Homework 7
CSCI4100

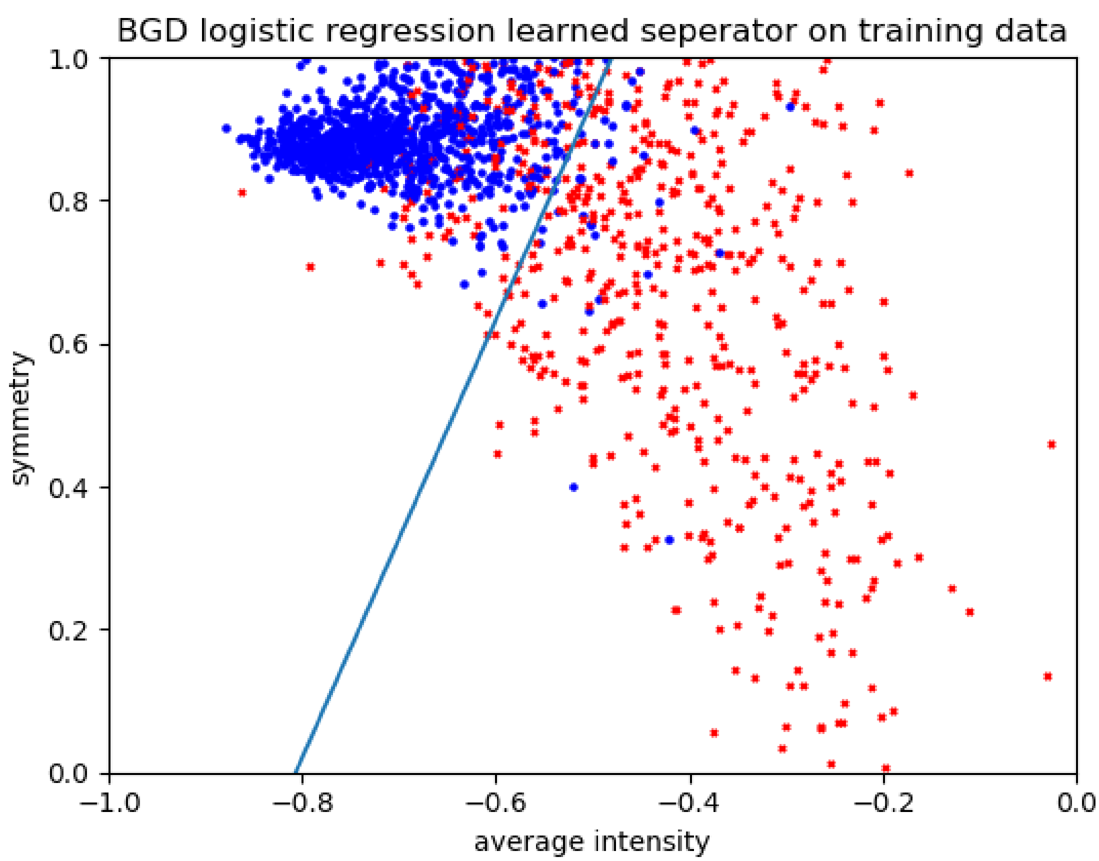
Han Hai
Rin:661534083
haih2@rpi.edu

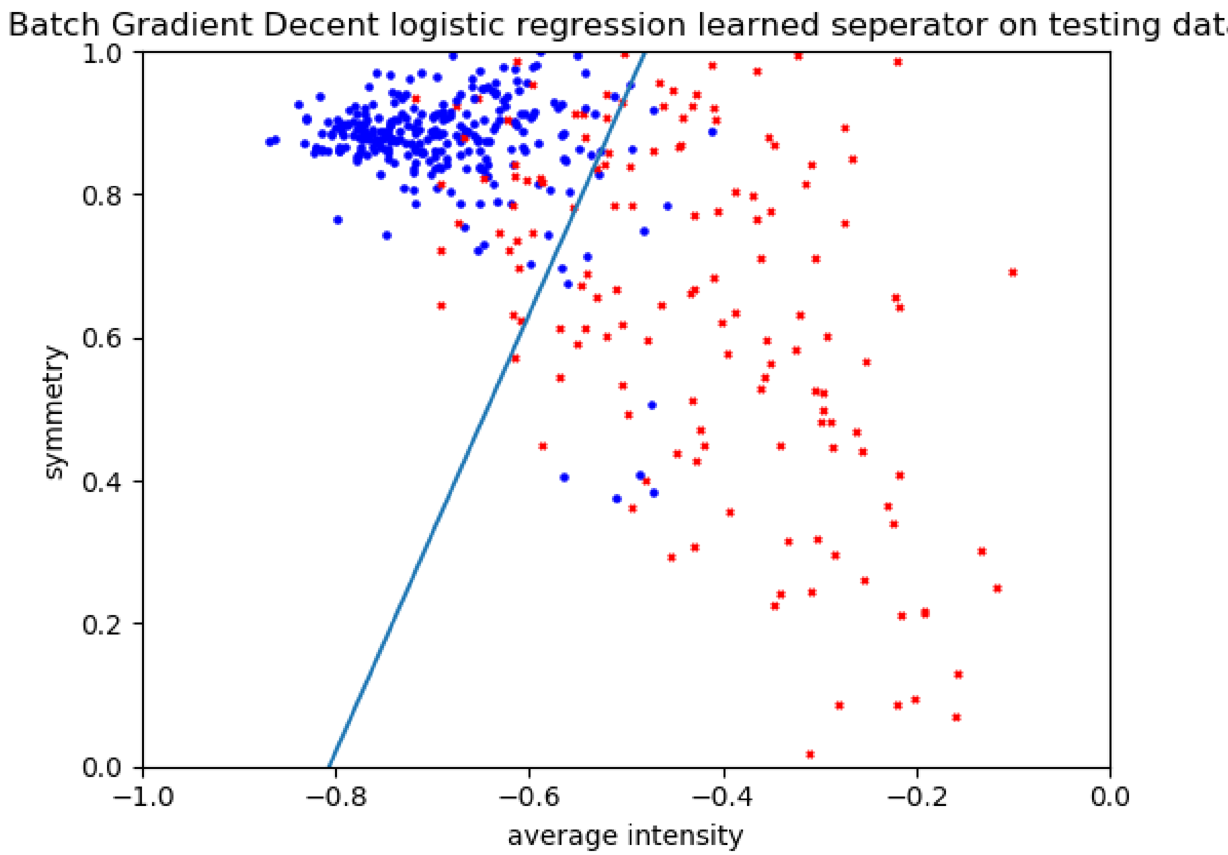
October 22th 2018

1. **1.(500)** Classifying Handwritten Digits 1 vs. 5
   Use your chosen algorithm to find the best seperator you can using the training data. The output is +1 if the example is a 1 and -1 for a 5.

   (iii.) I picked Logistic regression for classification using batch gradient decent

   (a) Give seperate plots of the training and test data, together with the seperators.
   Here are the two plots, note that blue dots represents 1s and red ones represent 5s



