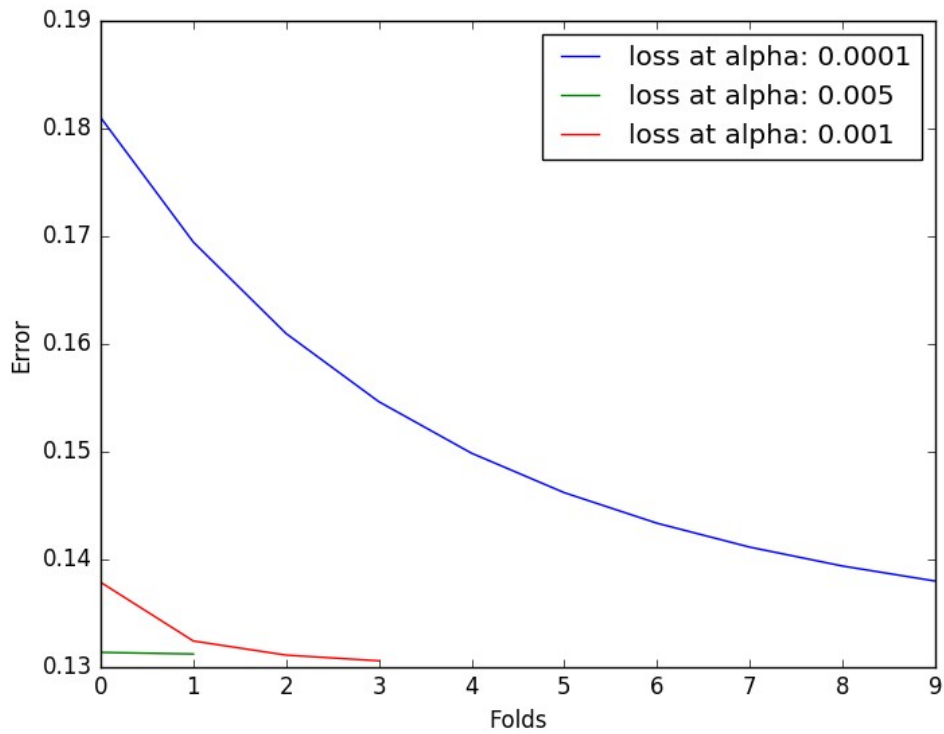


To run and view the code follow instructions on:

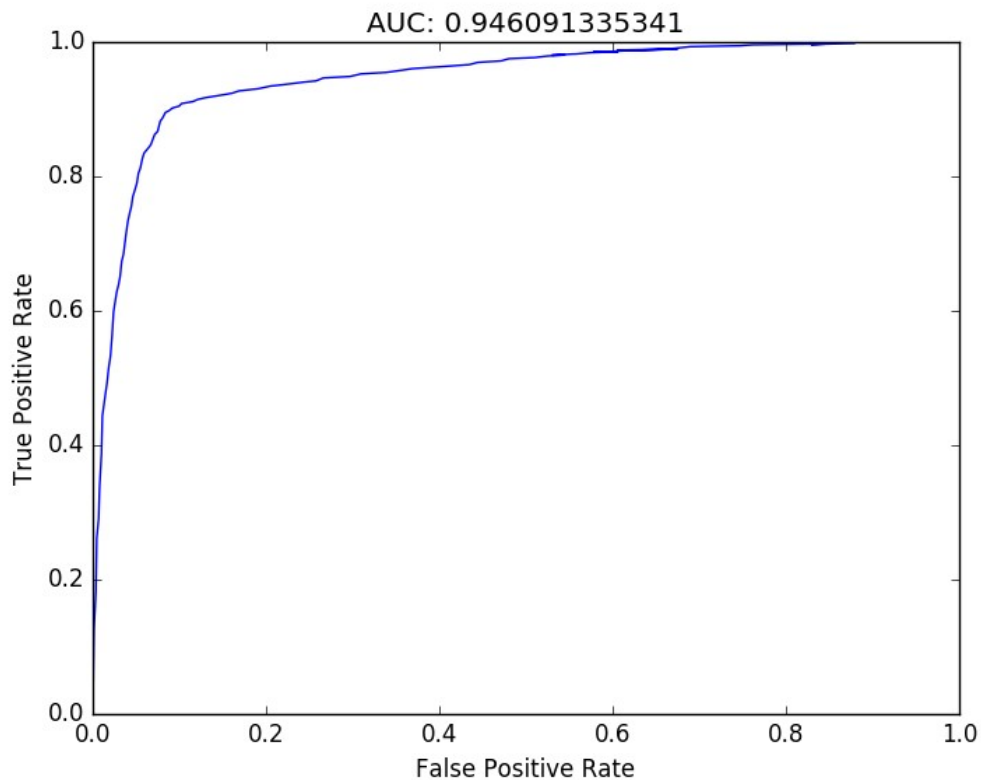
Codebase contains commented descriptions of employed methods.

