COMPGW02: Web Economics
Assignment 1 - Part B

Sergiu Tripon
Department of Computer Science
University College London
Gower Street, London, WC1E
sergiu.tripon.15@ucl.ac.uk

1 Convergence Criterion

The convergence criterion employed throughout this assignment came in the form of a threshold. In order to establish an optimal threshold, the gradient descent was first run without a stopping condition. Mostly, the mean squared error decreased by larger portions at the very beginning of a test and then continued with a uniform, regular and slower decrease. This indicated that the curve was drawn and the gradient descent was now following a straight line (see red line on Figure 1), which means convergence was achieved. The threshold was chosen based on error values post-convergence and epochs.

During each epoch, the mean squared error was compared with the set threshold, and if the error was smaller than the threshold, the test would stop. The threshold value varied from Linear Regression to Logistic Regression and from Stochastic Gradient Descent to Batch Gradient Descent. However, the ”Last Epoch MSE” values provide an indication on the range of the threshold set.

For the purpose of the report, another stopping condition of a maximum of 500 epochs for SGD and 50 epochs for BGD was also introduced.

2 Linear Regression with Stochastic Gradient Descent [1] [2]

2.1 Training

Table 1 shows the mean squared error (MSE) on the last epoch when training Linear Regression with SGD for 500 epochs with 4 different learning rates. It was found that it produced the lowest MSE with a learning rate of 0.001.

Figure 1 provides a visual representation of the performance of Linear Regression with SGD trained with 4 different learning rates for 500 epochs.

Table 1: Linear Regression with SGD training with 4 different learning rates. Lowest MSE on the last epoch shown in **bold**.

Epochs	**Regression**	**Gradient Descent**	**Learning Rate**	**Last Epoch MSE**
500	Linear	Stochastic	0.00001	0.19106040630421942
500	Linear	Stochastic	0.0001	0.17909130473210128
500	**Linear**	**Stochastic**	**0.001**	**0.17462479941829512**
500	Linear	Stochastic	0.01	0.18145635103672708

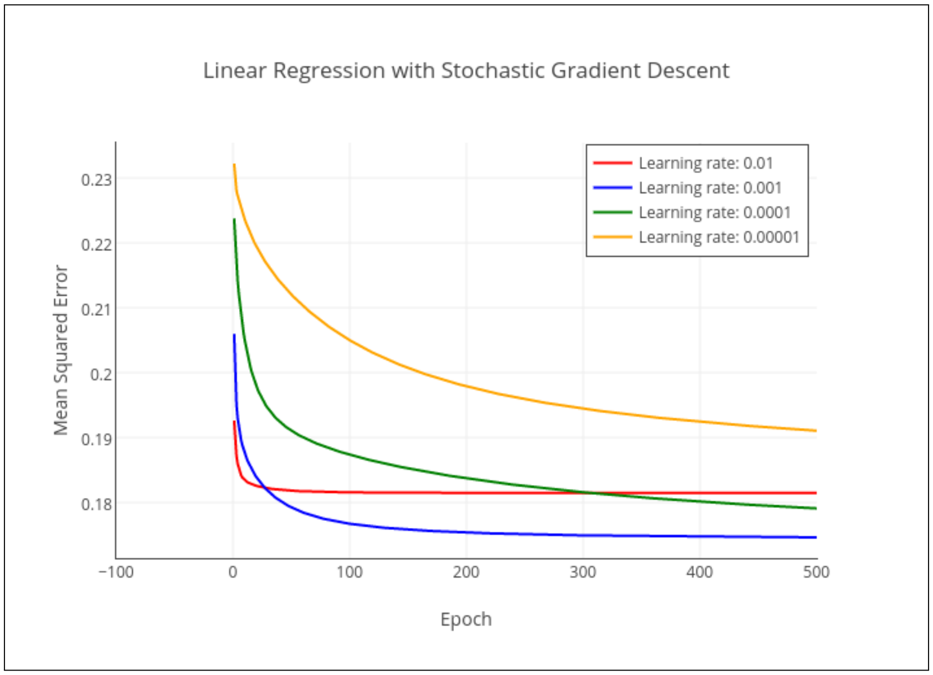

Figure 1: Linear Regression with SGD training with 4 different learning rates.

2.2 Evaluation

Receiver operating characteristic (ROC) Curve

Figure 2 shows the ROC curve of Linear Regression with SGD trained with 4 different learning rates for 500 epochs.

Figure 3 shows the ROC curve of Linear Regression with SGD trained with the optimal learning rate identified for 500 epochs.

Area under the curve (AUC)

Table 2 shows the AUC (Area under the curve) of Linear Regression with SGD trained with 4 different learning rates for 500 epochs.

Table 2: AUC of Linear Regression with SGD trained with 4 different learning rates. Highest AUC shown in **bold**.

