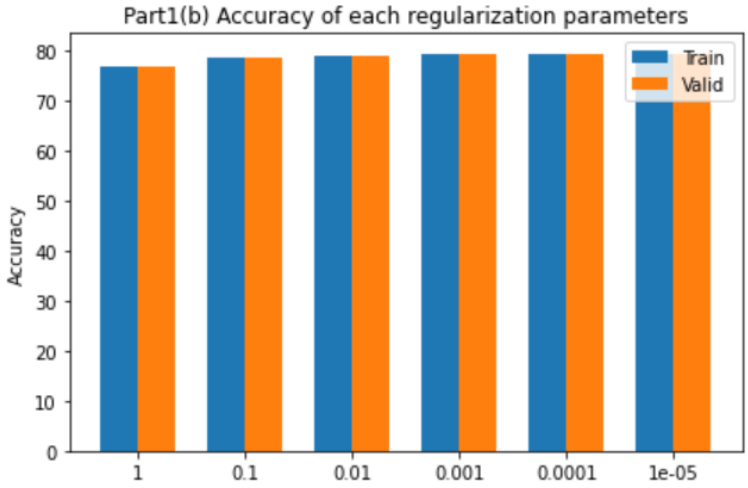

Figure 3. Accuracy of L2 regularization

Plotted result of training dataset accuracy and validation dataset accuracy regarding different regularization parameters. Below is the exact accuracy of training dataset and validation dataset.

```
train accuracy :  [76.84012078413829, 78.67584796536244, 78.99058970139477, 79.2
5740122381812, 79.28456167819654, 79.28615935198351]
validation accuracy :  [76.83998832268318, 78.65645982678646, 79.03597262317948,
 79.33114924259625, 79.40575432222907, 79.4025106231146]
{1: 76.84012078413829, 0.1: 78.67584796536244, 0.01: 78.99058970139477, 0.001: 7
9.25740122381812, 0.0001: 79.28456167819654, 1e-05: 79.28615935198351}
{1: 76.83998832268318, 0.1: 78.65645982678646, 0.01: 79.03597262317948, 0.001: 7
9.33114924259625, 0.0001: 79.40575432222907, 1e-05: 79.4025106231146}
```

Figure 4. Exact accuracy values

(2) What trend do you observe for the training accuracy as we increase λ? Why is this the case?

As we can see in the figure 2, the training accuracy gets lower as the regularization parameter increases. Since the regularization is a method to reduce or avoid the overfitting by adding penalty to the high valued regression features. So, by doing regularization the model reduces the parameter

and it will become simple, which will likely to perform better. As the regularization parameter gets smaller it will help the model to have minimize the cost function and it will perform better accuracy, so the accuracy of training dataset gets better as the lambda value gets smaller.

(3) What trend do you observe for the validation accuracy? What is the best λ value based on the validation accuracy?

As we can see from the figure2, until the regularization parameter set to 0.0001, the accuracy of validation dataset increase as the regularization parameter decreases. This is because with the larger regularization parameter it pays less attention to prior training and instead focus more on data, which could lead to an underfitting. When the lambda value is large, the model will become simple and the model will not learn enough about the training data so it would have lower accuracy. When the regularization parameter decreases less than 0.0001 the validation accuracy gets slightly lower than before. This is because the as the lambda value get too low, which will make the model very complex and it could lead to an overfitting. Therefore, setting an appropriate regularization parameter is important in training the dataset. In this case for validation dataset regularization value in 0.0001 shows the best accuracy.

(c)

(1) What are the top 5 features that are considered important according to the learned weights?

Based on the result in figure1 and 2, I chose the model with regularization parameter 0.0001 to be the best model. The top 5 features that are considered important according to the learned weights are in figure 5. They are 'Previously Insured', 'Vehicle Damage', 'Policy Sales Channel 152', 'Age' and Policy Sales Channel 160'.

```
[('Previously_Insured', -2.2694037979324326),
 ('Vehicle_Damage', 1.8673457896853058),
 ('Policy_Sales_Channel_152', -0.5694414615415085),
 ('Age', -0.5526297744981173),
 ('Policy_Sales_Channel_160', -0.548320555696633),
```

Figure 5. Top 5 important features

(2) How many features have $w_j = 0$? If we use larger λ value, do you expect more or fewer features to have $w_j = 0$?

