CSE 353
HomeWork 5

JeongYoon Lee (114133199)

a) Introduction. Brief summary of what you think the assignment is about

This assignment is about multi class classification using One-Versus-All(OVA) strategy and One-Versus-One(OVO) strategy. The given data X (3*80) contains 80 training dataset of dimension 3. And the ground truth value Y (1*80) is a row vector containing multi class labels which is {1,2,3,4}.

We can make classifier with some algorithm like PLA, linear regression, and logistic regression, and since our multi-class is categorical data, so it is better to use logistic regression since it calculate with proportion, not a value.

After making classifier, we can classify the class using this classifier, and the way to decide predicted labels is OVA and OVO. They all have pros and cons, so I will try to think which strategy is better for what dataset. Also, I can learn about hard-voting and soft-voting through this experiment.

To decide the class among similar dataset, many factors should considered in each same classes dataset. So, it is quite hard to decide only with a short decision or with simple factor. I think it will be the similar error rate if we do enough iteration, but I think there will be some differences in iteration time, training time and cost.

b) Method. Brief outline of your (algorithmic) approach

- One-Versus-All (OVA)

This will compare one class and other classes at once, and check the most biggest g(x) among each class classifiers, and that can decide the class of that data.

For k $\in$ Y (Y = {1,2,3,,,k})

$$D[k] = \{\left(xn, y'n = 2\llbracket yn = k\rrbracket - 1\right\}, \llbracket is\ indicator\ function$$

(xn , yn) → (xn , y'n) , yn $\in$ Y, y'n $\in$ Y′ = {+1, -1}

Return g(x) = argmax(k){ sigmoid(W(k).T x) }

We can get W through PLA, linear regression or logistic regression, but since our data is Categorical Data, it is best to use logistic regression to get optimal W value.

- One-Versus-One (OVO)

  This will compare one class and one of other classes at once, kC2 times. And then, we can decide using tournament champion method with soft voting.

  For (k,l) $\in$ Y*Y

  $$D[k] = \{(xn, y'n = 2\llbracket (yn = r) \rrbracket - 1\}, \llbracket is\ indicator\ function, and\ r = k\ or\ r = l \}$$

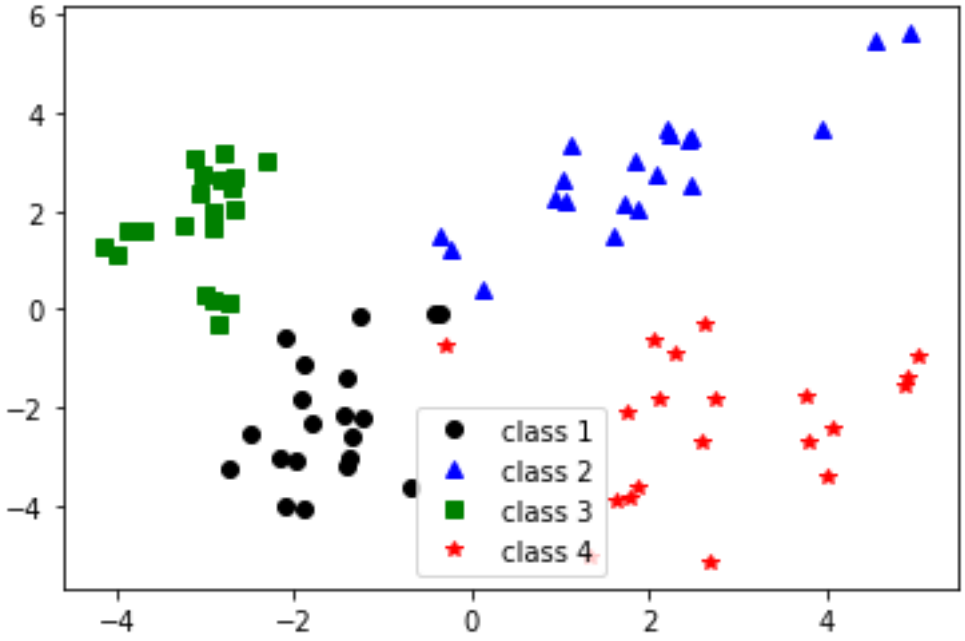
  (xn , yn) → (xn , y'n) , yn $\in$ Y, y'n $\in$ Y′ = {+1, -1}