Epochs	Learning Rate	AUC
500	0.00001	0.8475914766237341
500	0.0001	0.8810311552247047
500	**0.001**	**0.8923549568710862**
500	0.01	0.8922958761668431

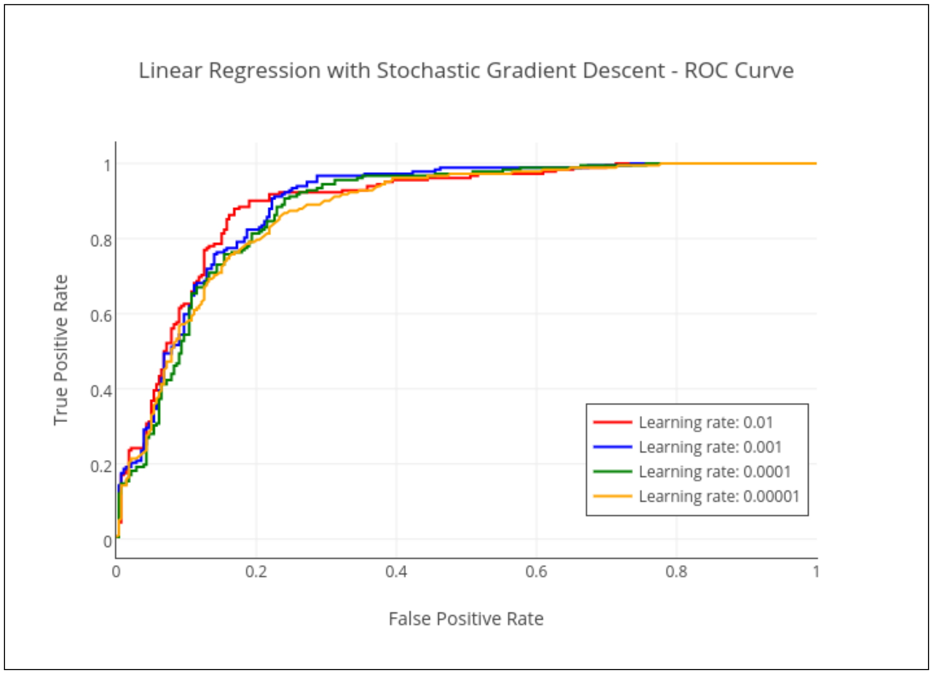

Figure 2: ROC Curve of Linear Regression with SGD training with 4 different learning rates.

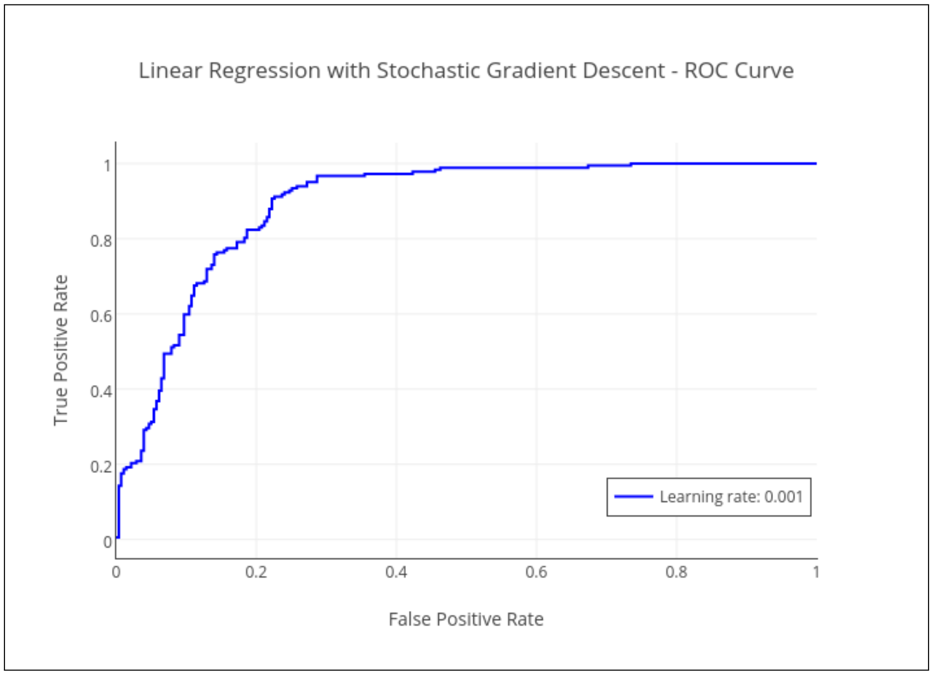

Figure 3: ROC Curve of Linear Regression with SGD training with optimal learning rate identified.

3 Linear Regression with Batch Gradient Descent [1] [2]

3.1 Training

Table 3 shows the mean squared error (MSE) on the last epoch when training Linear Regression with BGD for 50 epochs with 6 different learning rates. It was found that it produced the lowest MSE with a learning rate of 0.000001.

Figure 4 provides a visual representation of the performance of Linear Regression with BGD trained on 2 different learning rates for 50 epochs.

Table 3: Linear Regression with BGD training with 6 different learning rates. Lowest MSE on the last epoch shown in **bold**.

Epochs	Regression	Gradient Descent	Learning Rate	Last Epoch MSE
50	Linear	Batch	0.0000001	0.2494526190216533
50	**Linear**	**Batch**	**0.000001**	**0.22658756874338298**
50	Linear	Batch	0.00001	diverged
50	Linear	Batch	0.0001	diverged
50	Linear	Batch	0.001	diverged
50	Linear	Batch	0.01	diverged