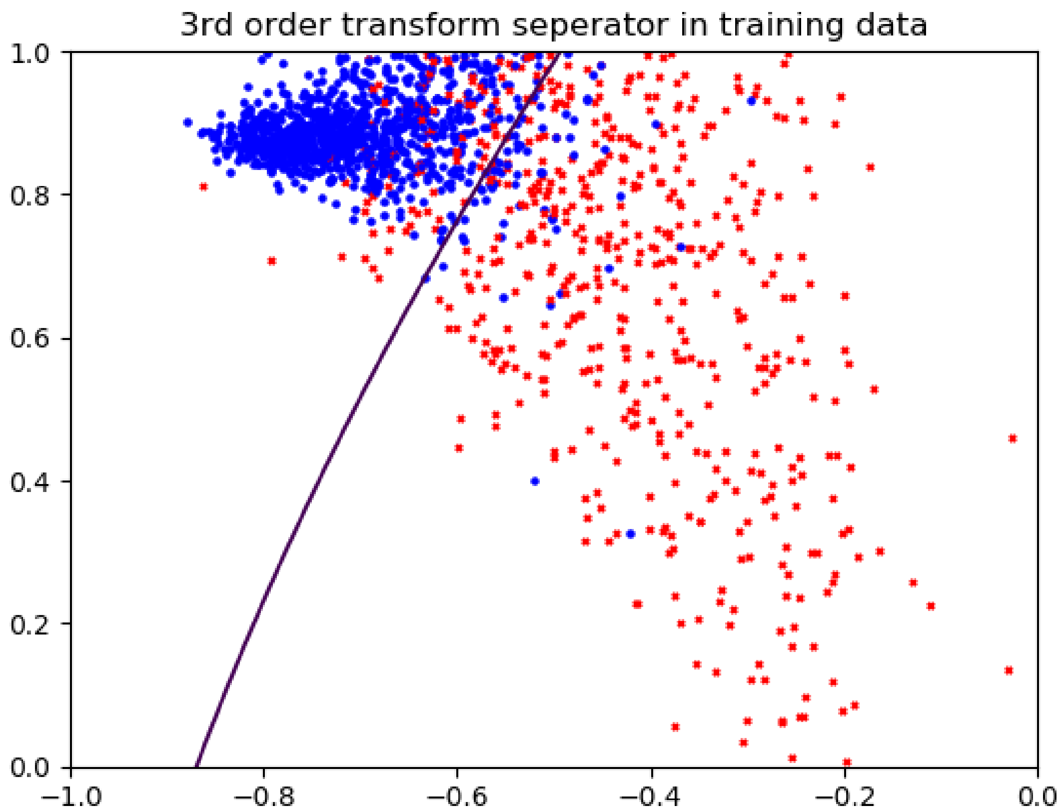


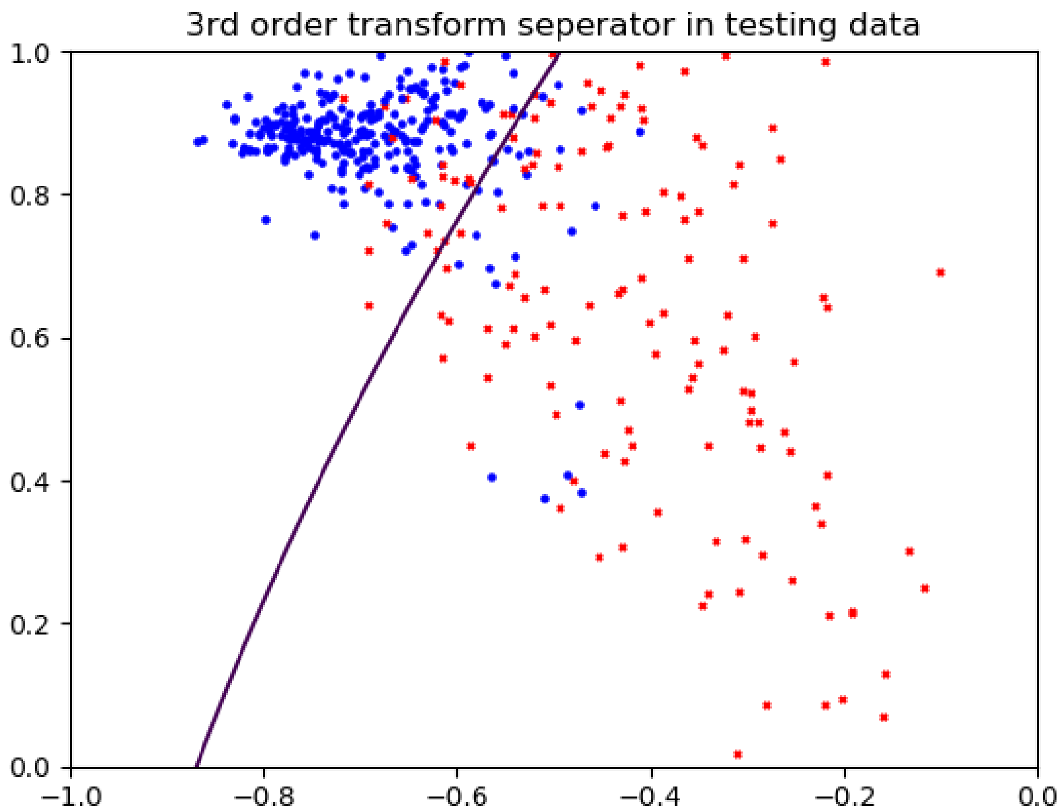
(b) Compute Ein on your training data and Etest, the test error on the test data
I calculated Ein by dividing the incorrectly classified points by all the points , and $E_{in} = 0.0929$.
Using the same method on testing set, I have $E_{test} = 0.113$