  Return g(x) = argmax(k){ tournament champion(W(k,l).T x) }

  Similarly, we can get W through PLA, linear regression or logistic regression. The difference between OVA and OVO is how to get g(x). In this OVO case, we need kC2 classifiers, but it will have less training time since it only train each 2 classes.

c) Experiments. Tables and/or pictures of intermediate and final results that convince us that the program does what you think it does.

1. Check original data distribution



[Original data class distribution]

➔ Can check the distribution of original dataset in this plot.

➔ It seems some datapoint of class 1, class 2 and class 4 around the origin are similar.

➔ So, I think classifying those datapoints in the middle is the most important part for this classifier (around origin point).

2. One-Versus-All (OVA) using logistic regression

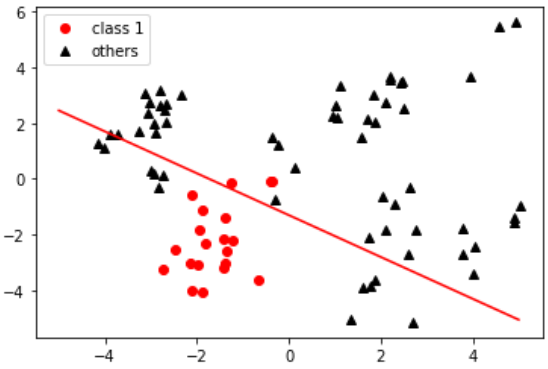
I tried with two different iteration number to check which model between using OVA and OVO is more efficient even with a smaller iteration number.

So, for the first OVA model, I scaled learning rate as 0.3 and iteration number as 10.
And then, I changed iteration number as 1000 for the second model.

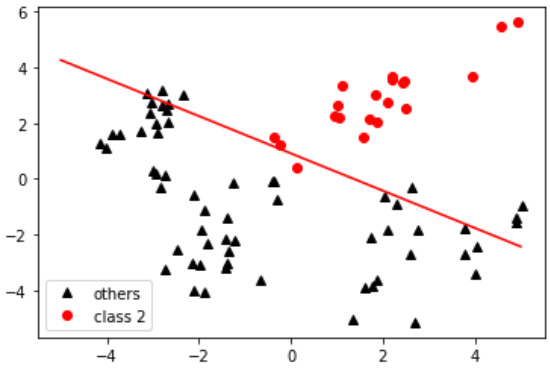
1) First OVA model with logistic regression (learning rate : 0.3, iteration number : 10)

```
---class 1---
learning rate : 0.3, Iteration : 10
>>W_logisticRegression :  [-0.60074013 -0.34633614 -0.46152083]

---class 2---
learning rate : 0.3, Iteration : 10
>>W_logisticRegression :  [-0.57883505  0.4235541   0.63324912]

---class 3---
learning rate : 0.3, Iteration : 10
>>W_logisticRegression :  [-0.61649379 -0.66229669  0.39662046]

---class 4---
learning rate : 0.3, Iteration : 10
>>W_logisticRegression :  [-0.58958731  0.62259432 -0.49776097]
```

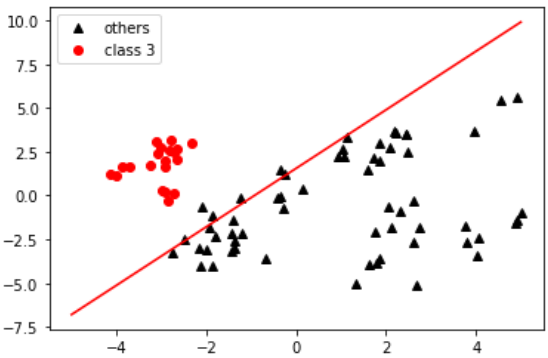
[Result of First model and Weight from logistic regression]



