Machine Learning Assignment-3

- Sudhamayi Nanduri

Linear Regression:

(A) Using Stochastic Gradient Descent

Procedure:

1. The z-scores of the spambase dataset are calculated using the following formula.
   z-score(i) = (x(i) - mu)/sd where,
   mu = mean of all features and sd = standard deviation of all features
2. This z-score data set is partitioned into training and testing sets, with all the records ending in 0 in the testing set and the rest into training set.
3. The Lambda constant value is fixed to a value.
4. The convergence criteria is that difference between the gradient descents of 2 successive iterations is negligible (almost equal to 0).
5. All the weights are initialized to 0.
6. The following values are calculated
   (i) h(wx) = sum(w(i) * x(i)) for i = 0 to D where w(i) is weight of feature xi
   (ii) Gradient Descent = (h(wx) – y) * x(i) for every feature x(i)
   (iii) Error per record = $(h(wx) - y)^2$
7. The weights are updated using the following formula.
   New weight, w(i) = Old weight, w(i) – (lambda * Gradient Descent)
8. The SSE, MSE and RMSE values are calculated for every iteration.
9. The error rate, false positive and false negative error rates are calculated.
10. The ROC graph is plotted and the AUC is calculated.

The following observations are made by varying the lambda value from 0.00001 to 0.001 for a fixed convergence value.

Note: In this case, the convergence criterion i.e., diff between successive gradients of 2 iterations is being checked for < 0.0000001

Lambda Constant	**Iteration At which Convergence occurs**	**SSE**	**MSE**	**RMSE**
0.00001	381	541.526295173	0.13080345294	0.36166760007
0.0001	59	541.635845118	0.13082991428	0.361704180622
0.001	17	558.059432214	0.134796964303	0.367147060866

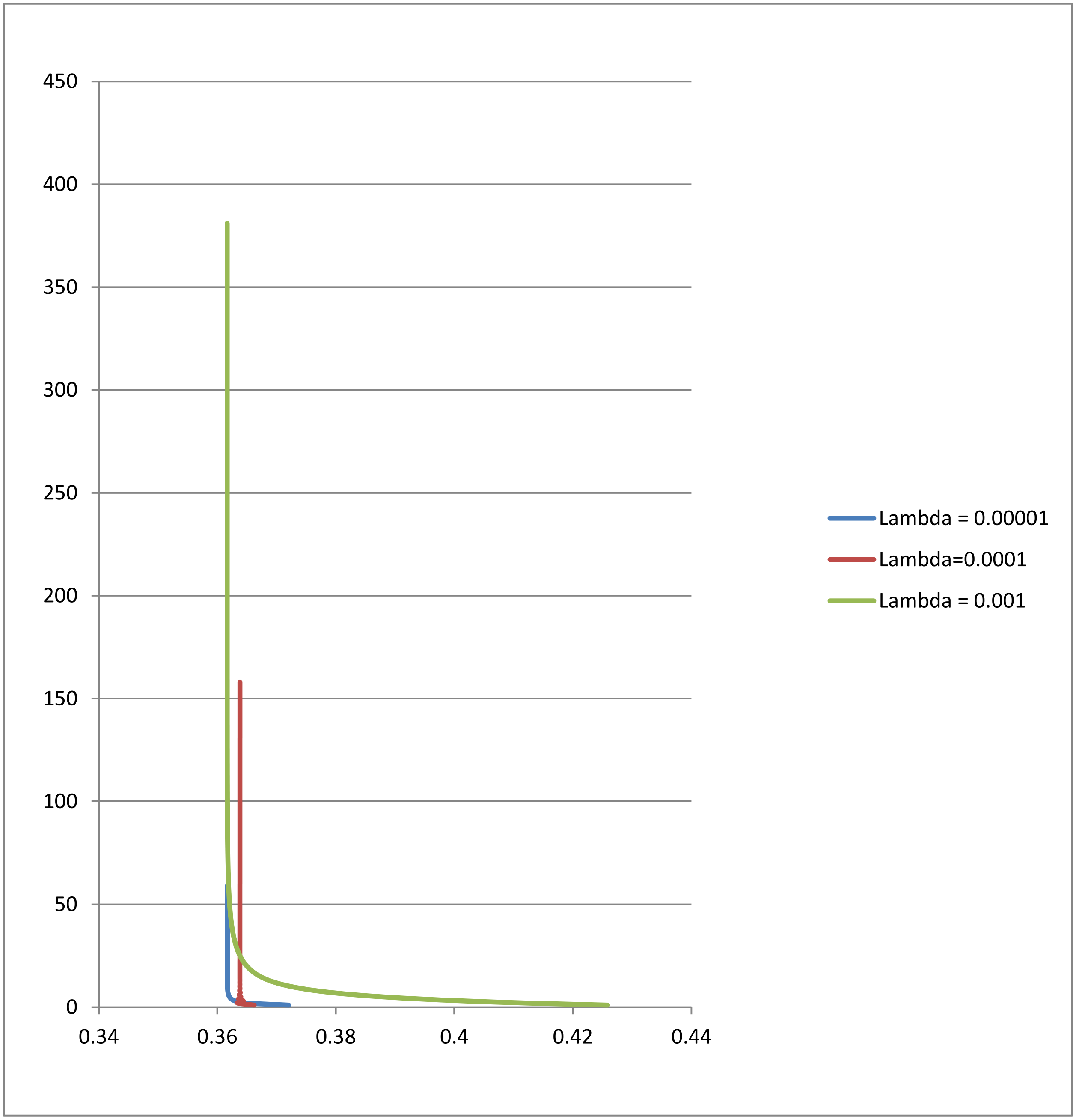
We can observe that the RMSE value is increasing with the increase in lambda value.

