Kelly Cheng
k.cheng@gatech.edu

Collaborators: Michael Simpson, Adelene Sim, Christopher Wedge, Cheryl Miller, Marvin Galang

1.1 Batch Gradient Descent

a)

$$-\sum_{i=1}^{N}[(1-y_i)\log(1-\sigma(w^T x)+y_i\log\sigma(w^T x))]$$

$$\frac{\partial}{\partial\sigma}-\sum_{i=1}^{N}[(1-y_i)\log(1-\sigma(w^T x)+y_i\log\sigma(w^T x))]$$

Given $\frac{\partial}{\partial\sigma}=\sigma(x)(1-\sigma(x))$ for sigmoid function

$$=-\sum_{i=1}^{N}(1-y_i)\frac{1}{1-\sigma(w^T x)}\left(\frac{\partial}{\partial\sigma}1-\sigma(w^T x)\right)+y_i\frac{1}{\sigma(w^T x)}\left(\frac{\partial}{\partial\sigma}\sigma(w^T x)\right)$$

$$=-\sum_{i=1}^{N}(1-y_i)\frac{x}{1-\sigma(w^T x)}(-(\sigma(w^T x)(1-\sigma(w^T x))))+y_i\frac{x}{\sigma(w^T x)}((\sigma(w^T x)(1-\sigma(w^T x))))$$

$$=-\sum_{i=1}^{N}[-\sigma(w^T x)x+\sigma(w^T x)xy_i+y_i x-\sigma(w^T x)xy_i]$$

$$=-\sum_{i=1}^{N}[-\sigma(w^T x)x+y_i x]$$

$$=\sum_{i=1}^{N}[\sigma(w^T x)-y_i]x$$

1.2 Stochastic Gradient Descent

a) $l=(1-y_t)\log(1-\sigma(w^T x_t))+y_t \log\sigma(w^T x_t)$
b) $w_t=w_{(t-1)}-\eta(\sigma(w^T x_t)-y_t)x_j$
c) linear $(O(n))$
d) ηcontrols how big of a step is taken when updating. Small η can cause gradient descent to be slow while large η can cause gradient descent to overshoot the minimum and fail to converge or diverge.
e) $w_{(t+1)}=w_t-\eta(\sigma(w^T x_t)-y_t)x_j-\mu\eta\|w\|^2$
f) $W_{(t+1)}=W_t-\eta_t y_t l'(y_t s_t W_t x_t)x_t / s_{(t+1)}$ where $w_t=s_t W_t$ and $s_{(t+1)}=(1-\eta_t\mu)s_t$
Complexity is linear O(n)

