Nikhil Rai
Resulta and inference about the logistic regression of binary and k-class classification.

- **Binary Class Classification**

➢ **Threshold:=0.5**

Learning Rate	Tolerance Value	Time	Accuracy (%)
0.1	20	0m15.971s	99.12
0.1	15	1m8.596	99.11
0.1	10	2m18.976s	99.05
0.01	5	0m12.607	99.04
0.01	3	0m10.604s	99.07

➢ **Interpretations:**

When we keep the stepsize as 0.1 and goes on decreasing the tolerance value, we found that accuracy goes on decreasing. This could be due to the generalisation error. Our model may be overfitting the test data.

The highest accuracy that we got is 99.12 when learning Rate= 0.1 and tolerance value= 20.

- **k-class Label Classification**

Learning Rate	Tolerance Value	Time	Accuracy
0.01	100	0m11.724sec	78.46
0.01	50	0m14.088s	80.68
0.01	30	0m36.112s	83.45
0.01	20	5m25.267s	85.24
0.01	10	31m18.239s	88.66

Interpretations:

Here, we can see that as the tolerance value goes on increasing our accuracy goes on increasing. This is because we are getting the optimum theata for which our error is least. So when the test data are tested using this optimum theata, we found the highest accuracy.

Nikhil Rai
Resulta and inference about the logistic regression of binary and k-class classification.

- **Binary label classification**

  - **Threshold:=0.9**

Learning Rate	Tolerance Value	Time	Accuracy (%)
0.1	20	0m16.216s	99.12
0.1	15	1m9.032s	99.11
0.1	10	2m22.517s	99.05

  - **Threshold:= 0**

Learning Rate	Tolerance Value	Time	Accuracy (%)
0.1	10	2m21.216s	98.9

  - **Threhsold:= -0.1**

Learning Rate	Tolerance Value	Time	Accuracy (%)
0.1	20	0m15.958s	9.8
0.1	15	1m11.772s	9.8
0.1	10	2m23.517s	9.8

Important observation:

By this experiment what we wanted to have some knowledge about the test data. We found that even though the threshold was kept very high, we got almost the same accuracy when threshold was 0.5. Same happened when threshold was kept equla to 0. And when we kept threshold as negative, accuracy got reduced to 9.8. Observe the following code:

```
threshold = some value.
prediction = hypothesisF ( optimumTheata, X_test )
for i in range ( y_testLabel.size ):
    if ( prediction [i] > threshold ):
        prediction [i] = 1
    else:
        prediction [i] = 0
```

Note that *'prediction'* contains values between 0 and 1. When threshold was kept negative, according to above code all test data are predicted as 1 since *'prediction'* contains all positive values. So when accuracy was calculated, we got accuracy as 9.8 which means that most of the instances' label in test data is zero. Hence we can infer that almost 90 percent of the test instances are zero.