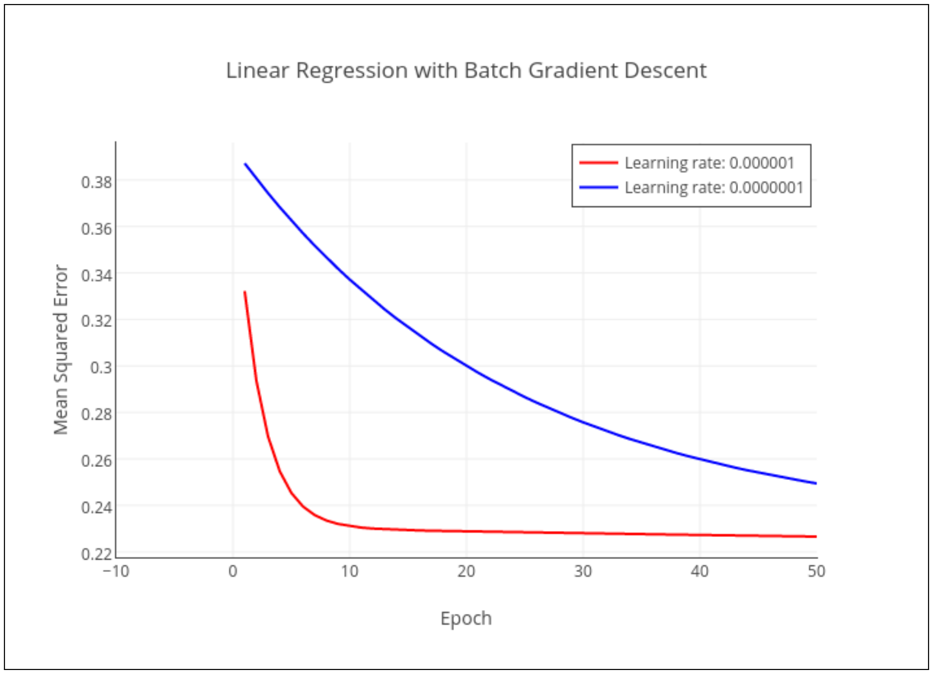

Figure 4: Linear Regression with BGD training with 2 different learning rates.

3.2 Evaluation

Receiver operating characteristic (ROC) Curve

Figure 5 shows the ROC curve of Linear Regression with BGD trained with 2 different learning rates for 50 epochs.

Figure 6 shows the ROC curve of Linear Regression with BGD trained with the optimal learning rate identified for 50 epochs.

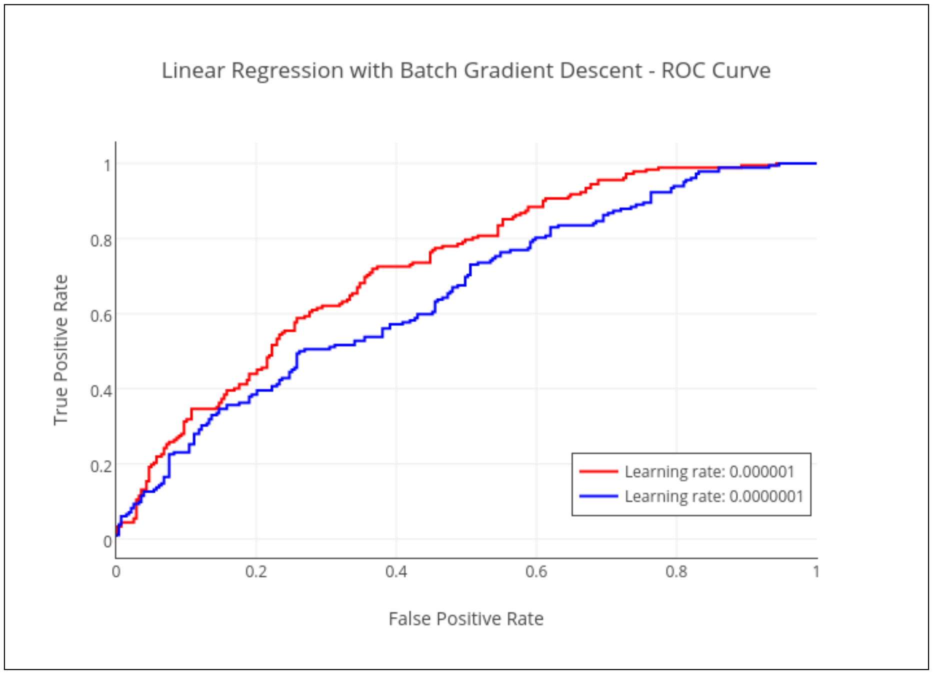

Figure 5: ROC Curve of Linear Regression with BGD training with 2 different learning rates.

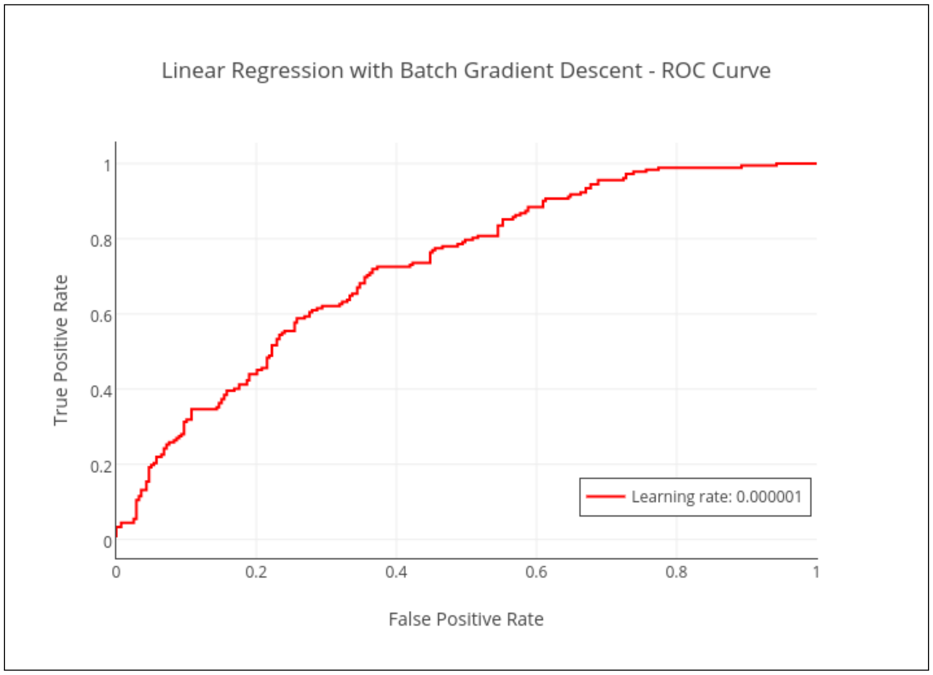

Figure 6: ROC Curve of Linear Regression with BGD training with optimal learning rate identified.