(c) Obtain a bound on the true out-of-sample error. You should get two bounds, one based on Ein and on based on etest. Use a tolerance $\delta = 0.05$ Which is the better bound?
Recall $E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} ln \frac{4mH(2N)}{\delta}}$
$=E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} ln \frac{4((2N)^{d_{vc}}+1)}{\delta}}$
for linear classifier in 2d $d_{vc} = 3$
for $E_{in} = 0.093$ and its N=1561
we have$E_{out} = 0.093 + 0.146 = 0.239$

for $E_{test} = 0.113$ and its N=424
we have$E_{out} = 0.113 + 0.464 = 0.577$
clearly Eout calculated by Ein in training data is a better bound

(d) Now repeat using a 3rd order polynomial transform.
First, 3rd order transform
$(x_0, x_1, x_2) \rightarrow (x_0, x_1, x_2, x_1^2, x_2^2, x_1x_2, x_1^3, x_2^3, x_1^2x_2, x_1x_2^2)$
the learned weight is here:
$[-6.18399373, -1.53960842, -1.29814265, 3.42171311, 0.26660066 -3.03213652, -3.44241927, 1.10338077, 3.8298719$
The two plots are below, note that it's a curve instead of a line


3rd order transform seperator in training data
1.0
0.8
0.6
0.4
0.2
0.0
−1.0
−0.8
−0.6
−0.4
−0.2
0.0



and the $E_{in} = 0.0993$
$E_{test} = 0.118$
Similar to linear, we can compute E_{out} bound using vc bound. Here, $d_{vc} = 10$ the order of polynomial
Eout bound from Ein:
$E_{out} \leq 0.099 + 0.659 = 0.758$
Eout bound from etest:
$E_{out} \leq 0.118 + 1.16 = 1.28$

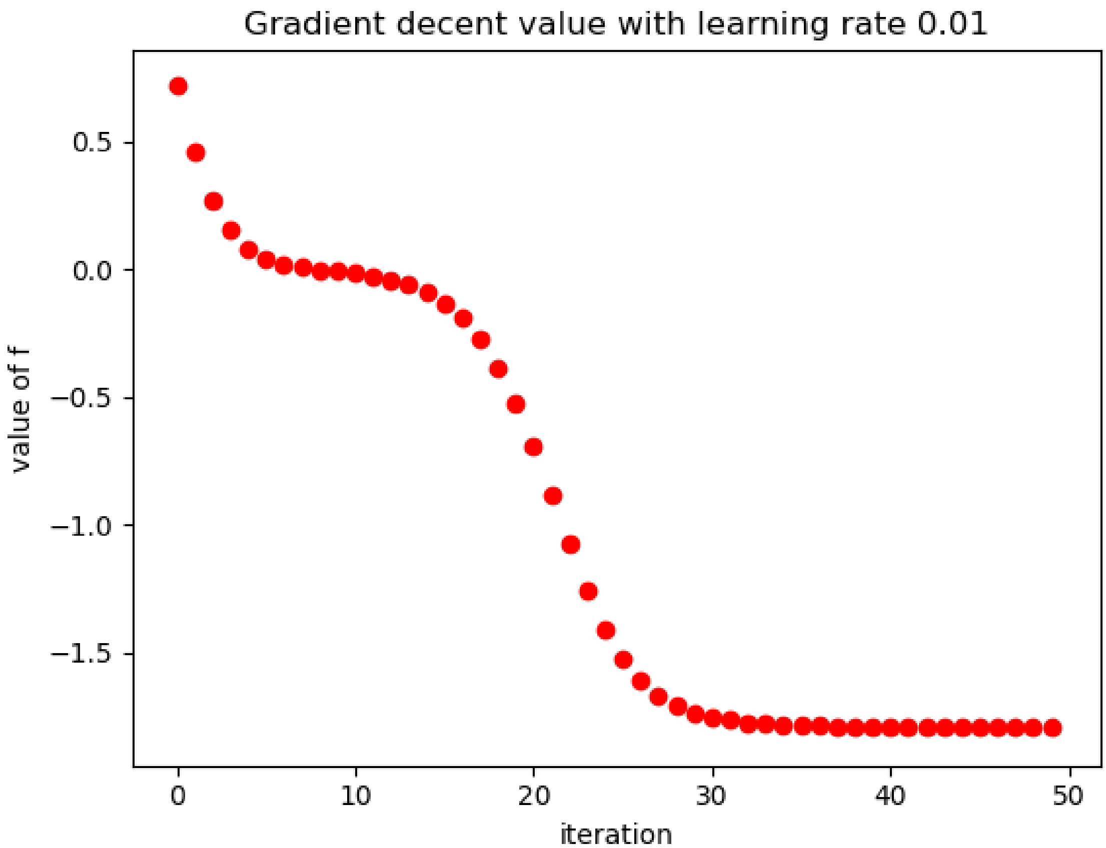
(e) As your final deliverable to a customer, would you use the linear model with or without the 3rd order polynomial transform? Explain.

