CSE6250: Big Data Analytics in Healthcare
Homework 2
Rajesh Pothamsetty

Logistic Regression

Batch Gradient Descent

NLL(D, w) = $-\sum_1^N[(1-\ y_t\ )\ log\big(1-\ \sigma(w^TX_t)\big)\ +\ y_t(\log\ \sigma(w^TX_t)]$

$$= -\sum_1^N[log\big(1-\ \sigma(w^TX_i)\big)\ +\ y_i(log\frac{\sigma(w^TX_i)}{1-\ \sigma(w^TX_i)}]$$

$$= -\sum_1^N[-log\big(1+\ e^{(w^TX_i)}\big)\ +\ y_i(w^TX_i)]$$

$$\frac{\partial NLL}{\partial w}\ =\ \sum_1^N[\frac{1}{1+\ e^{(w^TX_i)}}\ e^{(w^TX_i)}X_i-\ y_iX_i]$$

$$\frac{\partial NLL}{\partial w}\ =\ \sum_1^N[\sigma(w^TX_i)X_i-\ y_iX_i]$$

$$\frac{\partial NLL}{\partial w}\ =\ \sum_1^N[\sigma(w^TX_i)-\ y_i]X_i$$

Stochastic Gradient Descent

a) $log-likelihood\ =\ [(1-\ y_t\ )\ log\big(1-\ \sigma(w^TX_t)\big)\ +\ y_t(log\ \sigma(w^TX_t)]$

b) $w_t=\ w_{t-1}+\ \eta(y_t-\ \sigma(w^TX_t))X_t$

c) $Time\ Complexity\ for\ one\ sample$ (m = total no. of features)
$= O(no.of\ non-zero\ features)$
$= \boldsymbol{O(Avg\ no.of\ non_{zero}features)}$
For all the weight updates over N samples = O(N*Avg), where N size of the data set.
If very sparse, it is close O(N)

d) Large values of η implies larger step size of the weight of a feature along the direction of gradient in a manner that could possibly jump over the minima/optimality we are looking

for. Smaller η implies smaller step size and the optimization may take forever to converge.

e) $w_t = w_{t-1} + \eta[(y_t - \sigma(w^T X_t))X_t - 2\mu W^T]$
Since weight needs to be regularized irrespective of zero or non-zero feature, time complexity = O(m), where **m** is total number of features.
Over all the sample for all the weights: O(Nm)Type equation here.

f) Lazy SGD Implementation of Logistic Regression

- Let m = 0
- Weight (W) is a list of weights #initialized with zeros
- H is a hash table or list of zeroes and the size of number of features, H stores value of **m,** when W[i] is updated.
- For each example of X_t, y_t
    - m = m + 1
    - For each non-zero feature in $\boldsymbol{X_t}$ with index **i:**
        - W[i] = W[i]* ((1 - $2\eta\mu$) ^(m – H[i])
        - W[i] = W[i] + $\eta * (y - p(X))X[i]$
        - H[i] = m #update H with m, the recent weight update to feature
- For all the features update weights with pending regularization
    - W[i] = W[i]* ((1 - $2\eta\mu$)^(m – H[i])

Note: In general, the above algorithm should be iterated over the number of epochs, but in the present homework we are only optimizing the algorithm over the learning rate and cost