Area under the curve (AUC)

Table 4 shows the AUC (Area under the curve) of Linear Regression with BGD trained with 6 different learning rates for 50 epochs.

Table 4: AUC of Linear Regression with BGD trained with 6 different learning rates. Highest AUC shown in **bold**.

Epochs	Learning Rate	AUC
50	0.0000001	0.6521918941273781
50	**0.000001**	**0.7243491275749331**
50	0.00001	diverged
50	0.0001	diverged
50	0.001	diverged
50	0.01	diverged

4 Logistic Regression with Stochastic Gradient Descent [1] [2]

4.1 Training

Table 5 shows the mean squared error (MSE) on the last epoch when training Logistic Regression with SGD for 500 epochs with 4 different learning rates. It was found that it produced the lowest MSE with a learning rate of 0.1.

Figure 7 provides a visual representation of the performance of Logistic Regression with SGD trained with 4 different learning rates for 500 epochs.

Table 5: Logistic Regression with SGD training with 4 different learning rates. Lowest MSE on the last epoch shown in **bold**.

Epochs	Regression	Gradient Descent	Learning Rate	Last Epoch MSE
500	Logistic	Stochastic	0.0001	0.1883504718522505
500	Logistic	Stochastic	0.001	0.15310177432780345
500	Logistic	Stochastic	0.01	0.11686127839170125
500	**Logistic**	**Stochastic**	**0.1**	**0.08741143950554693**

4.2 Evaluation

Receiver operating characteristic (ROC) Curve

Figure 8 shows the ROC curve of Logistic Regression with SGD trained with 4 different learning rates for 500 epochs.

Figure 9 shows the ROC curve of Logistic Regression with SGD trained with the optimal learning rate identified for 500 epochs.

Area under the curve (AUC)

Table 6 shows the AUC (Area under the curve) of Logistic Regression with SGD trained with 4 different learning rates for 500 epochs.

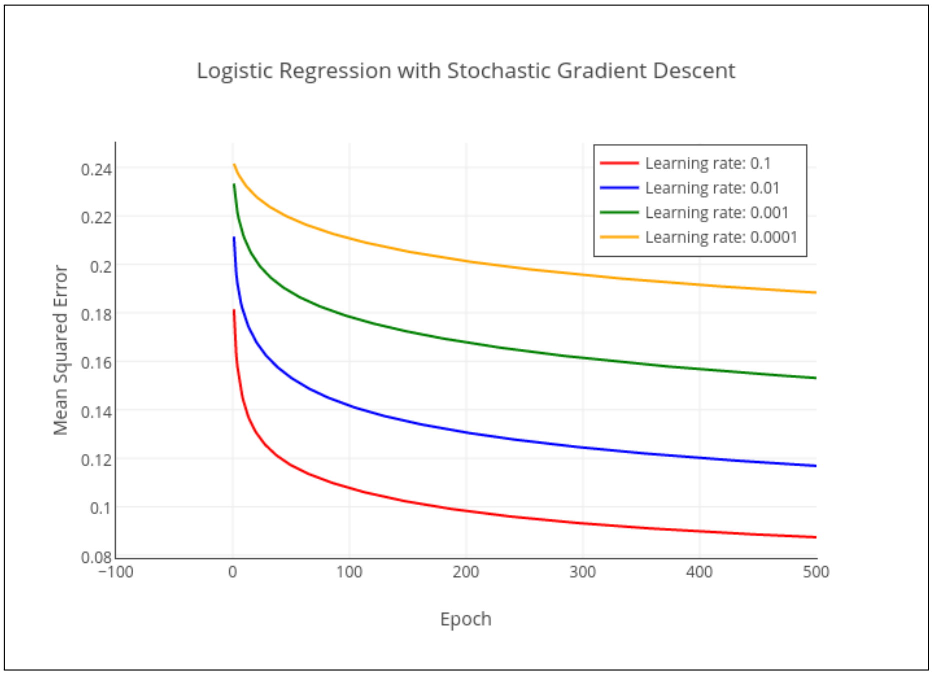

Figure 7: Logistic Regression with SGD training with 4 different learning rates.