This document presents the graphs that can be found in the “graphs” folder.



*Illustration 1: Stochastic Gradient Descent with linear regression trainer*

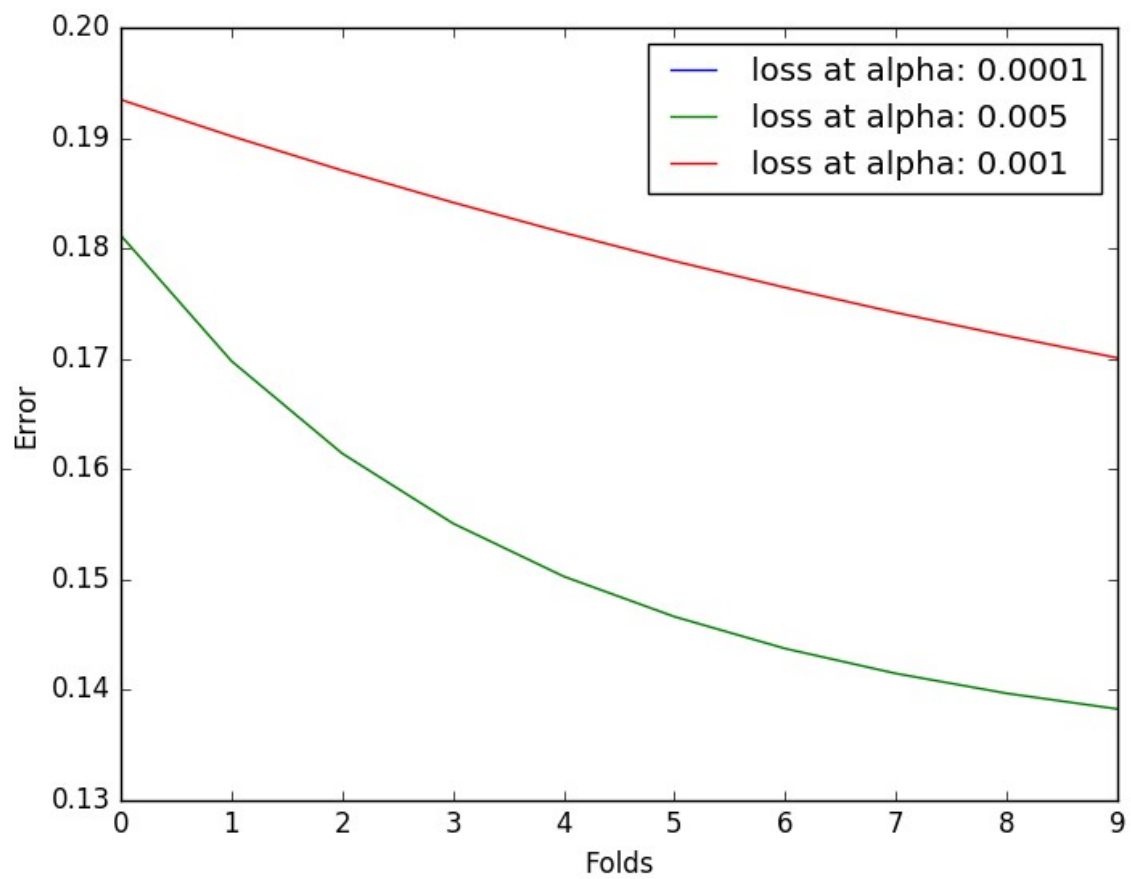
SGD terminates after maximal amount of epochs or if it detects the fall in loss is minimal. Only alpha of 0.0001 converged across 10 fold.