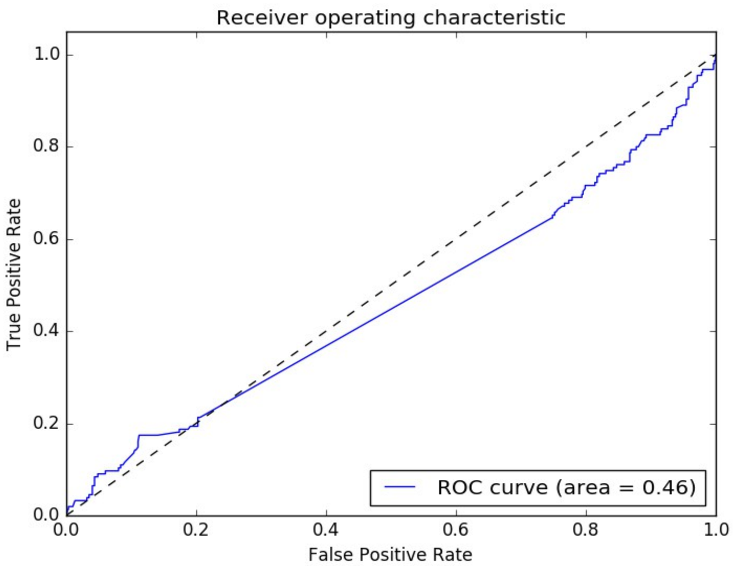
2.1

Metric	Deceased patients	Alive patients
Event Count		
1. Average Event Count	1029.059	682.42022
2. Max Event Count	16829	12627
3. Min Event Count	2	1
Encounter Count		
1. Average Encounter Count	24.861	18.66322
2. Max Encounter Count	375	391
3. Min Encounter Count	1	0
Record Length		
1. Average Record Length	151.397	194.65409
2. Max Record Length	2601	3103
3. Min Record Length	0	0
Common Diagnosis		
1. DIAG320128	1019	
2. DIAG319835	721	
3. DIAG317576	719	
4. DIAG42872402	674	
5. DIAG313217	641	
Common Laboratory Test		
1. LAB3009542	66910	
2. LAB3000963	57733	
3. LAB3023103	56967	
4. LAB3018572	54667	
5. LAB3007461	53548	
Common Medication		
1. DRUG19095164	12452	
2. DRUG43012825	10388	
3. DRUG19049105	9329	
4. DRUG19122121	7586	
5. DRUG956874	7294	

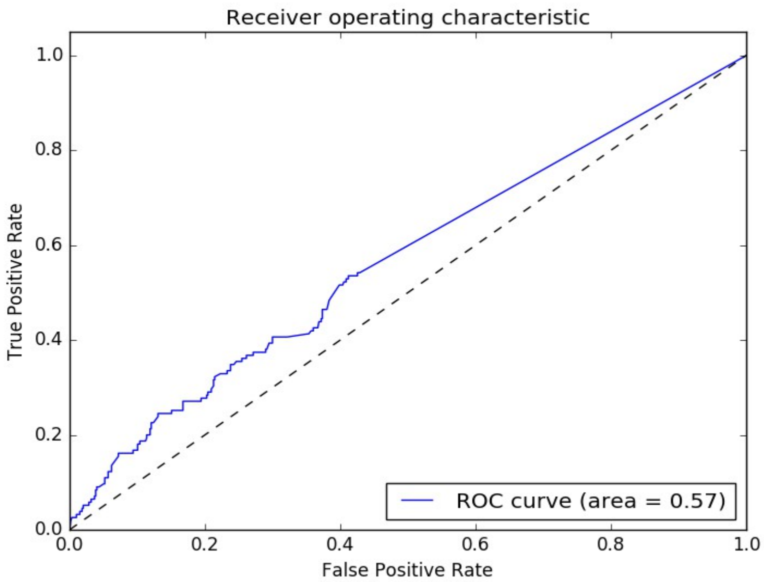
2.3

a) It took me a long time to get this piece. At first, I was getting “upside down” ROC curves (flipped across the diagonal with ROC ~.45). I forgot to take a record of this before I updated my code. I then updated and got this result with a very similar ROC.

This looks even more odd than the original, having no real pattern. The next attempt finally had the curve above the diagonal as expected (ROC > .5), but still looked extremely odd, with the perfectly straight line starting about halfway.