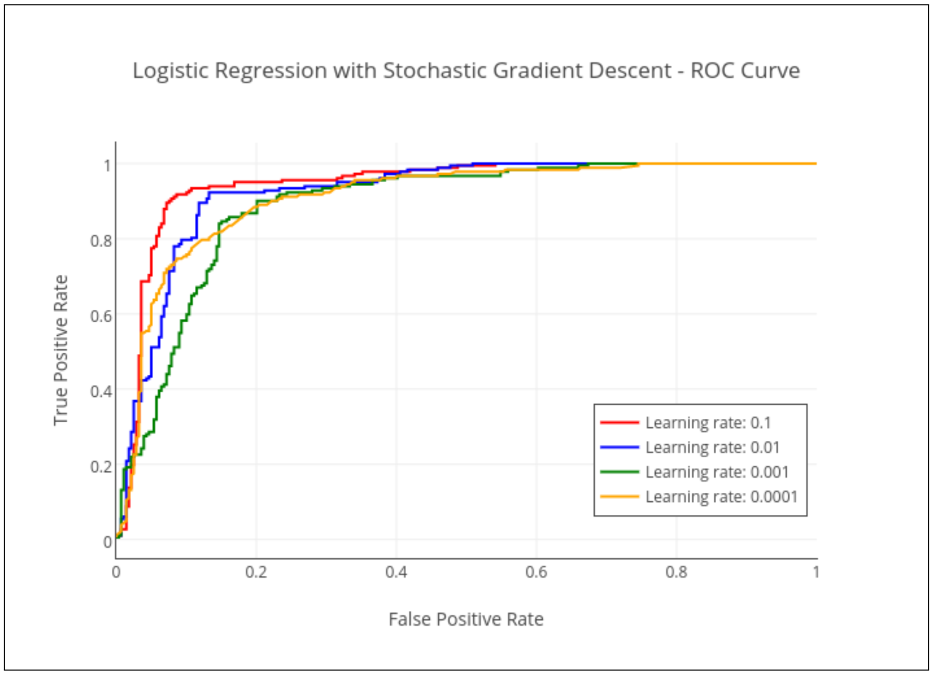

Figure 8: ROC Curve of Logistic Regression with SGD training with 4 different learning rates.

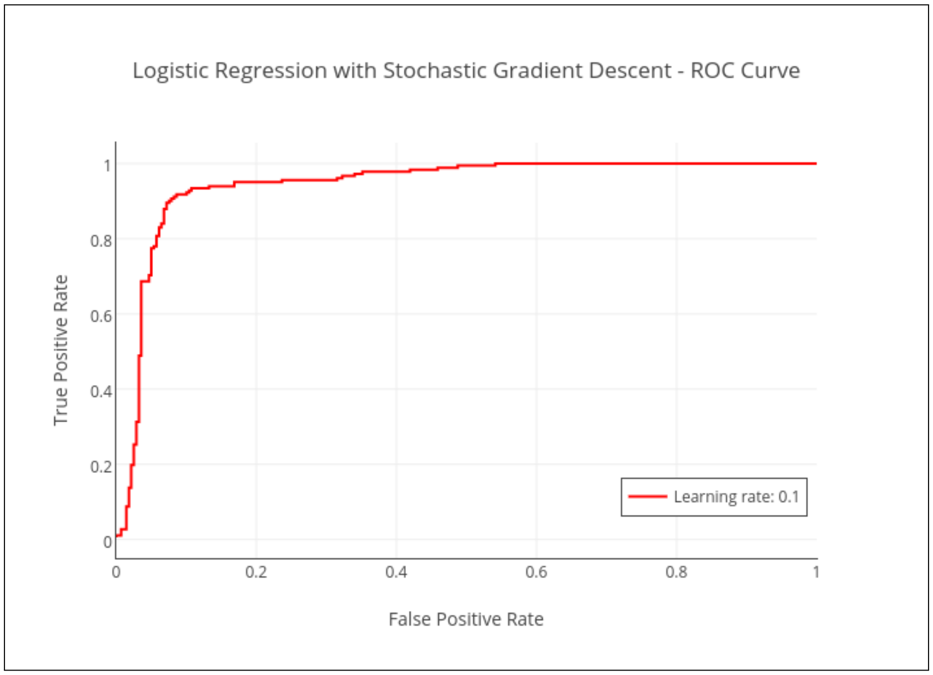

Figure 9: ROC Curve of Logistic Regression with SGD training with optimal learning rate identified.

Table 6: AUC of Logistic Regression with SGD trained with 4 different learning rates. Highest AUC shown in **bold**.

Epochs	Learning Rate	AUC
500	0.0001	0.8580093741384061
500	0.001	0.8896963251801965
500	0.01	0.923431407302375
500	**0.1**	**0.9437354759935405**

5 Logistic Regression with Batch Gradient Descent [1] [2]

5.1 Training

Table 7 shows the mean squared error (MSE) on the last epoch when training Logistic Regression with BGD for 50 epochs with 7 different learning rates. It was found that it produced the lowest MSE with a learning rate of 0.0001.

Figure 10 provides a visual representation of the performance of Logistic Regression with BGD trained with 4 different learning rates for 50 epochs.

5.2 Evaluation

Receiver operating characteristic (ROC) Curve

Figure 11 shows the ROC curve of Logistic Regression with BGD trained with 4 different learning rates for 50 epochs.

Figure 12 shows the ROC curve of Logistic Regression with BGD trained with the optimal learning rate identified for 50 epochs.

Table 7: Logistic Regression with BGD training with 7 different learning rates. Lowest MSE on the last epoch shown in **bold**.