*Illustration 2: ROC graph with AUC in the title for SGD, linear regression*

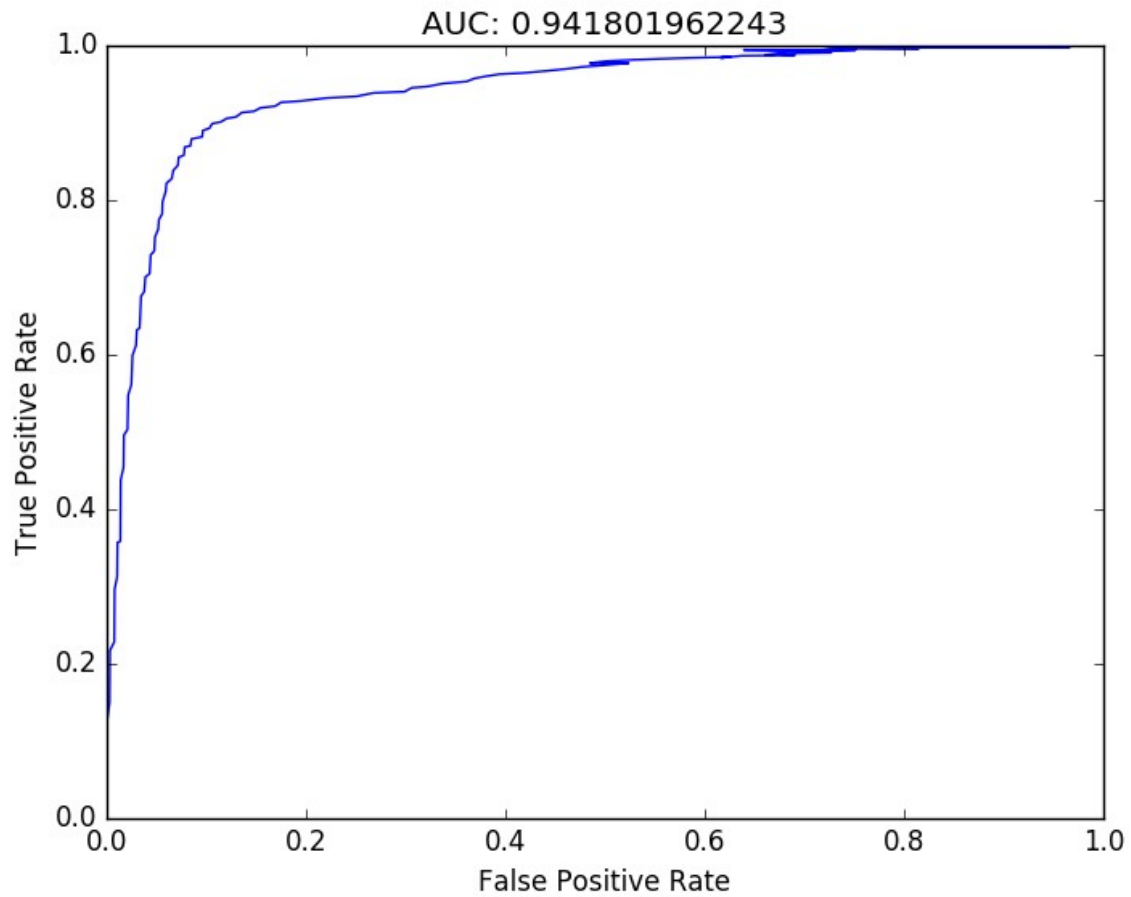
This is quite excellent performance, considering no feature engineering was required. Spam recognition systems need to however be almost perfect not to annoy users.

Training with `batch_size=1` was very slow due to the need to record loss on the whole dataset after each iteration. Batch gradient descent with sample as small as 20 was much faster. A Batch size of 920 was chosen to test the batch implementation since the full dataset of 4600 observations can be divided into 5 parts of 920 observations.