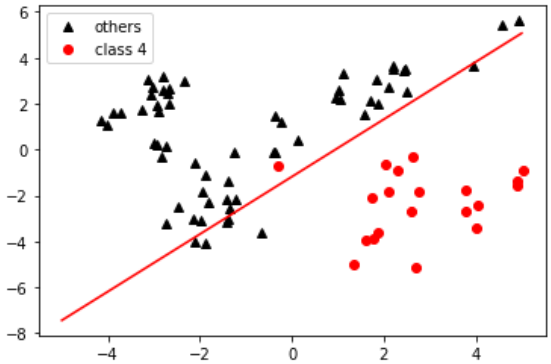
[class 1 VS other classes and W]



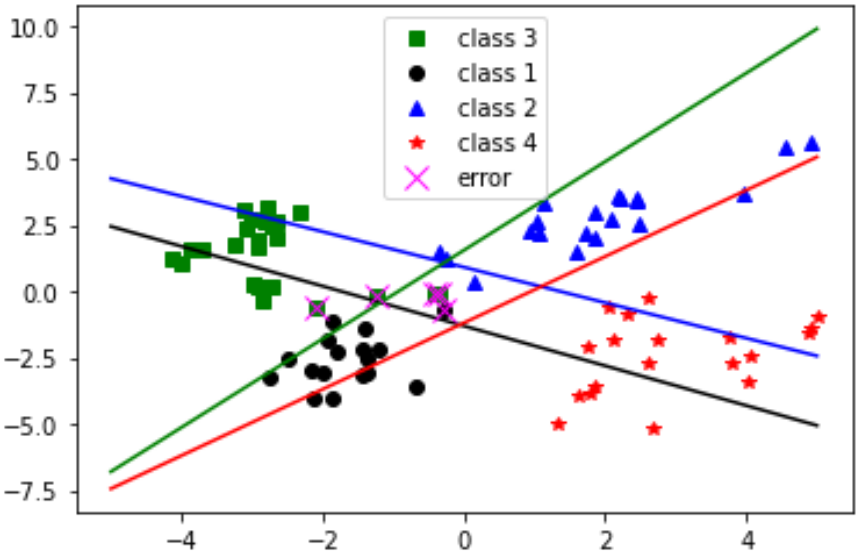
[class 2 VS other classes and W]



[class 3 VS other classes and W]



[class 4 VS other classes and W]



[Result of First OVA classified model]

Each 4 lines which is decision boundaries of the four one-versus-all binary classification, and since we are using g(x) value ([0,1]) to decide class of data in this OVA-based multiclass classification, it is hard to decide each data's class directly through this visualization method, but they are used for multi class classification.

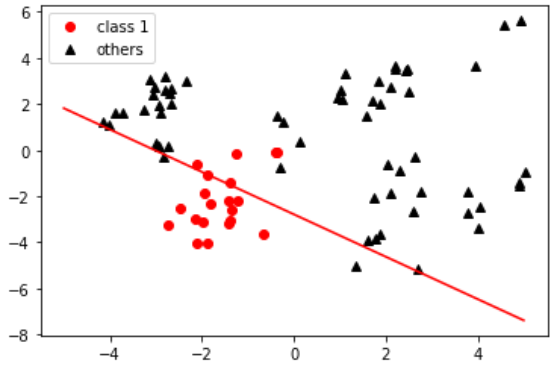
```
Classified label :  [3, 3, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1,
3, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4]
Number of error :  5
Error rate :  0.0625
```