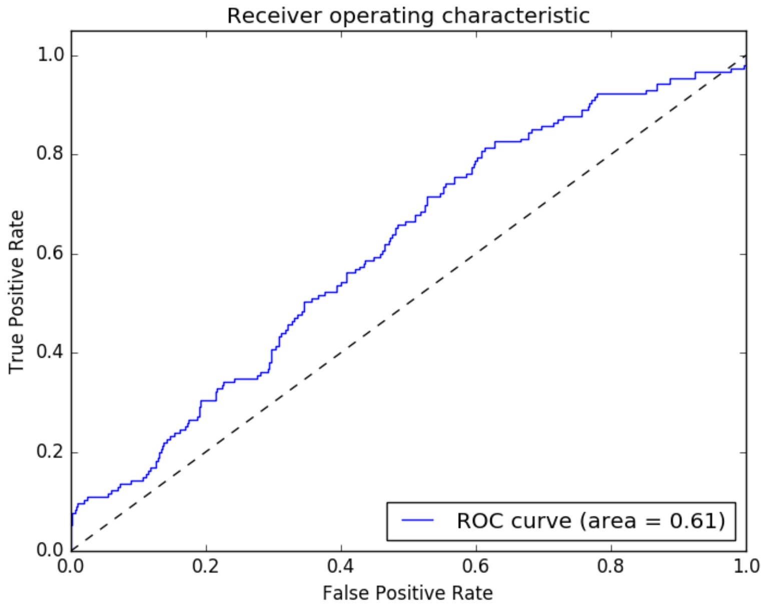
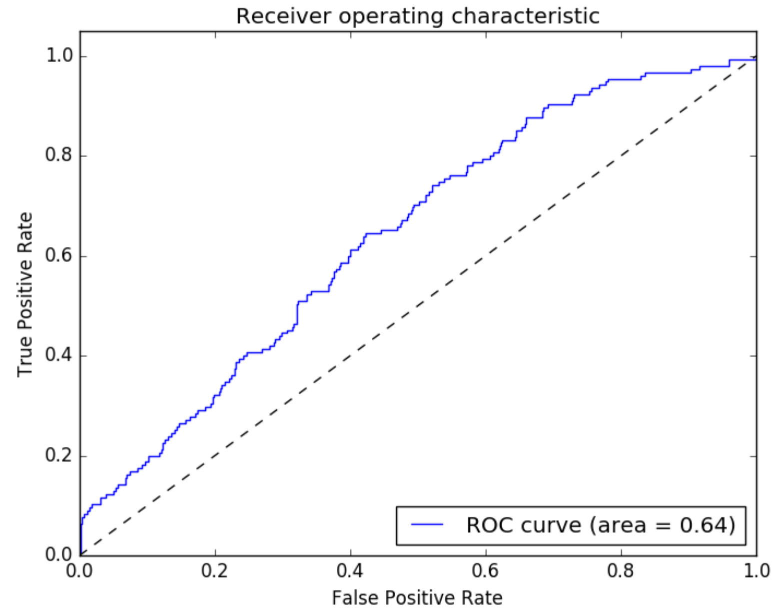
Attempt 1

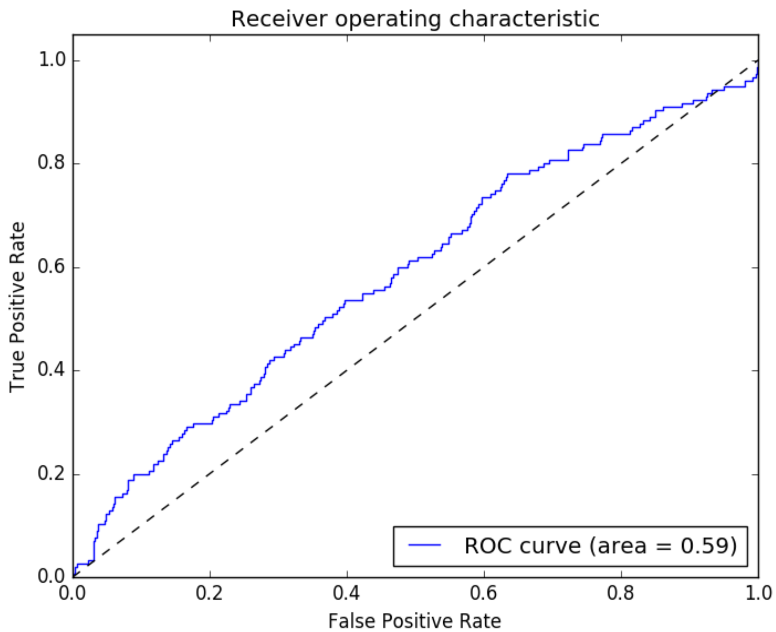
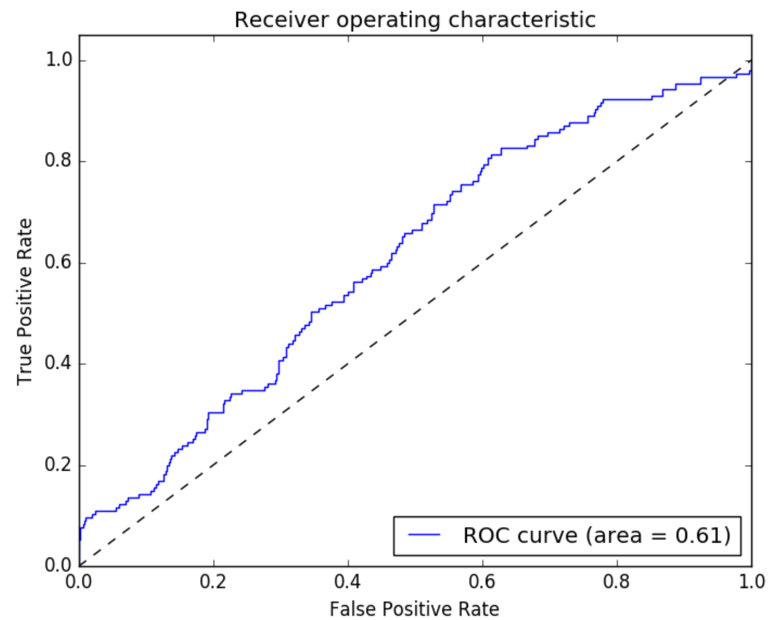