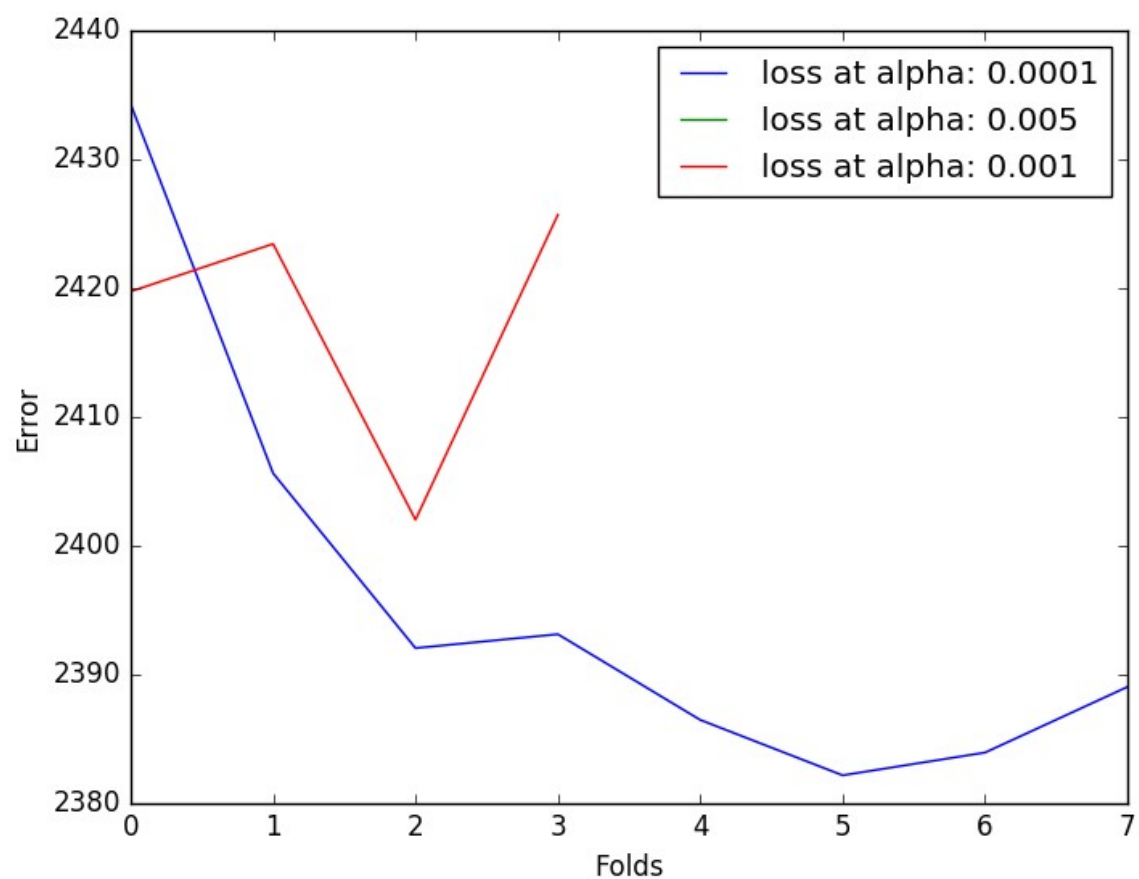
*Illustration 3: Batch gradient descent with linear regression trainer*

Alpha of 0.0001 seems to be too small with no recorded observations passing to graph drawing function. It seems batch sgD prefers higher alphas.



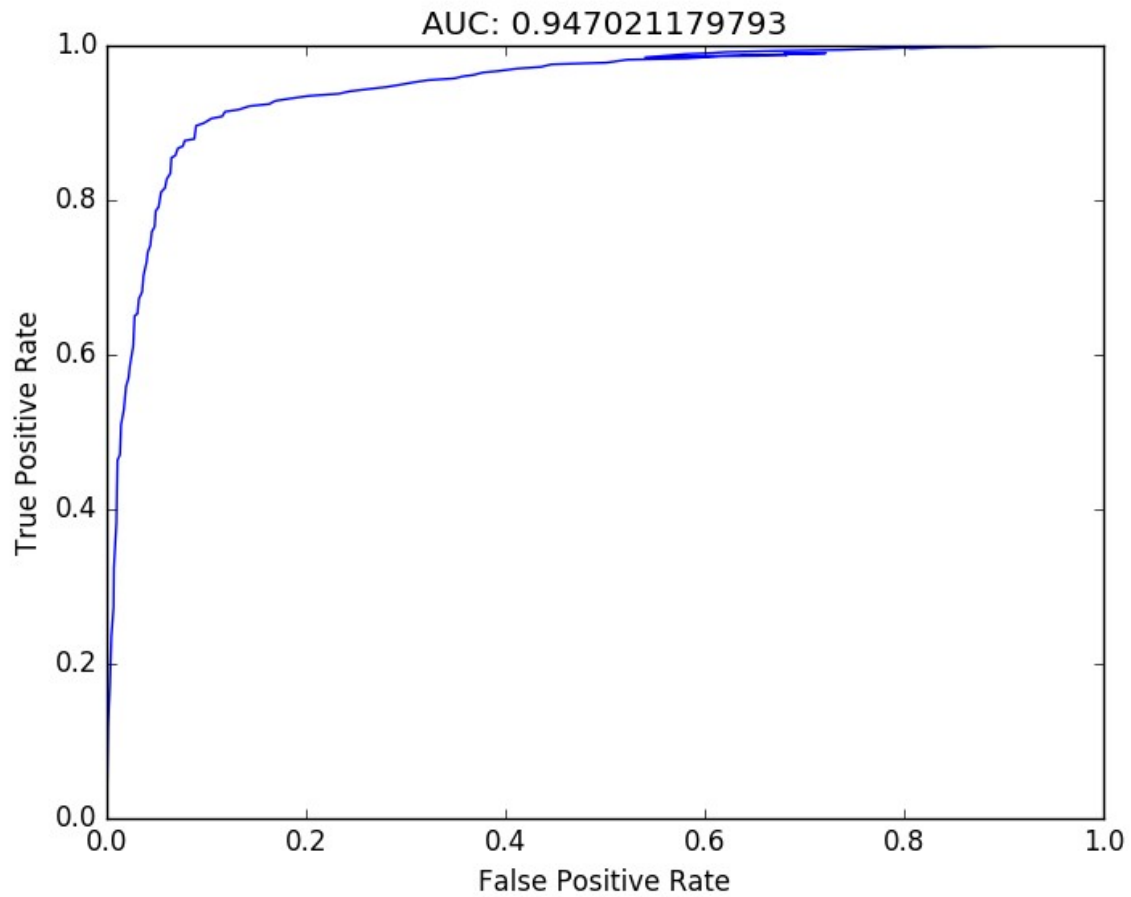
*Illustration 4: ROC and AUC of batch sgd linear regression trainer*