Table 1 below shows the number of zero weight value according to the λ value. For the weight values, there were three zero values for all regularization parameters. For L2 regularization, unlike the L1 regularization, it shrinks the feature parameter instead of making it to zero, so there are only

three zeros values and the number of zero values does not change as the regularization parameter varies. Therefore, even if we increase the λ value, there will be no change in the number of features with $w_j = 0$.

λ value	1	0.1	0.01	0.001	0.0001	1e-05
Non-zero	193	193	193	193	193	193
Zero	3	3	3	3	3	3
W norm	0.2029	1.006	2.3751	3.104	3.2124	3.2237

Table 1. Weight norm of L2 regularization

Part 2.

(a)

```
Training with reg_param 1
Time took : 182.72796654701233 with iteration count : 10000
Accuracy of training data with reg_param 1 is : 50.08707322138966
Accuracy of validation data with reg_param 1 is : 49.823218398261375
Training with reg_param 0.1
Time took : 205.9682161808014 with iteration count : 10000
Accuracy of training data with reg_param 0.1 is : 77.15006949880974
Accuracy of validation data with reg_param 0.1 is : 77.07029095981058
Training with reg_param 0.01
Time took : 210.78764629364014 with iteration count : 10000
Accuracy of training data with reg_param 0.01 is : 77.89139013596204
Accuracy of validation data with reg_param 0.01 is : 78.10827467644101
Training with reg_param 0.001
Time took : 174.9314775466919 with iteration count : 8234
Accuracy of training data with reg_param 0.001 is : 77.27309038040613
Accuracy of validation data with reg_param 0.001 is : 77.13516494209998
Training with reg_param 0.0001
Time took : 172.80158638954163 with iteration count : 8139
Accuracy of training data with reg_param 0.0001 is : 76.84970682686009
Accuracy of validation data with reg_param 0.0001 is : 76.95027409257517
Training with reg_param 1e-05
Time took : 185.22830200195312 with iteration count : 8729
Accuracy of training data with reg_param 1e-05 is : 77.07657650460928
Accuracy of validation data with reg_param 1e-05 is : 77.02812287132245
```

Figure 6. Algorithm 2 Implementation w = np.random.normal()