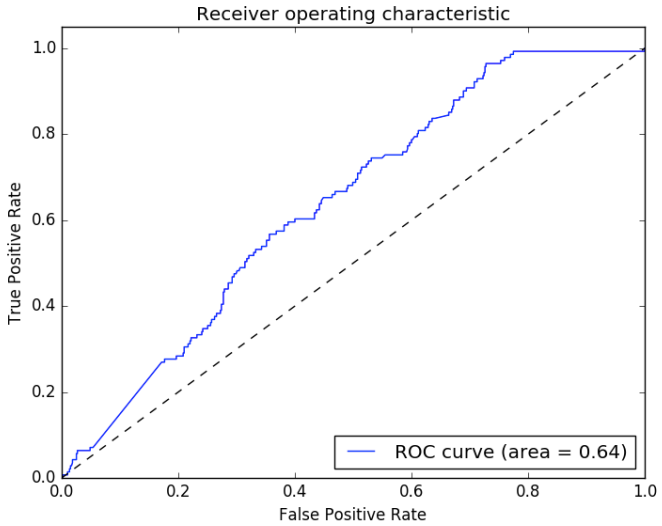
PROGRAMMING

Descriptive Statistics

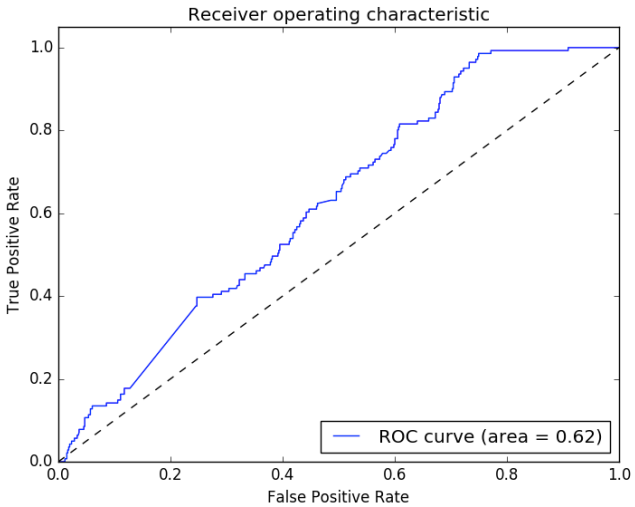
Metric	Deceased patients	Alive patients
Event Count		
1. Average Event Count	1027.74	683.16
2. Max Event Count	16829	12627
3. Min Event Count	2	1
Encounter Count		
1. Average Encounter Count	24.84	18.7
2. Max Encounter Count	375	391
3. Min Encounter Count Record Length	1	1
Record Length		
1. Average Record Length	157.04	194.7
2. Median Record Length	25	16
3. Max Record Length	5364	3103
4. Min Record Length	0	0
Common Diagnosis	DIAG320128 DIAG319835 DIAG313217 DIAG197320 DIAG132797	DIAG320128 DIAG319835 DIAG317576 DIAG42872402 DIAG313217
Common Laboratory Test	LAB3009542 LAB3023103 LAB3000963 LAB3018572 LAB3016723	LAB3009542 LAB3000963 LAB3023103 LAB3018572 LAB3007461
Common Medication	DRUG19095164 DRUG43012825 DRUG19049105 DRUG956874 DRUG19122121	DRUG19095164 DRUG43012825 DRUG19049105 DRUG19122121 DRUG956874