Performance was not hindered in run of batch descent, with best alpha of 0.005 producing the same performance as alpha of 0.0001 in the fully stochastic gradient descent.



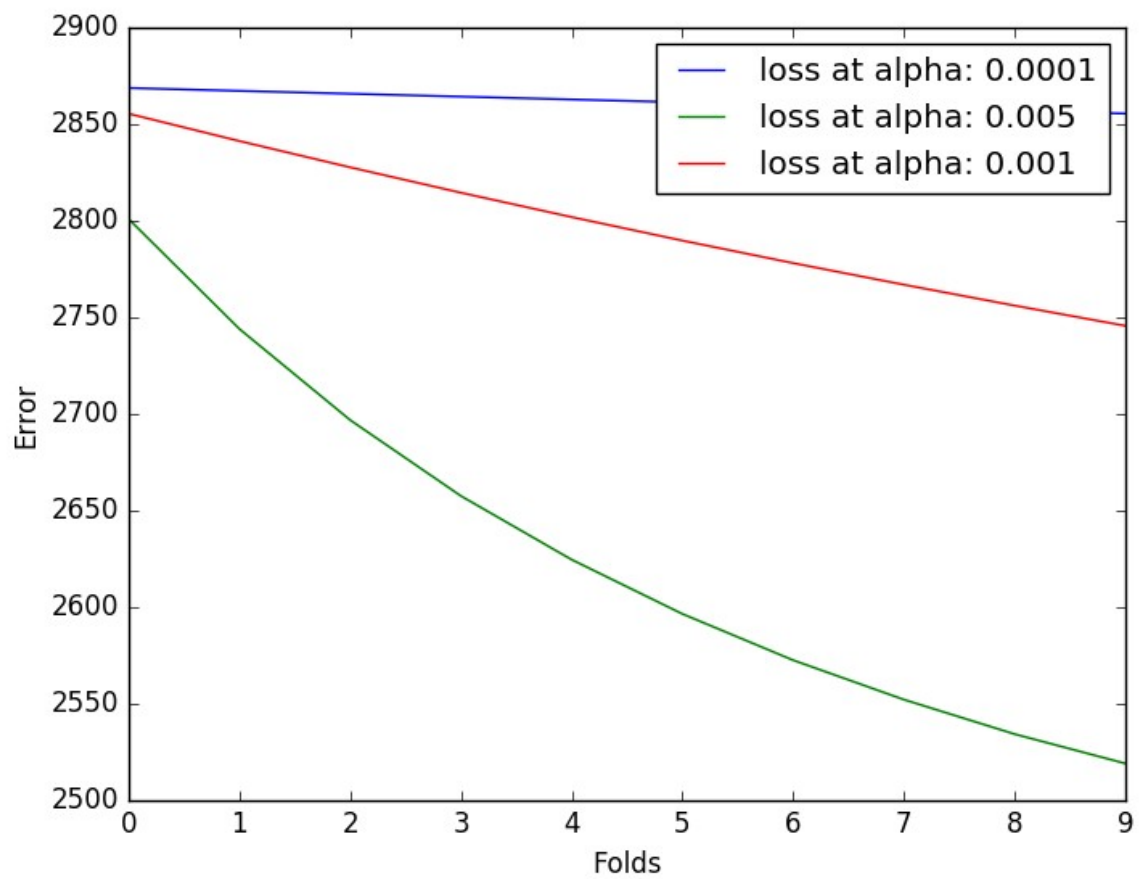
*Illustration 5: Stochastic logistic regression MSE over folds*