Epochs	Regression	Gradient Descent	Learning Rate	Last Epoch MSE
50	Logistic	Batch	0.0000001	0.24895157834707887
50	Logistic	Batch	0.000001	0.2435508003326362
50	Logistic	Batch	0.00001	0.23698192269954932
50	**Logistic**	**Batch**	**0.0001**	**0.21890249704799133**
50	Logistic	Batch	0.001	diverged
50	Logistic	Batch	0.01	diverged
50	Logistic	Batch	0.1	diverged

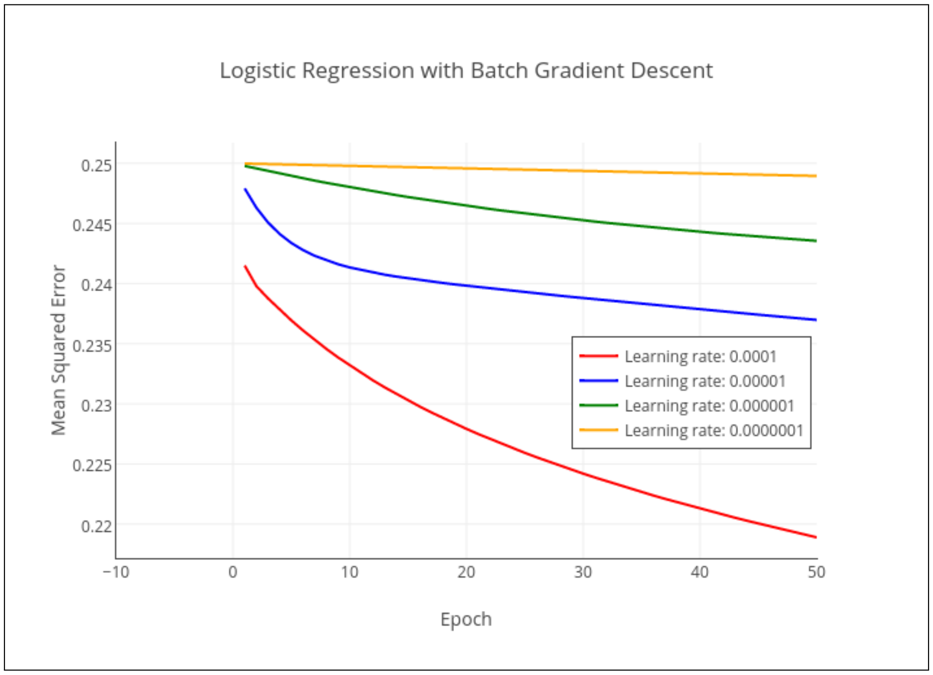

Figure 10: Logistic Regression with BGD training with 4 different learning rates.

Area under the curve (AUC)

Table 8 shows the AUC (Area under the curve) of Logistic Regression with BGD trained with 7 different learning rates for 50 epochs.

Table 8: AUC of Logistic Regression with BGD trained with 7 different learning rates. Highest AUC shown in **bold**.

Epochs	Learning Rate	AUC
50	0.0000001	0.5066761195793456
50	0.000001	0.528910157942416
50	0.00001	0.6403757532789796
50	**0.0001**	**0.7539879475363358**
50	0.001	diverged
50	0.01	diverged
50	0.1	diverged

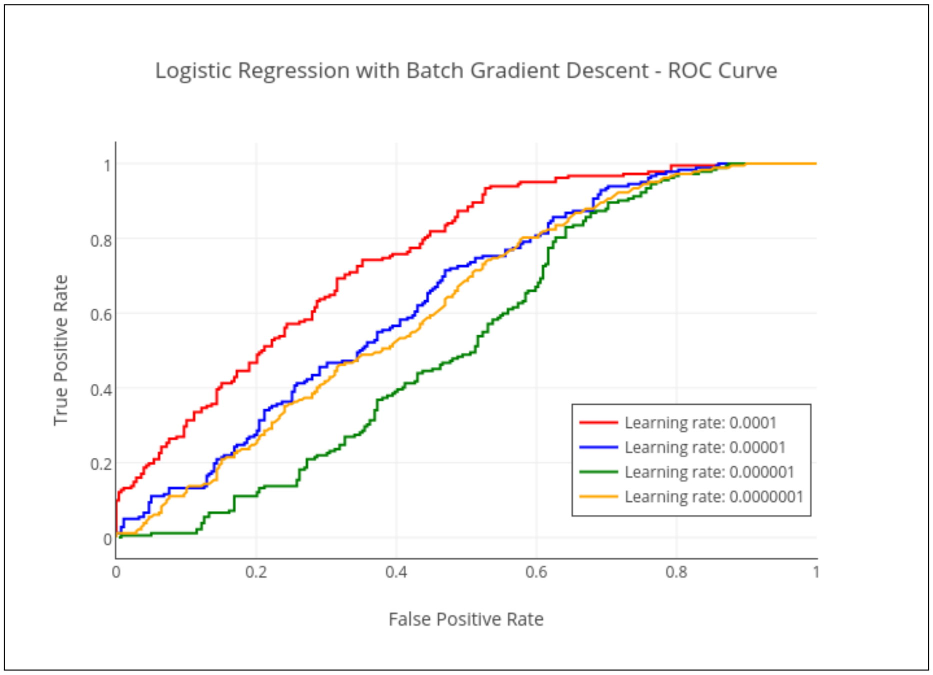

Figure 11: ROC Curve of Logistic Regression with BGD training with 4 different learning rates.