For lambda values of 0.1 or more, the RMSE increases exponentially and throws an overflow error.

For a lower lambda value, we observe the convergence at a higher iteration. For a higher value of lambda, the error starts increasing and throws an overflow error.

Linear Regression using Stochastic Gradient - RMSE Vs. Iteration Number:

The following graph plot shows the RMSE values (x-axis) versus iteration number (y-axis) for the above lambda values.



In the above graph, we can see that the RMSE values converge after a certain value.

(B) Using Batch Gradient Descent

The only diff in procedure from Stochastic Gradient Descent is that,

1. The sum of gradient descents for every feature for all the e-mails is calculated. Gradient Descent(i) = sum ((h(wx) – y) * x(i) for every feature x(i))
2. The weights are updated at the end each iteration.

The following observations are made by varying the lambda value from 0.00001 to 0.0002 for a fixed convergence value.

Note: In this case, the convergence criterion i.e., diff between successive gradients of 2 iterations is being checked for < 0.000001

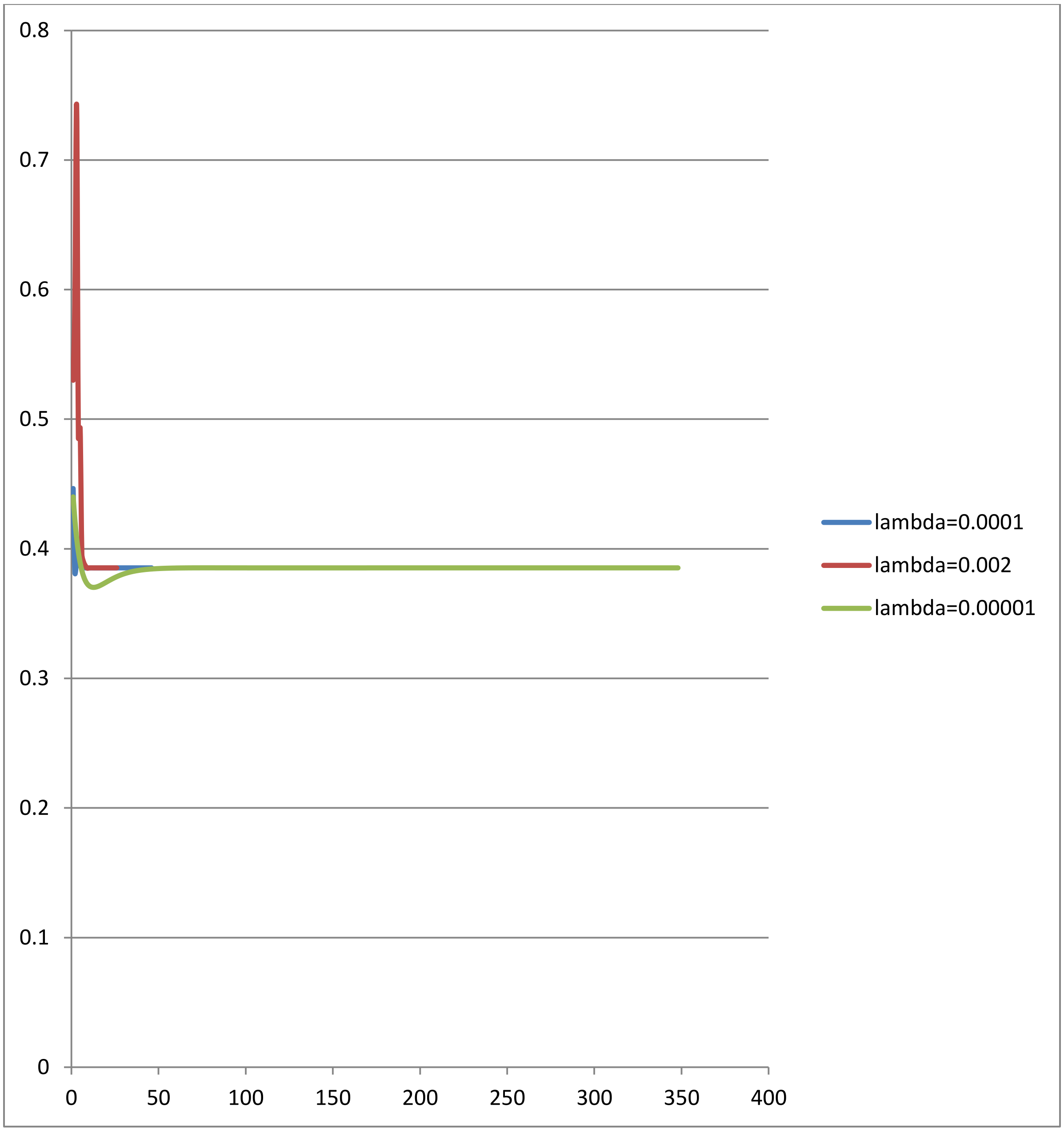
Lambda Constant	**Iteration At which Convergence occurs**	**SSE**	**MSE**	**RMSE**
0.00001	348	614.562557291	0.14844506214	0.38528557904
0.0001	46	614.562557508	0.1484450622	0.385285689067
0.0002	26	614.562558619	0.1484567812	0.385285688999

For lambda values of 0.001 or more, the RMSE increases exponentially and throws an overflow error.

For a lower lambda value, we observe the convergence at a higher iteration. For a higher value of lambda, the error starts increasing and throws an overflow error.

Linear Regression using Batch Gradient Descent - RMSE Vs. Iteration Number:

The following graph plot shows the RMSE values Vs. iteration number for the above lambda values.



From the above graph, we can see that the RMSE value converges after a certain value

Logistic Regression:

(A) Using Stochastic Gradient

Procedure:

The procedure is similar to the Linear Regression, but the following formulae are used instead.

(i) s(wx) = sum(w(i) * x(i)) for i = 0 to D where w(i) is weight of feature xi
(ii) h(wx) = 1 / (1 + exp(s(wx)))
(iii) Gradient Descent = (h(wx) – y) * h(wx) * (1 – h(wx)) * x(i) for every feature x(i)) for all emails
(iv) Error per record = $(h(wx) - y)^2$

The following observations are made by varying the lambda value from 0.01 to 1.0 for a fixed convergence value.