Attempt

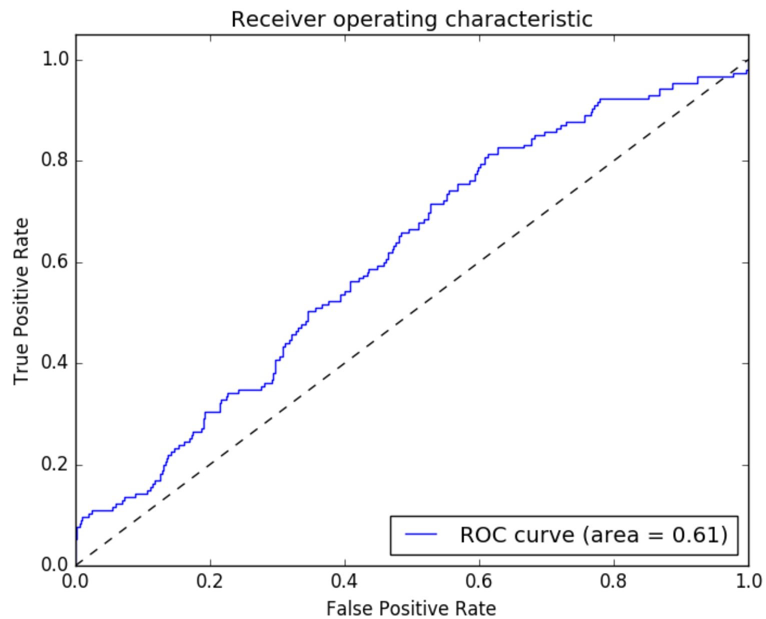
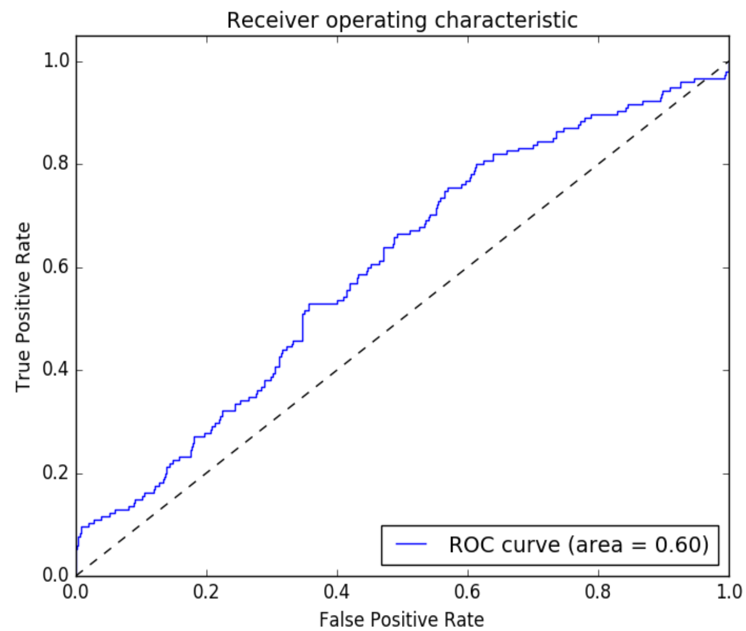
I did finally got an acceptable result, as I will show in section (b).