I would definitely choose a linear without 3rd order polynomial transform
Reasoning:
As we can see from part d, the error bar of the 3rd order polynomial is way higher due to the model complexity. The high error bar results in an unmanageable high Eout bound, and clearly, we do not want that

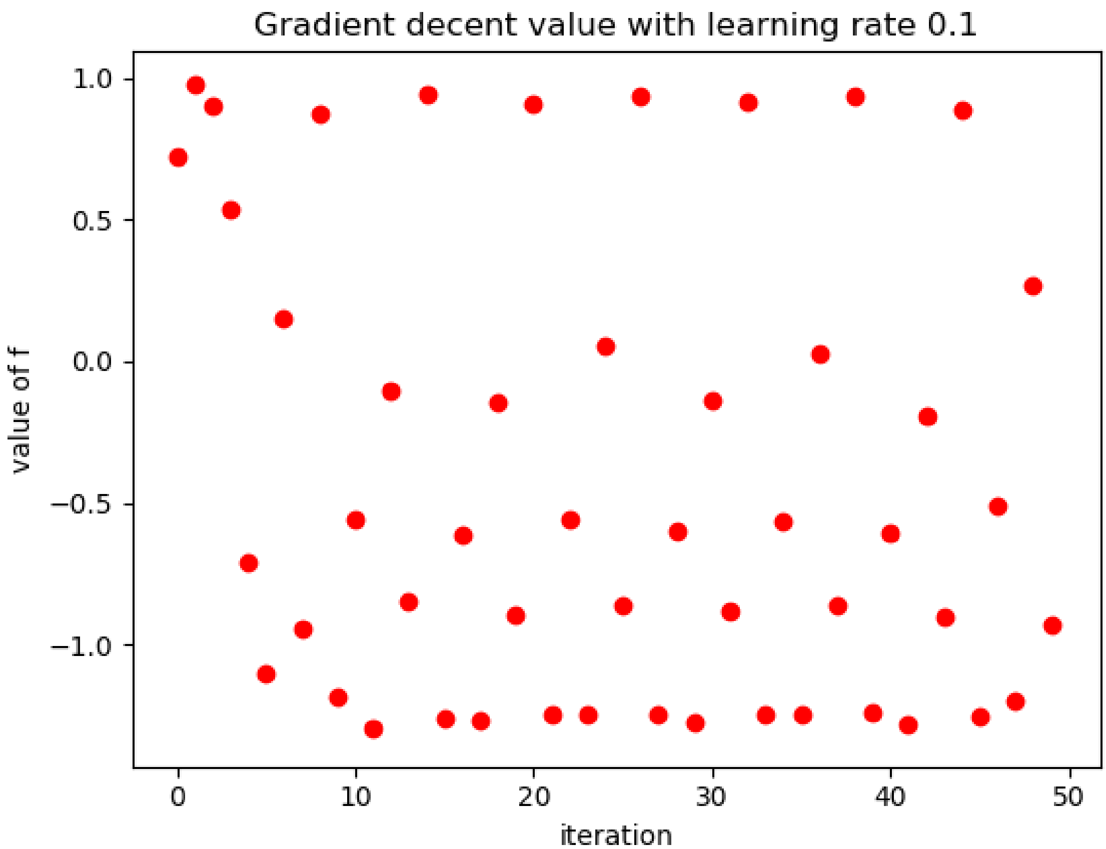
2. **2. (200)** Gradient Descent on a "Simple" Function
$f(x, y) = x^2 + 2y^2 + 2sin(2\pi x)sin(2\pi y)$.