SGD Logistic Regression

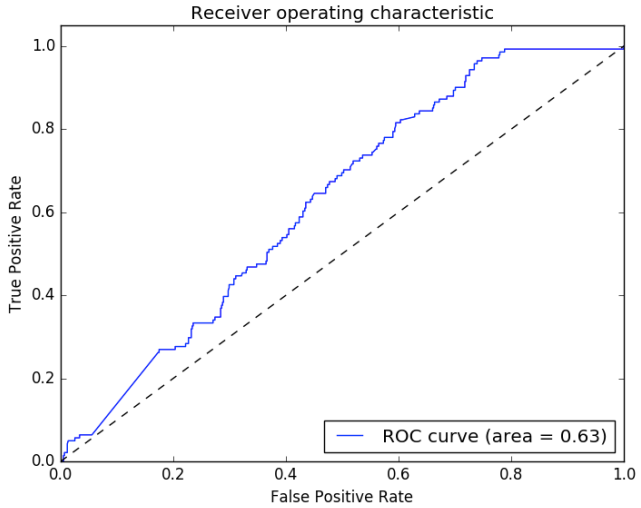
$\eta = 0.01;\ \mu = 0.0$



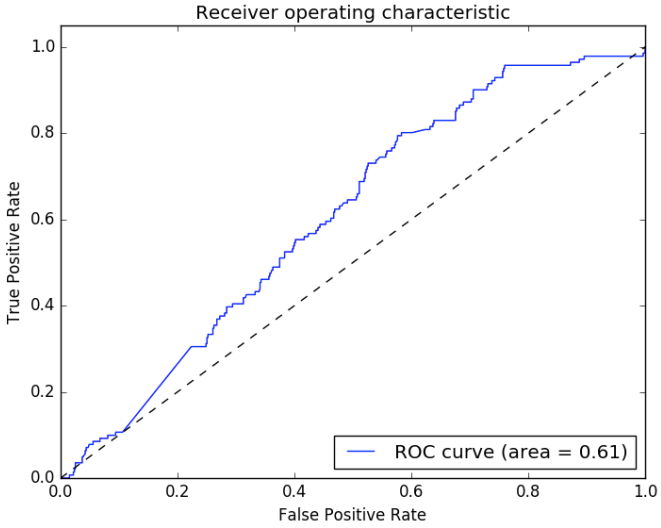
$\eta = 0.05;\ \mu = 0.0$



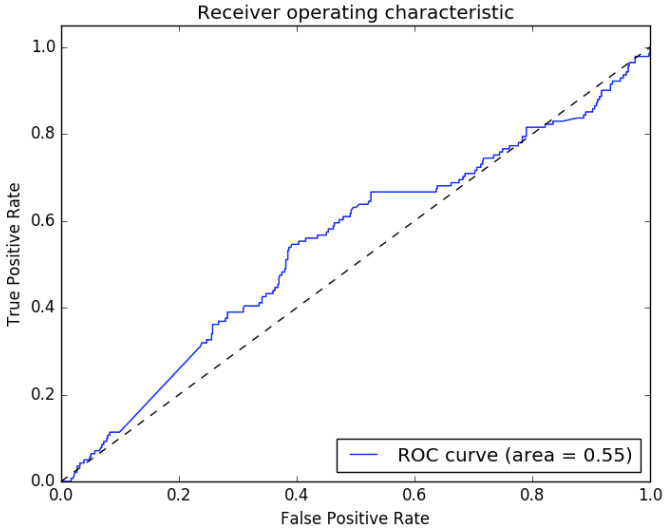
$\eta = 0.01;\ \mu = 0.05$



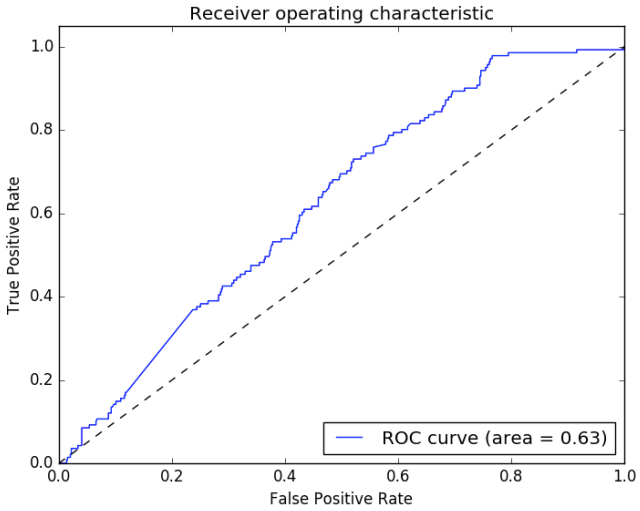
$\eta = 0.05;\ \mu = 0.05$



$\eta = 0.1;\ \mu = 0.05$



$\eta = 0.1;\ \mu = 0.01$

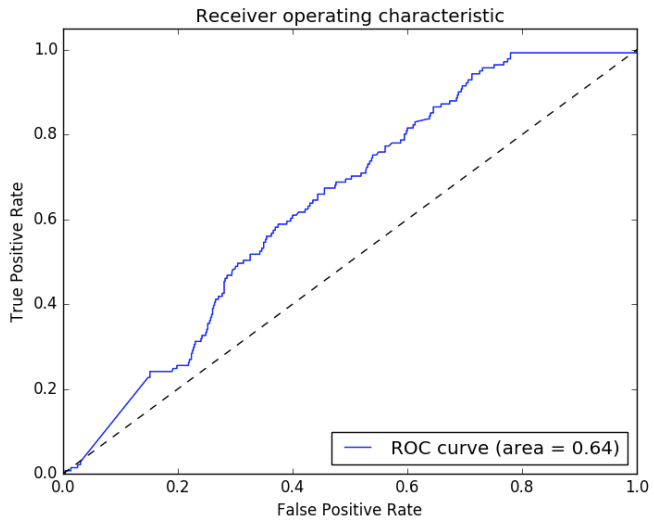


2.3.b Observations

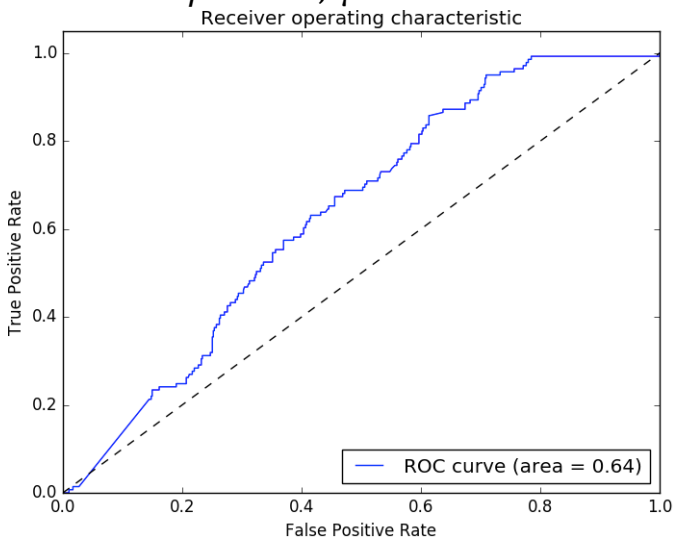
- Keeping regularization constant(mu), **eta** has a negative effect for values greater than 0.1.
- **eta** has observable effect compared to **mu**.
- Increasing **mu** has a negative effect on the ROC. **Mu** helps prevent over-fitting when large number of features are available, hence **mu**'s insignificant affect seen here must be examined.

Hadoop – Ensemble Results

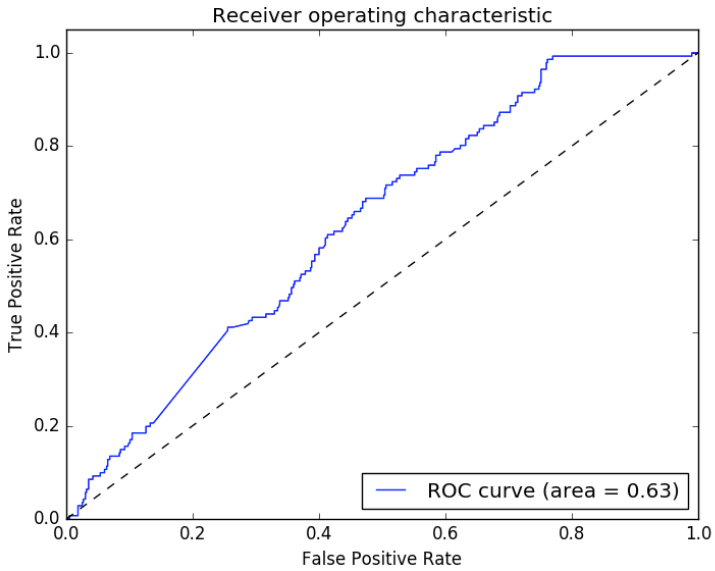