➔ Since the number of data is 80, error rate is 5/80 = 0.0625

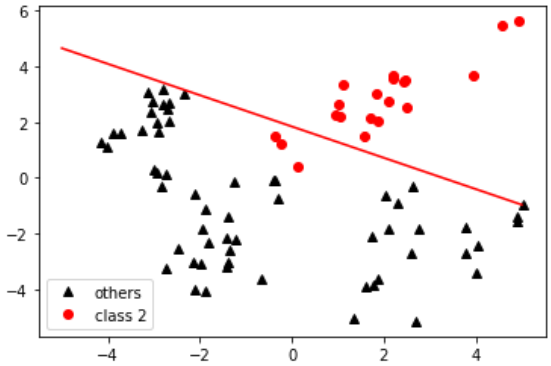
2) Second OVA model with logistic regression (learning rate : 0.3, iteration number : 1000)

```
---class 1---
learning rate : 0.3, Iteration : 1000
>>W_logisticRegression :  [-3.2518306  -1.07539567 -1.16891627]

---class 2---
learning rate : 0.3, Iteration : 1000
>>W_logisticRegression :  [-3.87846543  1.18206691  2.10553032]

---class 3---
learning rate : 0.3, Iteration : 1000
>>W_logisticRegression :  [-5.66594184 -2.72503404  1.71015994]

---class 4---
learning rate : 0.3, Iteration : 1000
>>W_logisticRegression :  [-4.16423785  2.16364695 -1.54497577]
```

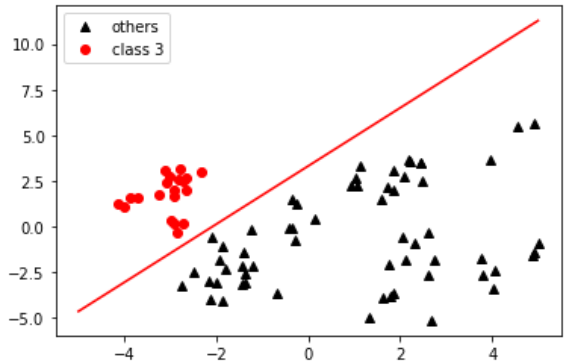
[Result of Second model and Weight from logistic regression]



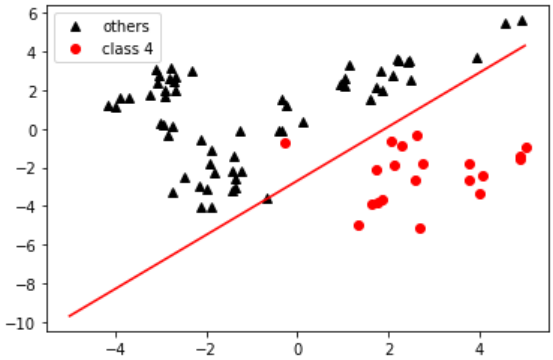
[class 1 VS other classes and W]



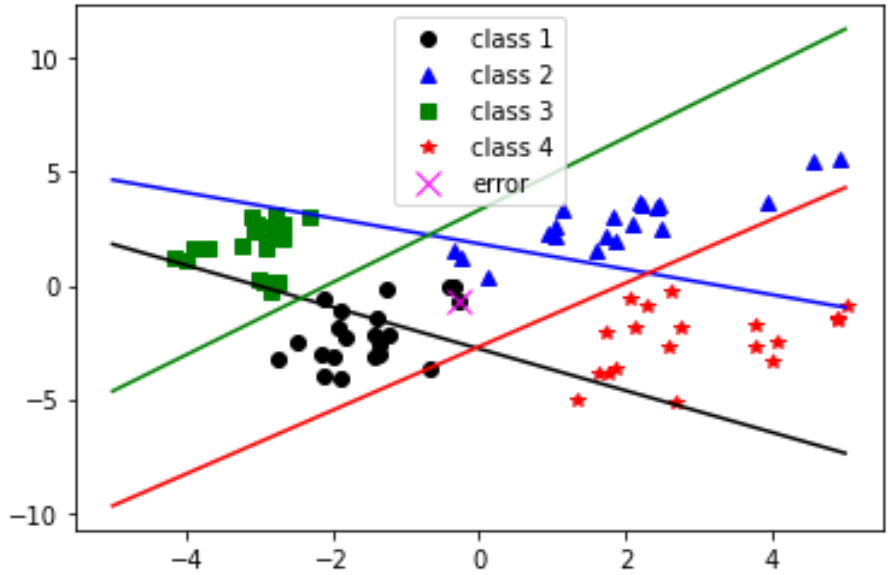
[class 2 VS other classes and W]