b)
There is no pairing to these, I'm just putting it in table form so I can show multiple per page.

	
eta = 0.01, mu = 0.00 ROC = 0.61	eta = .05 mu = .01 ROC = 0.64
This is run using default values. It will be the baseline for further observations.	ROC is slightly higher than the default values. Mu lets us take into account L2 regularization, which helps prevent overfitting (allowing more accurate predictions because overfitting can cause issues with strange prediction patterns).

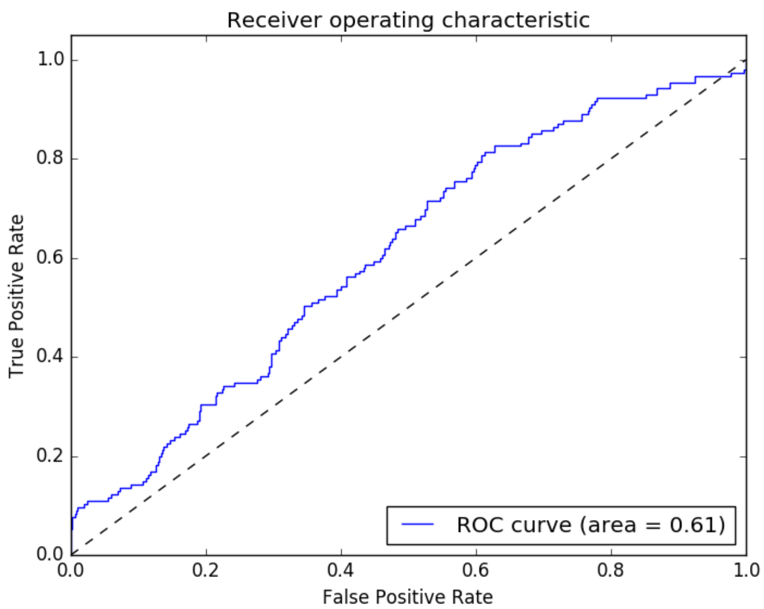
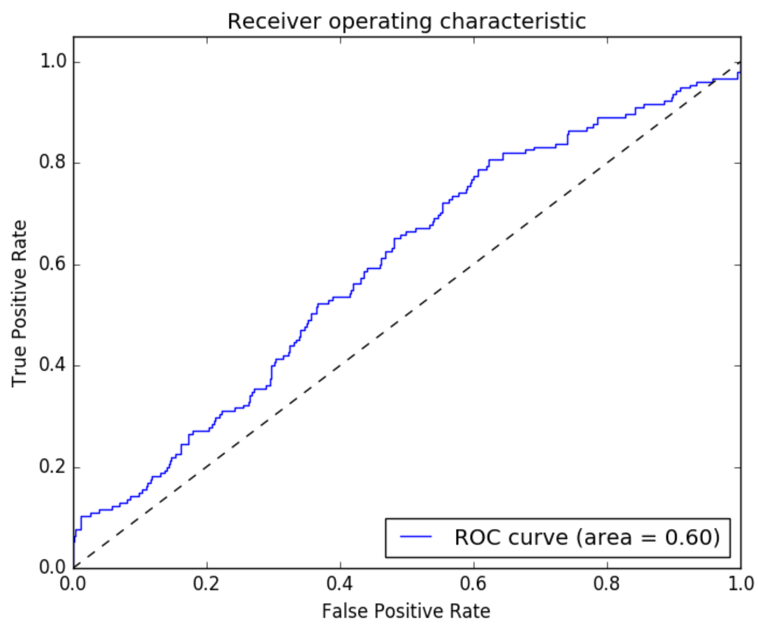
<table>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>eta = .50, mu = .05 ROC = .59</td>
<td>eta = .01 mu = .50 ROC = .61</td>
</tr>
<tr>
<td>A large eta (large learning rate) caused a worse ROC. Large learning rates are faster, but can cause problems with steps too large. They may make the algorithm unable to find the minimum / converge.</td>
<td>Using a small eta and a large mu also caused a degradation from small eta, small mu. This is possibly caused by putting too much emphasis on the correction factor of L2, making the algorithm off for weights as well.</td>
</tr>
</table>