$\eta = 0.01;\ \mu = 0.0$



$\eta = 0.01;\ \mu = 0.05$



$\eta = 0.2;\ \mu = 0.0$



2.4.c Observations

- Ensemble learner had the ROC isn't much different compared the LRSGD from the previous section.
- Regularization constant has no observable affect.
- Compared to single LRSGD learner Ensemble learner is less affected by increasing the learner rate (eta). The averaging of all the ensemble classifiers minimizes the effect of overshooting of optimizer over the minima with high learning rates.

Lazy LRSGD:

OBSERVATIONS:

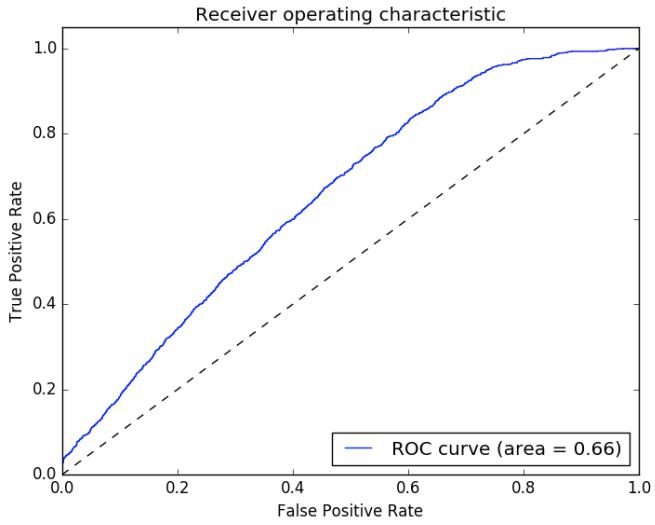
1. Improvement in the ROC from 0.66 to 0.76, refer the images below.
2. Efficient w.r.t time taken for training the data

Lazy LRSGD(seconds)	LRSGD(seconds)
90.6878430843	126.213227987

a.

Original LRSGD on the new training.data

$\eta = 0.01;\ \mu = 0.00$



LRSGD_Fast on the new training data

$\eta = 0.01;\ \mu = 0.00$