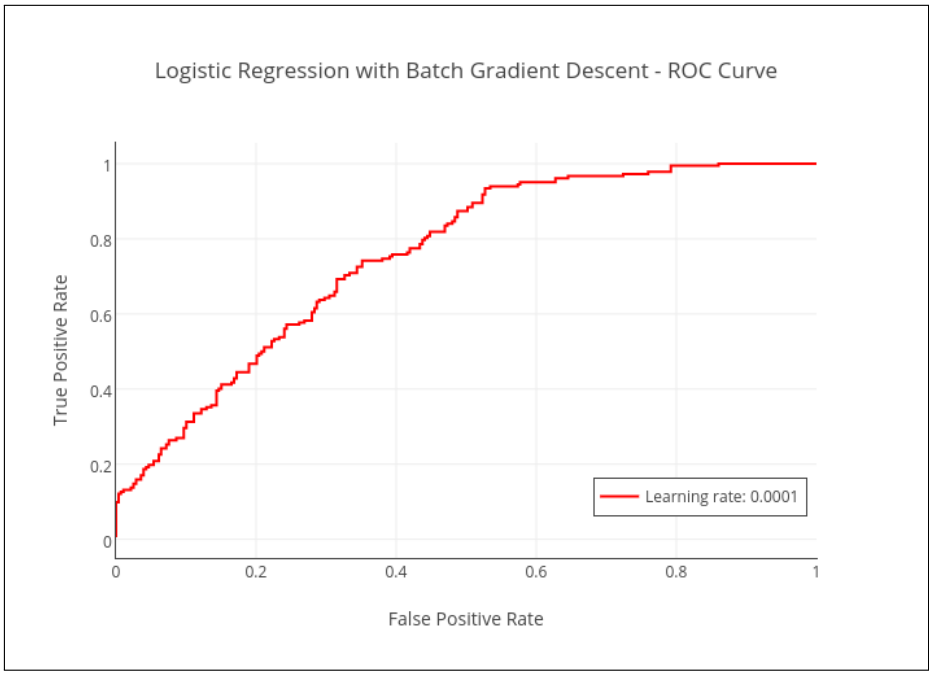

Figure 12: ROC Curve of Logistic Regression with BGD training with optimal learning rate identified.

6 Stochastic vs Batch Gradient Descent

Stochastic and Batch Gradient Descent bring advantages to different scenarios mainly dependant on the size of the data. Stochastic Gradient Descent is used when data size is big. In SGD, ”we repeatedly run through the training set, and each time we encounter a training example, we update the parameters according to the gradient of the error with respect to that single training example only.” [1] Batch Gradient Descent is used when data size is small. In BGD, we ”scan through the entire training set before taking a single step - a costly operation if m is large - stochastic gradient descent can start making progress right away, and continues to make progress with each example it looks at. Often, stochastic gradient descent gets θ ”close” to the minimum much faster than batch gradient descent.” [1]

Because SGD only considers one example at a time, it means that it completes a single epoch faster and therefore it can also learn quicker based on the previous epoch. In comparison, BGD considers the whole data set on each epoch, which means it takes significantly longer to complete one epoch. When SGD learns, it learns from the weights of one single example. When BGD learns, it learns from a sum of weights of all examples in the data set. This leads to a slower learning process as BGD doesn’t have a concrete example to learn from, instead it learns from a sum of examples.

6.1 Stochastic vs Batch Gradient Descent for Linear Regression

Linear Regression with Stochastic Gradient Descent was trained using 4 different learning rates for 500 epochs. The learning rates used were: 0.00001, 0.0001, 0.001 and 0.01, respectively. The Lowest MSE (0.17462479941829512) and highest AUC (0.8923549568710862) were produced using 0.001 as the learning rate. Both the MSE and AUC values produced are good.

I believe that for Linear Regression with SGD, 500 epochs is an optimal amount in order to obtain both a ”good” predictor and mean squared error.

On the other hand, Linear Regression with Batch Gradient Descent was trained using 6 different learning rates for 50 epochs. The learning rates used were: 0.0000001, 0.000001, 0.00001, 0.0001, 0.001, and 0.01, respectively. The Lowest MSE (0.22658756874338298) and highest AUC (0.7243491275749331) were produced using 0.000001 as the learning rate. Both the MSE and AUC values produced are fairly average. Based on this, I believe that BGD requires to run for more than 50 epochs. However, that will also increase the time it will take to run, considering that BGD is an already slower process than SGD.

In comparison, SGD has an undeniably faster and better learning process in comparison to BGD as the difference in MSE and AUC between the two is 0.051962769 and 0.168005829, respectively. However, I believe that if run for enough epochs, BGD for Linear Regression could get close to SGD’s performance.

As a result of this comparison, I attempted to run Linear Regression with Batch Gradient Descent for 2000 epochs. Tables 9 and 10 provide a numerical representation of this attempt, while Figures 13 and 14 provide a visual representation of it.

Table 9: Linear Regression with BGD trained with 0.000001 as the learning rate for 2000 epochs.

Epochs	**Regression**	**Gradient Descent**	**Learning Rate**	**Last Epoch MSE**
2000	Linear	Batch	0.000001	0.19789523905781295

Table 10: AUC of Linear Regression with BGD trained with 0.000001 as the learning rate for 2000 epochs.