[class 3 VS other classes and W]



[class 4 VS other classes and W]



[Result of Second classified OVA model]

```
Classified label :[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4]
```

```
Number of error :  1
Error rate :  0.0125
```

➔ Since the number of data is 80, so the error rate is 1/80 = 0.0125

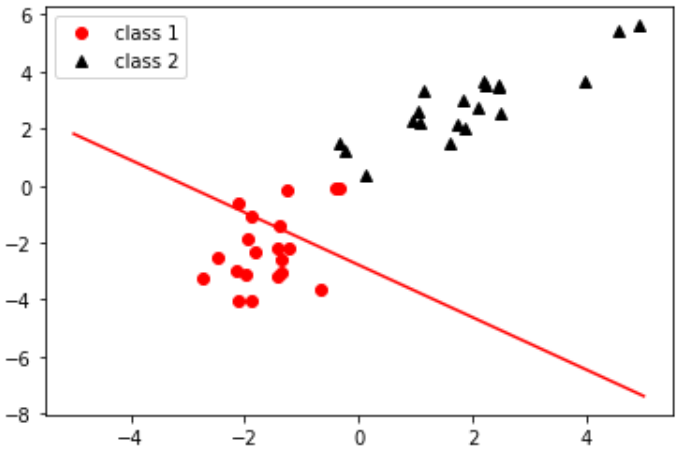
3. One_Versus_One (OVO) using logistic regression

Similar to OVA, I tried two models with learning rate = 0.3, and iteration number = {10,1000}.

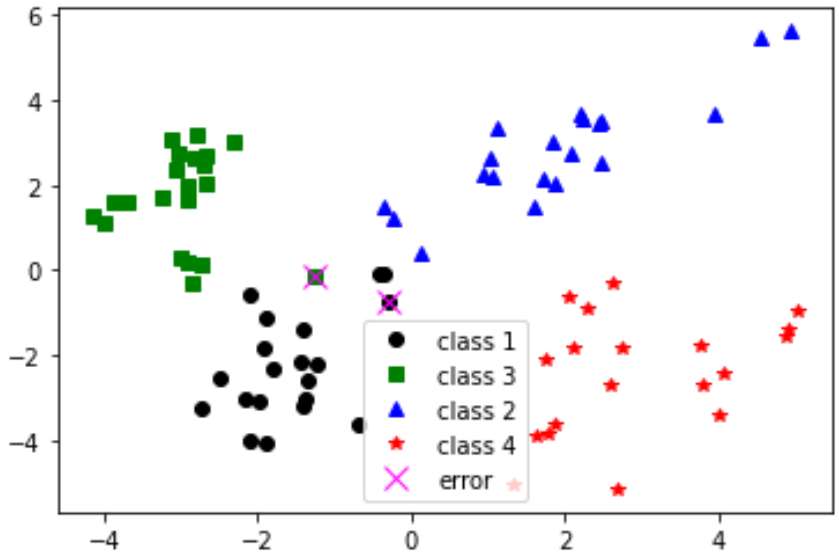
1) First OVO model with logistic regression (learning rate : 0.3, iteration number : 10)

```
---class 1---
learning rate : 0.3, Iteration : 10
>>W_logisticRegression for class (1, 2 ) :  [ 0.03516043 -0.62313168 -0.88665707]
>>W_logisticRegression for class (1, 3 ) :  [ 0.11751909  0.28980418 -1.0623112 ]
>>W_logisticRegression for class (1, 4 ) :  [ 0.08139701 -1.14546976 -0.05465102]

---class 2---
learning rate : 0.3, Iteration : 10
>>W_logisticRegression for class (2, 1 ) :  [-0.03516043  0.62313168  0.88665707]
>>W_logisticRegression for class (2, 3 ) :  [0.07505208 1.10150708 0.24447151]
>>W_logisticRegression for class (2, 4 ) :  [ 0.026313   -0.25127259  1.11257528]

---class 3---
learning rate : 0.3, Iteration : 10
>>W_logisticRegression for class (3, 1 ) :  [-0.11751909 -0.28980418  1.0623112 ]
>>W_logisticRegression for class (3, 2 ) :  [-0.07505208 -1.10150708 -0.24447151]
>>W_logisticRegression for class (3, 4 ) :  [-0.02247642 -0.87067221  0.58857694]

---class 4---
learning rate : 0.3, Iteration : 10
>>W_logisticRegression for class (4, 1 ) :  [-0.08139701  1.14546976  0.05465102]
>>W_logisticRegression for class (4, 2 ) :  [-0.026313    0.25127259 -1.11257528]
>>W_logisticRegression for class (4, 3 ) :  [ 0.02247642  0.87067221 -0.58857694]
```

