Aslı Aykan
2016400222

CMPE 462 | Assignment 2

PART 1

The first part of the assignment is about logistic regression and has two steps that uses different gradient descent methods. For each step, cross-validation should be applied for different step sizes (learning rates) with a threshold value.

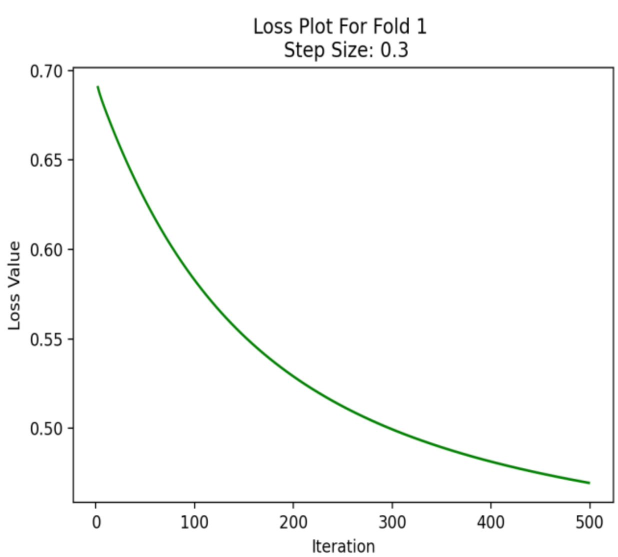
Code Summary:

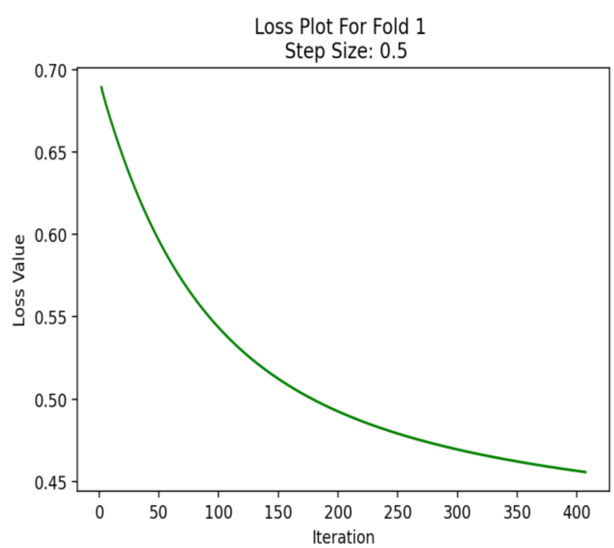
- I created *calculateGradient()* function for batch gradient descent and *calculateMiniGradient()* for stochastic gradient descent, which calculate gradient values for given input data. In calculateMiniGradient() function, it enters a loop for an epoch by selecting all indexes randomly and updates the feature coefficients for each points(**w**). In calculateGradient(), however, it directly calculates the gradient value.
- I created *calculateSigmoid()* function to calculate sigmoid function value, which is used while calculating batch gradient value.
- I created *calculateLoss()* function to calculate the loss according to the formula we learned in class.
- I created *foldData()* function to return a fold with train and test values. For 5-fold case(the case in the assignment), a for loop calls this function 5 times to get the fold values.
- For step sizes, I chose **small value as 0.3**, **medium value as 0.5**, and **large value as 0.9**

STEP 1

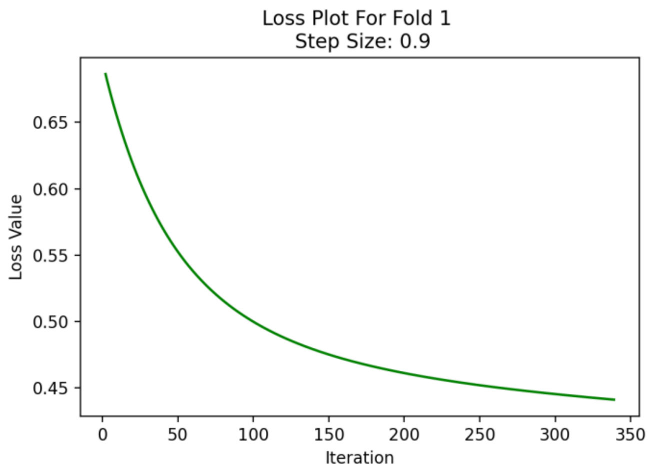
Step 1 is about batch gradient descent. I used different threshold values to compare the results.