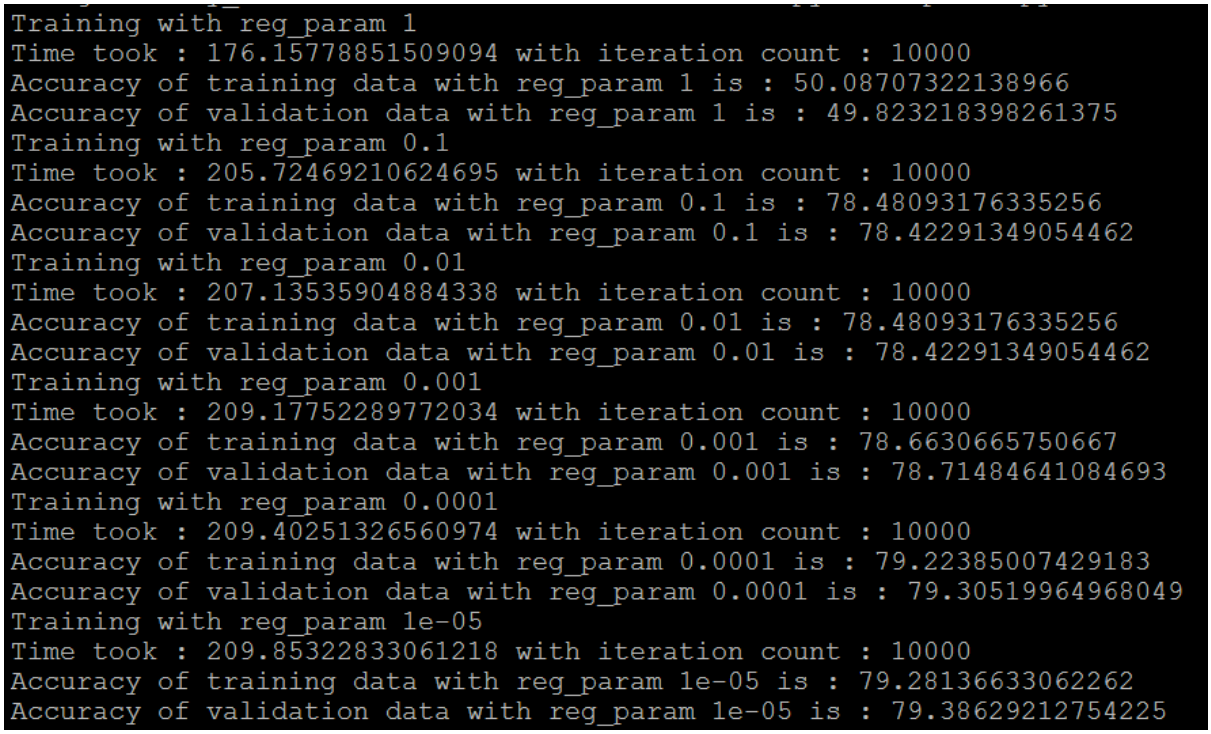

Figure 7. Algorithm 2 Implementation w = np.zeros()

Implemented the model based on the pseudo code of Algorithm 2. In both figure 6 and 7, I set the early stopping condition to maximum iterations to be 10,000 and just for figure 6 I set epsilon value to be 0.0002. In figure 6, I set the initial weight vector to be random normal value that follows gaussian distribution. The validation accuracy goes higher until the regularization parameter gets lower until 0.01 and decrease slightly after 0.01. However, in figure 7, where I set the initial vector to zero vector it shows constant increase in validation accuracy.

(b)

(1) Plot the training accuracy and validation accuracy of the learned model as the λ value varies.

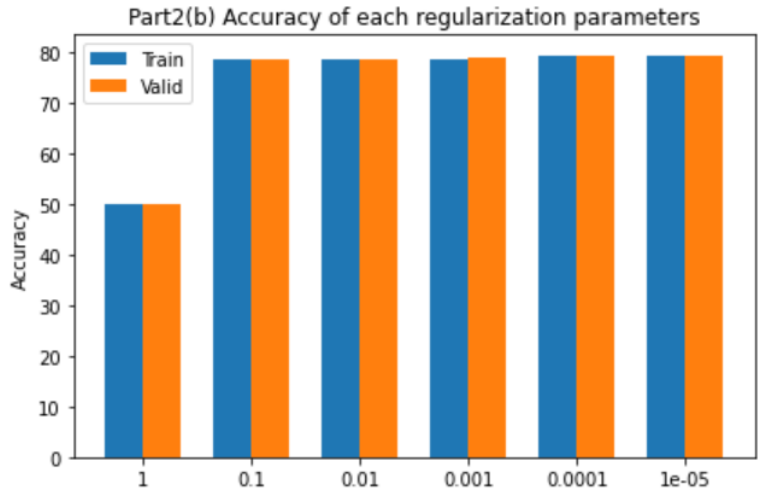

Figure 8. Accuracy of L1 regularization

The figure 8 is a plotted result of training and validation accuracy according to the regularization parameters. Below figure is the exact value of training dataset accuracy and validation dataset accuracy.

```
train accuracy :  [50.08707322138966, 78.48093176335256, 78.48093176335256, 78.4361968973175, 78.48732245850042, 78.5208736080267]
validation accuracy :  [49.823218398261375, 78.42291349054462, 78.42291349054462, 78.57861104803918, 78.68889681793117, 78.68889681793117]
{1: 50.08707322138966, 0.1: 78.48093176335256, 0.01: 78.48093176335256, 0.001: 78.4361968973175, 0.0001: 78.48732245850042, 1e-05: 78.5208736080267}
{1: 49.823218398261375, 0.1: 78.42291349054462, 0.01: 78.42291349054462, 0.001: 78.57861104803918, 0.0001: 78.68889681793117, 1e-05: 78.68889681793117}
```

Figure 9. Accuracy of training and validation data

(2) What trend do you observe for the training accuracy as we increase λ? Why is this the case?