[Result of First model and Weight from logistic regression]



[class 1 VS class 2 in OVO]



[Result of First OVO classified model]

```
Classified label :[1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4]
```

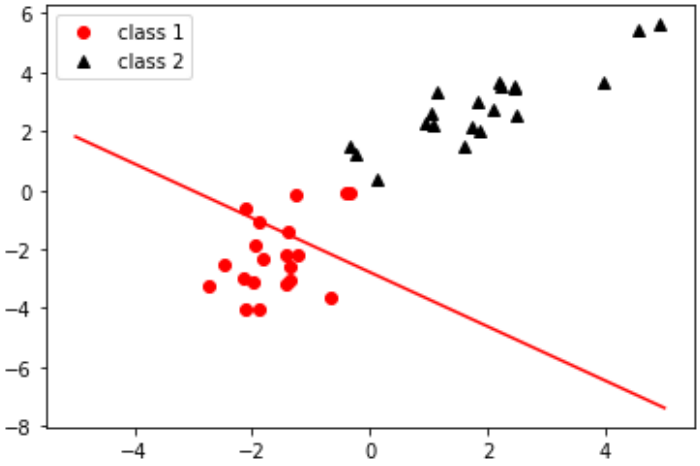
```
Number of error :  2
Error rate :  0.025
```