<u>Case 1:</u> Threshold = 0.0001





Aslı Aykan
2016400222



As can be seen from these 3 figures, as step size increases, the number of iterations decreases. For 5 fold, I took the averages of loss values of test data and the number of iterations needed for each fold.

Average Values **for step size 0.3** are:

Average Loss: 0.48687783241223465

Average Number of Iterations: 478.8

Average Values **for step size 0.5** are:

Average Loss: 0.4740777700040141

Average Number of Iterations: 390.8

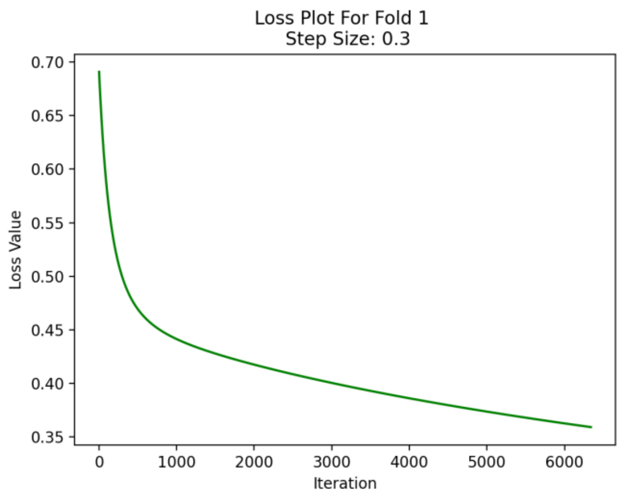
Average Values **for step size 0.9** are:

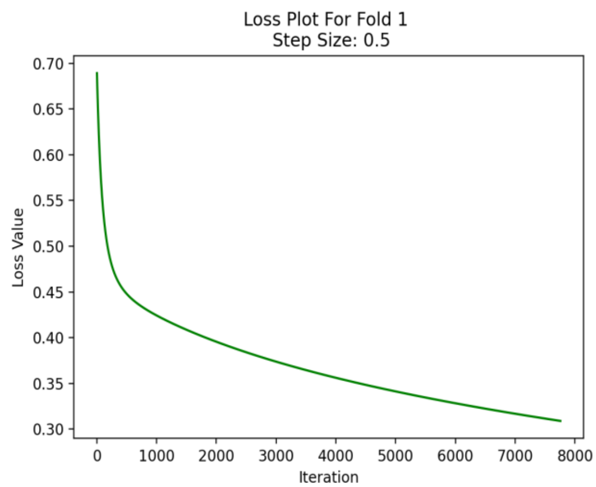
Average Loss: 0.46022949979465777

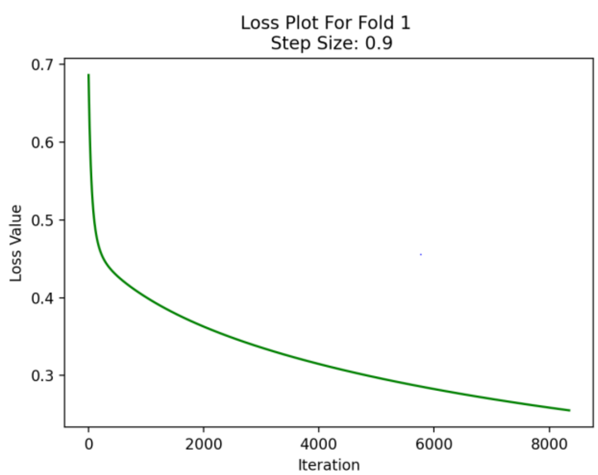
Average Number of Iterations: 328.4

Aslı Aykan
2016400222

<u>Case 2:</u> Threshold = 0.00001







As can be seen from these 3 figures, as step size increases, the number of iterations also increases. (Opposite of case 1). I think the reason of this difference between two cases is that if the threshold is too small, it gets harder to converge for larger step sizes.

For 5 fold, I took the averages of loss values of test data and the number of iterations needed for each fold.