(a) Implement gradient descent to minimize this function with $x_0 = 0.1; y_0 = 0.1$ let learning rate be 0.01 and 50 iterations. Give a plot of value vs iteration? What happened if we change the learning rate to 0.1

Solution:
$dfx = 2x + 4cos(2\pi x)sin(2\pi y)$
$dfy = 4y + 4sin(2\pi x)cos(2\pi y)$
I used fixed rate gradient descent described in pg95
Here is the plot for learning rate 0.01



Here is the plot for learning rate 0.1



As we can see with a learning rate of 0.1, the value of f bounce around and cannot converge, it fails to learn because the learning rate is too fast.

(b) Obtain the minimum value and the location with (0.1,0.1), (1,1),(-0.5,-0.5),(-1,-1). A table with the location of the minimum and the minimum value.

x0	y0	neta	xfinal	yfinal	minimum
0.1	0.1	0.01	0.23	-0.22	-1.793
1	1	0.01	0.69	0.21	-1.26
-0.5	-0.5	0.01	-0.69	-0.21	-1.25
-1	-1	0.01	-0.69	-0.21	-1.26

As we can learn from the table, finding a global "true" minimum is hard because the gradient descent converges to the local minimum that is closest to the initial points.

3. [**300 points**] **Problem 3.16** In Example 3.4, it is mentioned that the output of the final hypothesis g(x) learned using logistic regression can be thresholded to get a 'hard' classification. This problem show how to use the risk matrix introduced in Example 1.1 to obtain such a threshold.
See Example 1.1

(a) Define the cost(accept) as you expected cost if you accept the person. Similarly define cost(reject). Show that
$cost(accept) = (1 - g(x))c_a$

$cost(reject) = g(x)c_r$
The cost of accept should be $P[accept|correct] * cost(correctandaccept) + P[accept|incorrect] * cost(incorrectandaccept)$
since cost(correct and accept)=0
$=P[accept|incorrect] * c_a$
$=(1 - P[y = +1|x]) * c_a$
$=(1 - g(x))c_a$
the cost of reject should be
$P[reject|correct] * cost(rejectandcorrect) + P[reject|incorrect] * cost(rejectandincorrect)$
similarly,
$=g(x) * c_r + 0$
$=g(x)c_r$

(b) Use part a to derive k
let's derive k by setting the cost of accept less than the cost of reject
$(1 - g(x))c_a \leq g(x)c_r$
$c_a - g(x)c_a \leq (x)c_r$
$c_a \leq g(x)c_r + g(x)c_a$
$c_a \leq g(x)(c_r + c_a)$
$g(x) \leq \frac{c_a}{c_a+c_r}$

therefore $k = \frac{c_a}{c_a+c_r}$

(c) Use the cost-matrices for the supermarket and CIA applicationsin Example 1.1 to compute the threshhold k for each othese two cases. Give some intuition for the threshholds you get.

for the case of supermarket,
$c_a = 1, c_r = 10$
and $k = \frac{1}{11} = 0.0909$
for the case of CIA,
$c_r = 1, c_a = 1000$
and $k = \frac{1000}{1001} = 0.999$
The result makes sense intuitively. We want to set the threshold of acceptance high in CIA, because the consequences of letting an intruder in is very serious. In contrary, it's OK if the true agent is locked outside for a while. While in the supermarket, we want to set the threshold of acceptance low. Because it's less harmful to let an intruder in, and the benefit for such threshold is that the average employee will not frequently locked outside.