Note: In this case, the convergence criterion i.e., diff between successive gradients of 2 iterations is being checked for < 0.00001

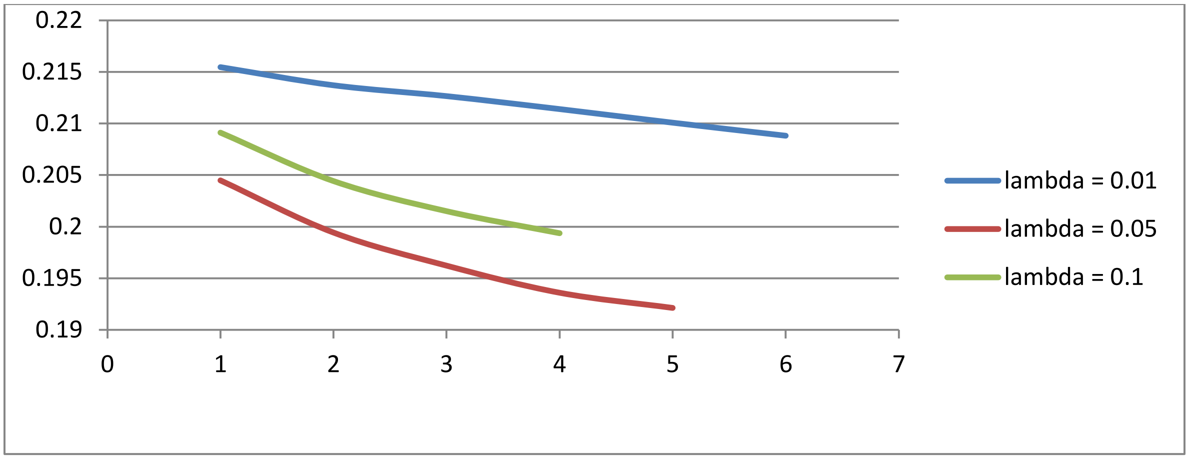
Lambda Constant	**Iteration At which Convergence occurs**	**SSE**	**MSE**	**RMSE**
0.01	13	149.867623206	0.0361999089869	0.190262736727
0.05	4	164.554150043	0.0397473792375	0.199367447788
1.0	3	176.108331521	0.0425382443287	0.206248016545

For lambda value of 10.0, we observe that the RMSE increases exponentially and throws an overflow error.

For a lower lambda value, we observe the convergence at a higher iteration. For a higher value of lambda, the error starts increasing and throws an overflow error. The RMSE value decreases with the decrease in lambda value.

Logistic Regression using Stochastic Gradient – RMSE Vs. Iteration Number:

The following graph plot shows the RMSE values Vs. iteration number for the above lambda values.



We can observe that the RMSE values converge towards the end.

(B) Using Batch Gradient Descent

The only difference in procedure from Stochastic Gradient Descent is that,

1. The sum of gradient descents for every feature for all the e-mails is calculated. Gradient Descent(i) = sum ((h(wx) – y) * h(wx) * (1 – h(wx)) * x(i) for every feature x(i)) for all the e-mails
2. The weights are updated at the end each iteration.

The following observations are made by varying the lambda value from 0.01 to 0.05 for a fixed convergence value.

Note: In this case, the convergence criterion i.e., diff between successive gradients of 2 iterations is being checked for < 0.00001