Average Values **for step size 0.3** are:

<u>Average Loss:</u> 0.3737183371597029

<u>Average Number of Iterations</u>: 7008.4

Average Values **for step size 0.5** are:

Average Loss: 0.3203508802151037

Average Number of Iterations: 8498.6

Average Values **for step size 0.9** are:

Average Loss: 0.26819004843062483

Average Number of Iterations: 8633.0

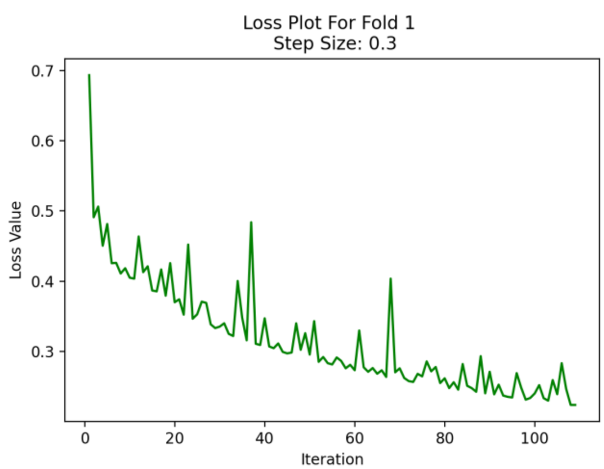
As can be seen from these two cases, we can say that the decrease in the threshold makes the average loss decrease for these two values of threshold.

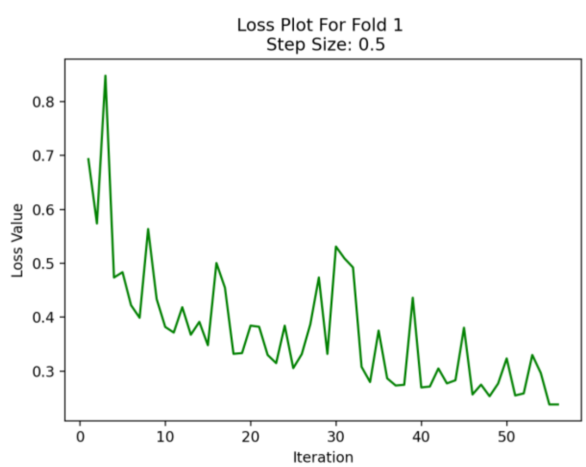
Additionally, 0.9 is the step size that gives the lowest loss values among others.

STEP 2

Step 2 is about stochastic gradient descent. I used different threshold values to compare the results.

Case 1: Threshold = 0.0001





Aslı Aykan
2016400222

For 5 fold, I took the averages of loss values of test data and the number of iterations needed for each fold.

Average Values **for step size 0.3** are:

<u>Average Loss:</u> 0.20386797979028257

<u>Average Number of Iterations</u>: 211.8

Average Values **for step size 0.5** are:

<u>Average Loss:</u> 0.19861224258845947

<u>Average Number of Iterations:</u> 176.8

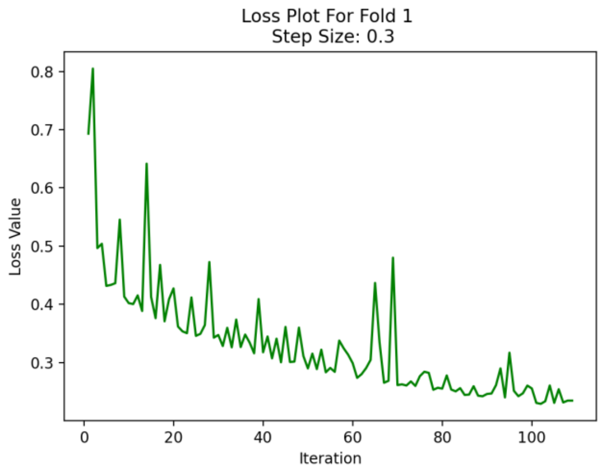
Average Values **for step size 0.9** are:

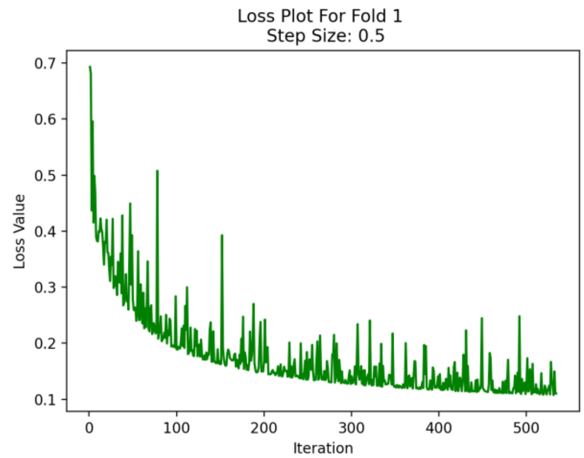
<u>Average Loss:</u> 0.13564935778028073