➔ Since the number of data is 80, so, the error rate is 2/80 = 0.025

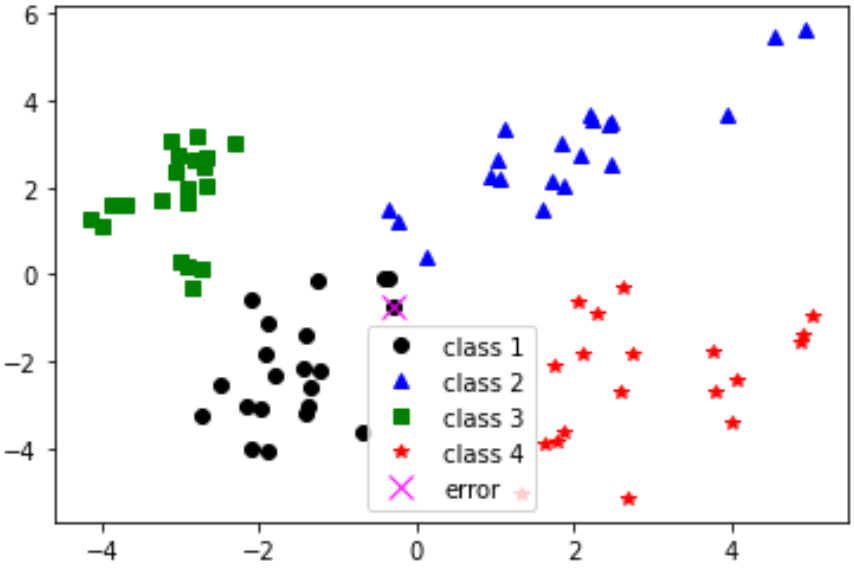
2) Second OVO model with logistic regression (learning rate : 0.3, iteration number : 1000)

```
---class 1---
learning rate : 0.3, Iteration : 1000
>>W_logisticRegression for class (1, 2 ) :  [ 0.51600938 -2.70440182 -3.76866182]
>>W_logisticRegression for class (1, 3 ) :  [ 3.47889996  2.12836091 -4.01459391]
>>W_logisticRegression for class (1, 4 ) :  [-0.34393352 -3.9736074   0.03069733]

---class 2---
learning rate : 0.3, Iteration : 1000
>>W_logisticRegression for class (2, 1 ) :  [-0.51600938  2.70440182  3.76866182]
>>W_logisticRegression for class (2, 3 ) :  [2.19566433 3.90612169 1.15833484]
>>W_logisticRegression for class (2, 4 ) :  [ 0.36236255 -1.09926136  4.22472994]

---class 3---
learning rate : 0.3, Iteration : 1000
>>W_logisticRegression for class (3, 1 ) :  [-3.47889996 -2.12836091  4.01459391]
>>W_logisticRegression for class (3, 2 ) :  [-2.19566433 -3.90612169 -1.15833484]
>>W_logisticRegression for class (3, 4 ) :  [-1.26911763 -2.15985529  2.07715526]

---class 4---
learning rate : 0.3, Iteration : 1000
>>W_logisticRegression for class (4, 1 ) :  [ 0.34393352  3.9736074  -0.03069733]
>>W_logisticRegression for class (4, 2 ) :  [-0.36236255  1.09926136 -4.22472994]
>>W_logisticRegression for class (4, 3 ) :  [ 1.26911763  2.15985529 -2.07715526]
```

[Result of Second model and Weight from logistic regression]



[class 1 VS class 2 in OVO]



[Result of Second classified OVO model]

```
Classified label :[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4]
```

```
Number of error :  1
Error rate :  0.0125
```

➔ Since the number of data is 80, so the error rate is 1/80 = 0.0125

The OVA method and the OVO method have advantages and disadvantages of each other.

OVA is efficient because it can use K logistic regression, but on the other hand, the amount of training data tends can be biased. On the other hand, the OVO method has an advantage in that the size of training is very small because training is performed by comparing each one classes. However, since there must be as many classifiers as kC2 and more regression models need to be processed, it is difficult to see that there is a good benefit in terms of training time.

Both model using OVA and model using OVO had the same error rate as 0.0125 when the iteration reached 1000. However, when the iteration number is 10, it can be seen that the model using OVO has a lower error rate than the model using OVA. In other words, I thought this could create a model with fewer errors in fewer iterations when using OVO on our dataset. Of course, there are other factors that evaluate the performance of the model besides the error rate, but if reducing the error rate is the top priority, in our data, the method using OVO showed a small error rate in fewer iterations.

d) **Discussions and Conclusions. Any design decisions you had to make and your experimental observations. What do you observe about the behavior of your program when you run it? Does it seem to work the way you think it should? Play around a little with different setting to see what happens. Note, your open-ended exploration is highly valued.**

It was interesting to do multi-class classification after learning binary classification. I can learn how the multi- class classification works through each binary classification, and the difference between OVO and OVA strategy was very interesting. I had an opportunity to use TensorFlow before, but at that time, I just memorize that logistic regression can used for multi-class data with more than three classes, and sigmoid is used as the activation function, so I always have a question while putting these values as hyper parameters in the neural network using tensorflow. But because of this homework, I can learn why it is good to use logistic regression in a network and to use the softmax function for multi-class classification is appropriate. It is very good to learn the mathematical principles of things than only put in the neural network as one parameter value.

Even though I used proper hyper parameters at that time, the model keep occurring overfitting. So, the training error was small but test error was big because I obtained the data by myself, and also I preprocess the data by myself. Each time I used several regulatory methods while I did not fully understand the mathematical principles, so it would be good if I could cover regulation further in this upcoming cse353 class.