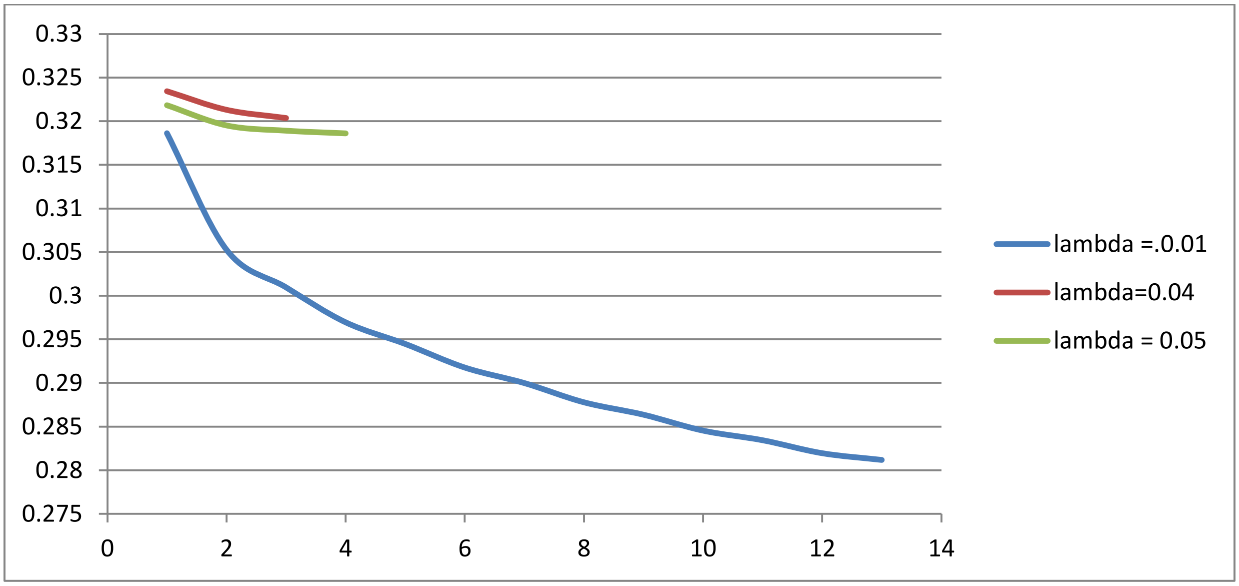
Lambda Constant	Iteration At which Convergence occurs	SSE	MSE	RMSE
0.01	13	327.327485154	0.0790646099405	0.281184298887
0.04	4	420.236750856	0.101506461559	0.318600787129
0.05	3	424.882518226	0.102628627591	0.320357031437

For lambda value of 10, we observe that the RMSE increases exponentially and throws an overflow error.

For a lower lambda value, we observe the convergence at a higher iteration. For a higher value of lambda, the error starts increasing and throws an overflow error.

Logistic Regression using Batch Gradient Descent – RMSE Vs. Iteration Number:

The following graph plot shows the RMSE values Vs. iteration number for the above lambda values.



From the above graph, we can observe the RMSE values converging after a certain value.

Conclusion from the above observations:

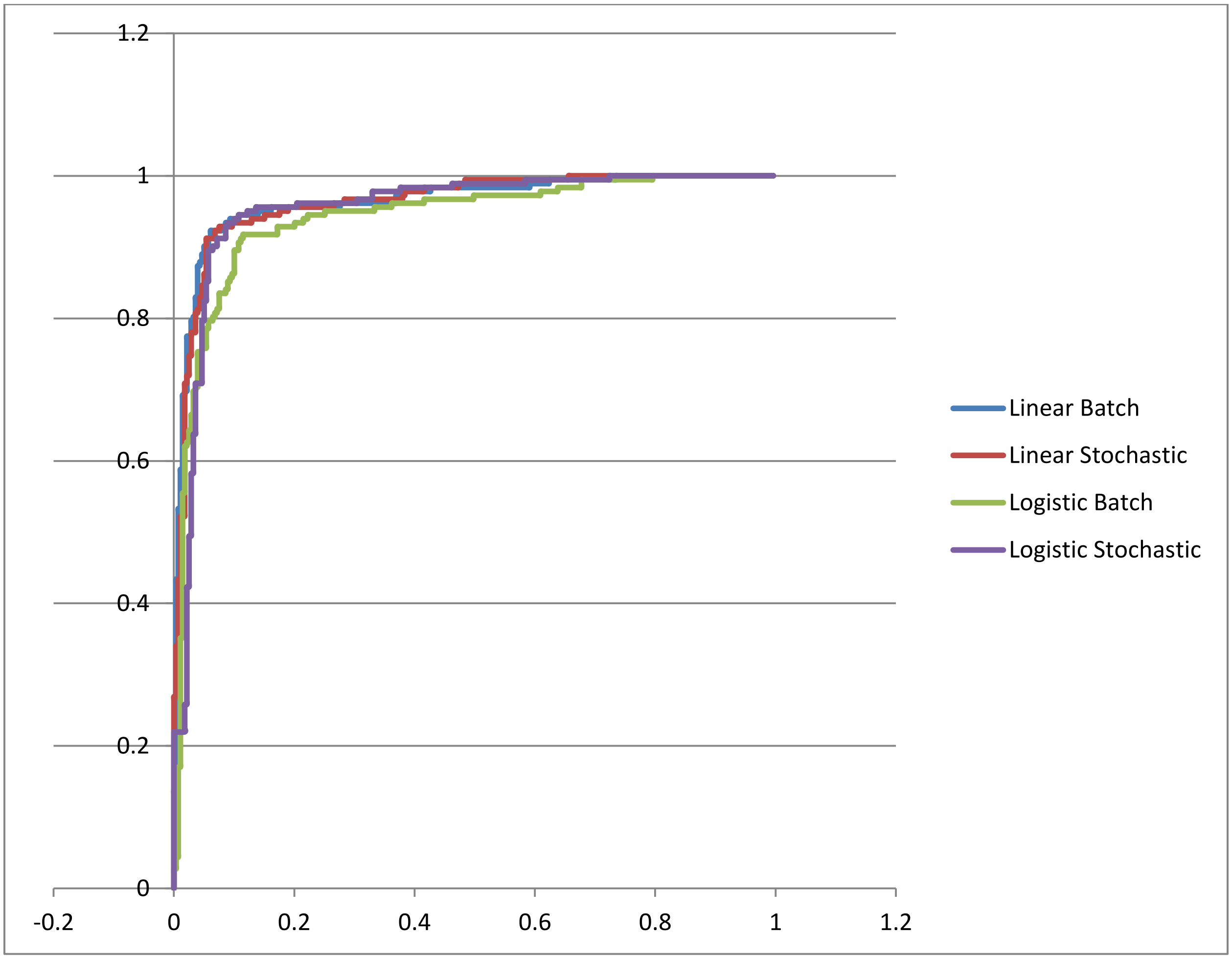
From the above observations, we can conclude that the RMSE is of the following order.

Linear Regression (Batch GD) > Linear Regression (Stochastic GD) > Logistic Regression (Batch GD) > Logistic Regression (Stochastic GD)

Hence, Logistic Regression using Stochastic Gradient Method has the least root mean square error (RMSE). Hence, it is the best method. The least performing method is Linear Regression using Batch Gradient Descent method.

ROC Graph Plot:

The following graph contains the ROC plot for Linear and Logistic Regressions.



We can observe that the ideal plot (near to 1) is Logistic Regression using Stochastic Gradient Descent method.

Area Under Curve (AUC):

The following table shows the AUC values for each type of regression.

Type Of Regression	**AUC**
Linear Regression (Using Stochastic Gradient)	0.959588798298
Linear Regression (Using Batch Gradient)	0.937374453503
Logistic Regression (Using Stochastic Gradient)	0.965280239474
Logistic Regression (Using Batch Gradient)	0.960278073181

From the above table,

Linear Regression (Batch GD) < Linear Regression (Stochastic GD) < Logistic Regression (Batch GD) < Logistic Regression (Stochastic GD)

We can see that, Logistic Regression (Using Stochastic Gradient Descent) is the best of all regression methods. It has the highest AUC value, i.e., 0.965280239474 (ideally closer to 1).

Also, from the AUC values of the distribution models from previous assignment, we can observe that the Logistic Regression using Stochastic Gradient has a higher AUC value. The AUC values for the Bernoulli, Gaussian and Histogram distributions are less than 0.95.

Hence, Logistic Regression using Stochastic Gradient is the best method according to AUC values.

Conclusion:

From all the above observations, we can see that Logistic Regression using stochastic gradient has the least root mean square error (RMSE) value. It also has the highest AUC value. Hence, it is the best regression method.

It is followed by Logistic Regression using batch gradient, which is followed by Linear Regression using stochastic gradient method. The least performing method is Linear Regression using batch gradient method, which has the highest RMSE value and least AUC value.

Hence,

Logistic Regression (Stochastic GD) > Logistic Regression (Batch GD) > Linear Regression (Stochastic GD) > Linear Regression (Batch GD)