2.4
c)
eta = .01 mu = .50 ROC = .60

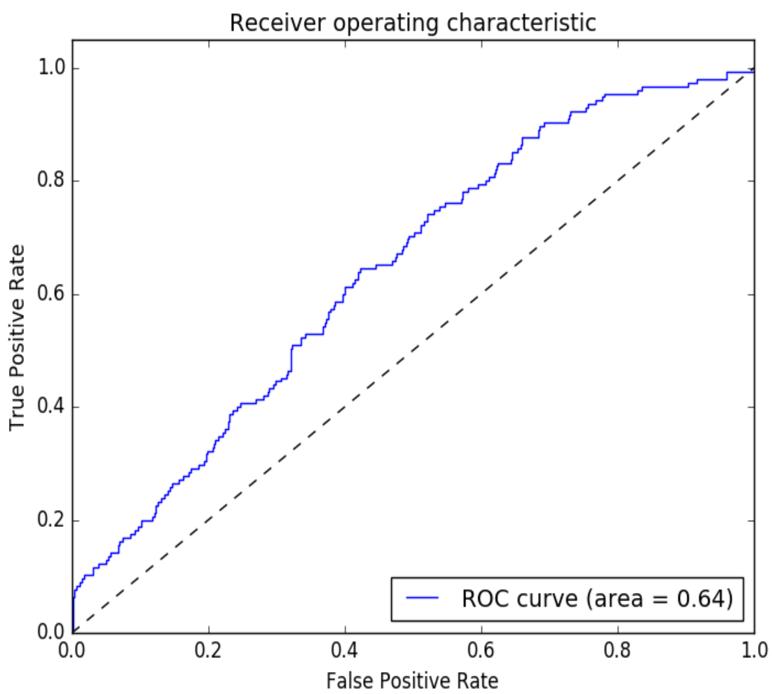
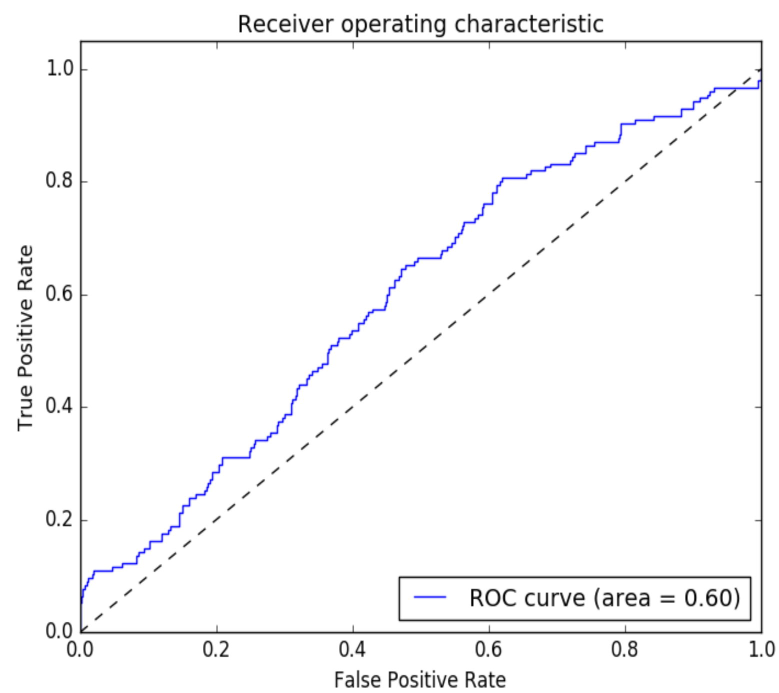
Without Hadoop	With Hadoop
	

Compared to the original run with these parameters, there was a very slight degradation of ROC (.01), but no significant difference. However, it seems to be slightly more consistent across the board compared to the original graph.