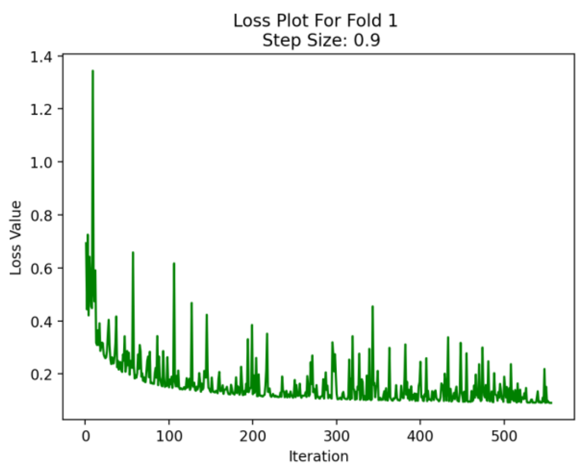
<u>Average Number of Iterations:</u> 364.4

Aslı Aykan
2016400222

<u>Case 2:</u> Threshold = 0.00001







For 5 fold, I took the averages of loss values of test data and the number of iterations needed for each fold.

Average Values **for step size 0.3** are:

<u>Average Loss:</u> 0.1749344780620009

<u>Average Number of Iterations:</u> 545.6

Aslı Aykan
2016400222

Average Values **for step size 0.5** are:

Average Loss: 0.13436855722315538

Average Number of Iterations: 739.2

Average Values **for step size 0.9** are:

Average Loss: 0.10679080442318223

Average Number of Iterations: 992.2

As can be seen from these two cases, we can say that the decrease in the threshold makes the average loss decrease for these two values of threshold.

Additionally, 0.9 is the step size that gives the lowest loss values among others.

Since the stochastic gradient algorithm works by choosing random sample points, we cannot easily generalize the iteration numbers among different step sizes.

It is also worth to note that since the iteration is an epoch for stochastic gradient descent, it has less iterations compared to the batch gradient descent. According to the values I gave for threshold and step sizes, stochastic has lower loss values compared to the batch gradient descent.

Note: The code runs for step size 0.9 (the best among other step sizes) and threshold 0.0001. (Although 0.00001 gives smaller loss, I wrote that value of threshold to the code because of the performance issues. For smaller threshold, the time for the code to run is longer)

The code saves the plot for first fold with step size 0.9. It also prints the average loss and iteration number to the console.

Image names are:

- part1_step1.png
- part1_step2.png

PART 2

For Naïve Bayes classification, we assume that the features are conditionally independent. The formulas we learned in the class is:

Aslı Aykan
2016400222

$P(Y=y_k \mid X) = P(X|Y=y_k)P(Y=y_k)/P(X)$ and $P(X|y) = \prod_j p_j(x^j|Y)$

and $y^* = \text{argmax}\ y_k \prod_j p_j(x^j|Y=y_k)P(Y=y_k)$

So to apply these formulas to this question, we need to find the conditional probabilities first.

P(giveBirth|Mammal) = 6/7

P(canFly|Mammal) = 1/7

P(liveInWater|Mammal) = 2/7

P(haveLegs|Mammal) = 5/7

P(giveBirth|Non-Mammal) = 1/3

P(canFly|Non-Mammal) = 3/13

P(liveInWater|Non-Mammal) = 3/13

P(haveLegs|Non-Mammal) = 9/13

P(Mammal) = 7/20

P(Non-Mammal) = 13/20

Test values = (yes, no, yes, no) = (1,0,1,0)

So, $P(y=\text{Mammal} \mid 1,0,1,0) \propto (6/7)(6/7)(2/7)(2/7)(7/20) = 0.02099$

$P(y=\text{Non-Mammal} \mid 1,0,1,0) \propto (1/13)(10/13)(3/13)(4/13)(13/20) = 2{,}73 \times 10^{-3} = 0.00273$

Since **0.02099 > 0.00273**, we can say that the test value is classified as **Mammal.**