Logistic loss trainer seems to be more volatile than linear regression. It seems to also favour small alphas.



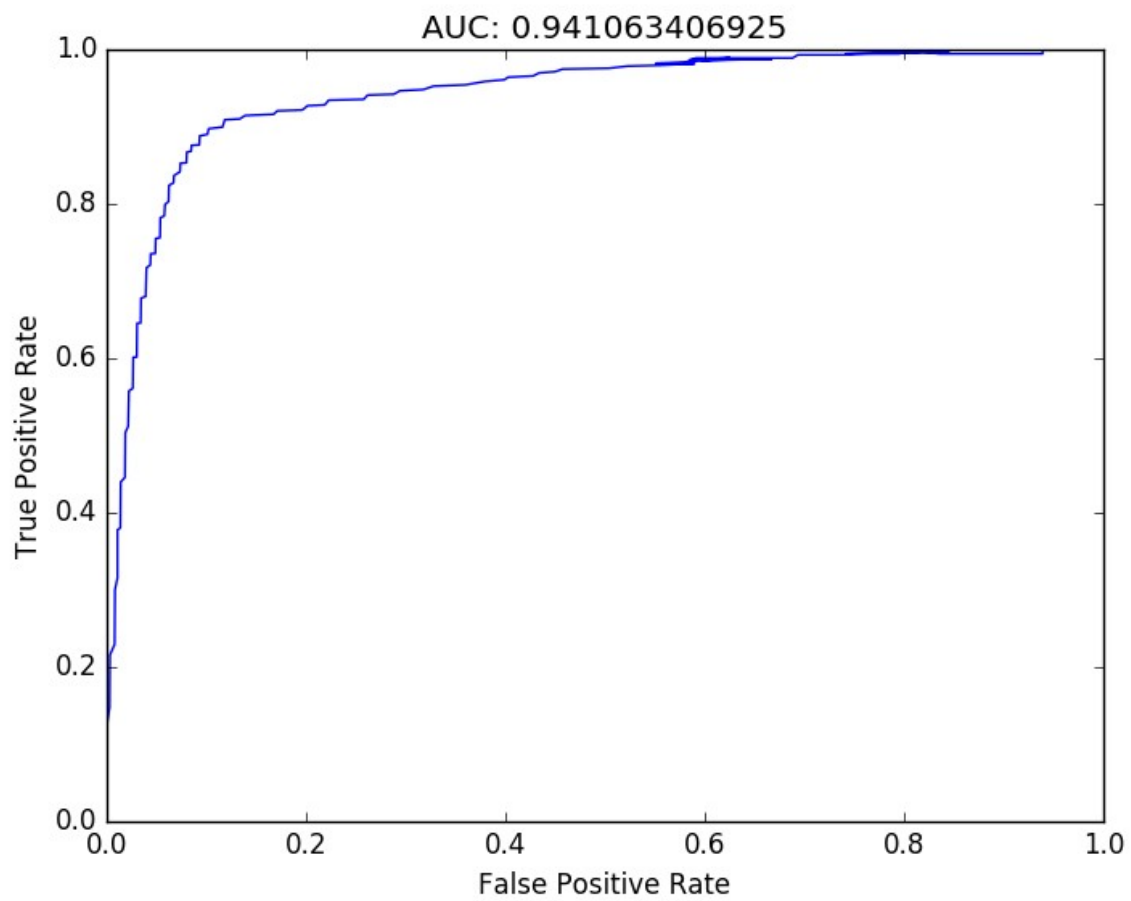
*Illustration 6: ROC and AUC of stochastic logistic regression*

Performance was again not hindered, what mean the trainer converged to the objective loss minimum.



*Illustration 7: Batch gradient descent for logistic regression trainer*

Large alphas seem to be preferred for this trainer. Convergence rate is very small for very small alpha.



Performance was again not hindered what suggests global optimum was reached.