Default eta = 0.01, mu = 0.00 ROC = .60

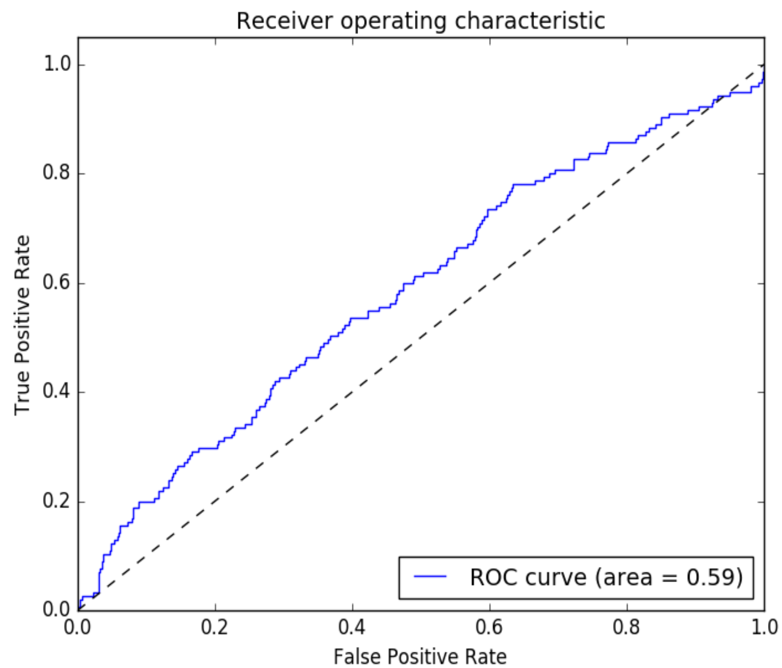
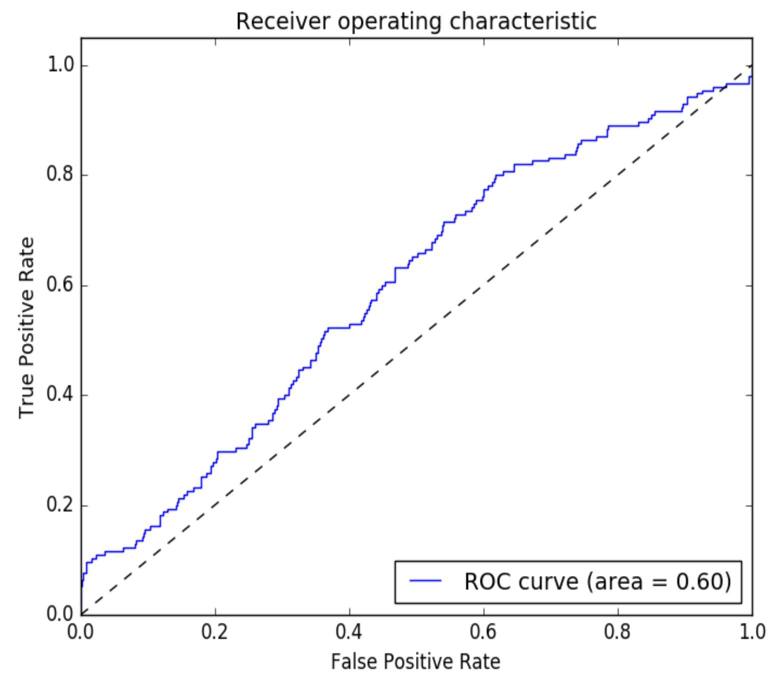
Without Hadoop	With Hadoop
	