As we can see in figure 7, the training accuracy tends to increase as the regularization parameter gets lower. With the regularization parameter 0.001, training accuracy decreases extremely slightly than 0.01, but as the lambda value gets more lower training accuracy increases again. With the help of regularization parameter, we can add certain value to cost and delay the fitting period, and it will prevent the model from getting too complicated. The goal of this model training is to lower the cost function's value, and to minimize the value of cost function, regularization parameter should be small. Therefore, as the lambda value gets smaller the accuracy gets higher. Since, as the lambda value gets smaller training accuracy gets higher, so does the validation accuracy.

(3) What trend do you observe for the validation accuracy? What is the vest value based on the validation accuracy?

In figure 7 and plot 8, we can see that validation data gets higher when the lambda value gets lower. Therefore, we can say that lambda value 1e-05 is the best value based on both validation accuracy and training accuracy.

(c)

(1) What are top 5 features that are considered important?

In (b), we observed that the regularization parameter value with 1e-05 shows the highest accuracy in validation data. So, I chose the weighted value result lambda value 1e-05. Below figures is the top 5 features that are considered important in this process. The top 5 important features are 'Previously Insured', 'Vehicle Damage', 'Policy Sales Channel 152', 'Age' and Policy Sales Channel 160'. If we compare this value with the result from part 1, we can see that these top 5 important features are same. I want to interpret this result that both L1 and L2 regression derives almost similar result.

```
[('Previously_Insured', -2.278644038744914),
 ('Vehicle_Damage', 1.8730750524612143),
 ('Policy_Sales_Channel_152', -0.5709917639411921),
 ('Age', -0.5550426707671976),
 ('Policy_Sales_Channel_160', -0.5502378890900914),
```

Figure 10. Top 5 features from lambda value 1e-05

(2) How many features have $w_j = 0$? If we use larger λ value, do you expect more or fewer features to have $w_j = 0$?

Also, the table 2 below shows norm of each weights and the number of non-zero and zero values.

λ value	1	0.1	0.01	0.001	0.0001	1e-05
Zero	196	194	191	171	115	40
Non-Zero	0	2	5	25	81	156
W norm	0.0	0.5168	2.5258	3.0913	3.2077	3.2231

Table 2. Zero, Non-zero and norm values of L1 regularization

Unlike L2 regularization, L1 regularization limits the feature size by changing the coefficient to zero, and it will simplify the model and generate sparser model. As we can see from the table, as the lambda value gets lower, the number of non-zero value gets higher. More smaller lambda value will let the features not turned into a zero since it will have smaller threshold. Therefore, we can expect more features to have $w_j = 0$ as the regularization parameter lambda gets higher.

(d)

If we compare the part1 and part2 result, the final validation accuracy of part 1 is 79.40 and final validation accuracy of part 2 is 79.38. They are almost the same and the time took to get this result

is 186 and 209. For this difference, I implemented the L1 regularization with for loop for updating the weight value.

With the table 1 and 2, we can see the zero and non-zero values in L1 regularization and L2 regularization. In table 1, for all regularization parameter zero and non-zero value are equal to 3 and 196 where we update the weight value with learning rate and regularization parameter, but in table 2, the non-zero value is much less than part 1 result. It is because in Lasso regularization, we choose the feature as zero if that features are not important. I think this regularization will work well with larger feature selection. Since the method of feature selection is different, it shows the different amount of zero values in L1 and L2 regularization.

Even though the method of feature selection varies from L1 and L2, it shows the almost same result in accuracy result. I want to interpret this result as L1 and L2 shows the same performance.