eta = .05 mu = .01 ROC = 0.60

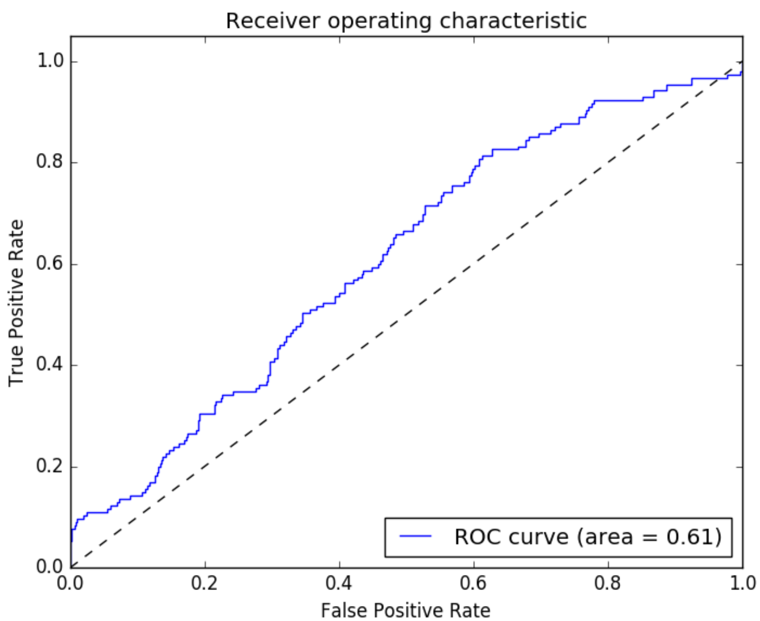
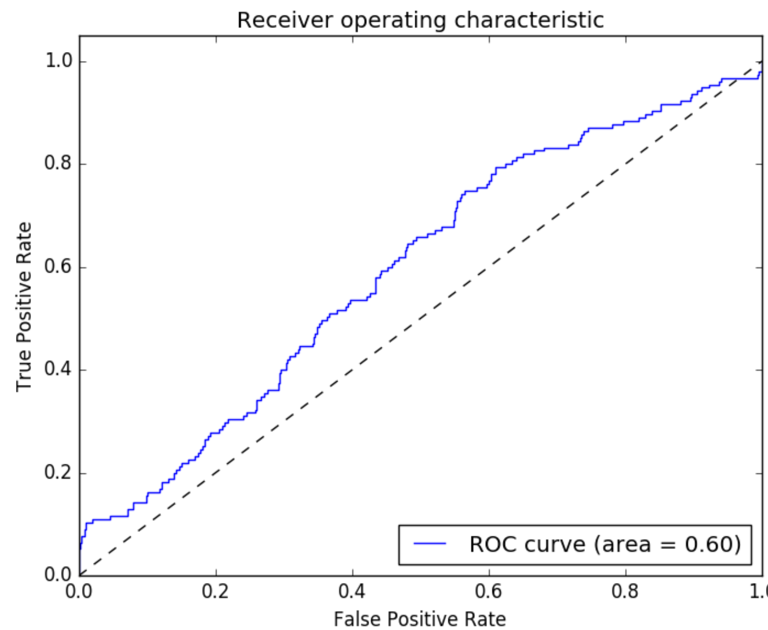
Without Hadoop	With Hadoop
	

Again, in comparison to the run without Hadoop, there was a slight degradation in ROC but slightly more consistent curve.

eta = .50, mu = .05 ROC = .60

Without Hadoop	With Hadoop
	

eta = .01, mu = .50, ROC = .60

Without Hadoop	With Hadoop
	

Note that all of the curves with Hadoop ended up with an ROC of .60. I also noted a couple of times that the curves looked slightly more consistent than without Hadoop. I conclude that with Hadoop, the predictions may cause a slight degradation of ROC, but sacrifices that for a more consistent conclusion.