Epochs	**Learning Rate**	**AUC**
2000	0.000001	0.8280751506557958

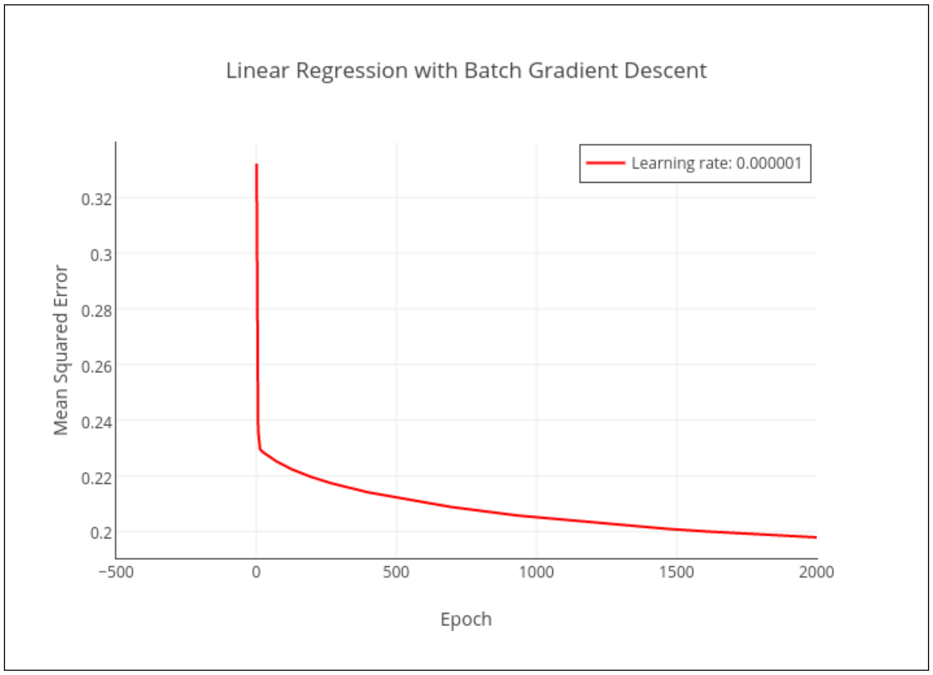

Figure 13: Linear Regression with BGD trained with 0.000001 as the learning rate for 2000 epochs.

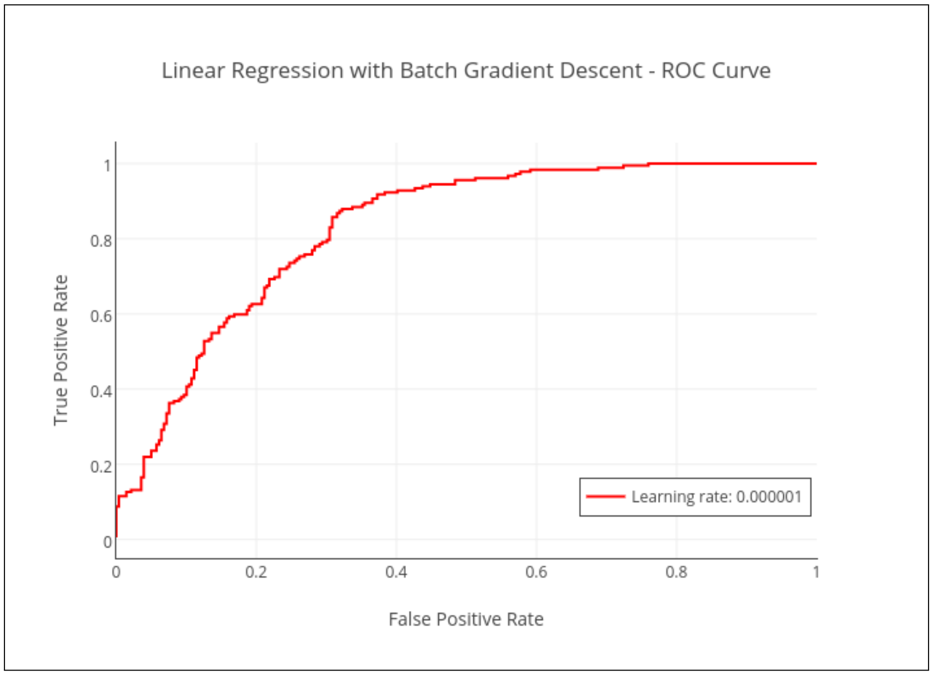

Figure 14: ROC Curve of Linear Regression with BGD trained with 0.000001 as the learning rate for 2000 epochs.

6.2 Stochastic vs Batch Gradient Descent for Logistic Regression

Logistic Regression with Stochastic Gradient Descent was trained using 4 different learning rates for 500 epochs. The learning rates used were: 0.0001, 0.001, 0.01 and 0.1, respectively. Lowest MSE (0.08741143950554693) and highest AUC (0.9437354759935405) were produced using 0.1 as the learning rate. Both the MSE and AUC values produced are good.

I believe that for Logistic Regression with SGD, 500 epochs is an optimal amount in order to obtain both a ”good” predictor and mean squared error.

On the other hand, Logistic Regression with Batch Gradient Descent was trained using 7 different learning rates for 50 epochs. The learning rates used were: 0.0000001, 0.000001, 0.00001, 0.0001, 0.001, 0.01 and 0.1, respectively. Lowest MSE (0.21890249704799133) and highest AUC (0.7539879475363358) were produced using 0.0001 as the learning rate. Both the MSE and AUC values produced are fairly average. Based on this, I believe that BGD requires to run for more than 50 epochs. However, that will also increase the time it will take to run, considering that BGD is an already slower process than SGD.

In comparison, SGD has an undeniably faster and better learning process as the difference in MSE and AUC between the two is 0.131491058 and 0.189747528, respectively. However, I believe that if run for enough epochs, BGD for Logistic Regression could get close to SGD's performance.

As a result of this comparison, I attempted to run Logistic Regression with Batch Gradient Descent for 2000 epochs. Tables 11 and 12 provide a numerical representation of this attempt, while Figures 15 and 16 (next page provide a visual representation of it.

Table 11: Logistic Regression with BGD trained with 0.0001 as the learning rate for 2000 epochs.

Epochs	Regression	Gradient Descent	Learning Rate	Last Epoch MSE
2000	Logistic	Batch	0.0001	0.16774604152828332

Table 12: AUC of Logistic Regression with BGD trained with 0.0001 as the learning rate for 2000 epochs.

Epochs	Learning Rate	AUC
2000	0.0001	0.8764425538619093

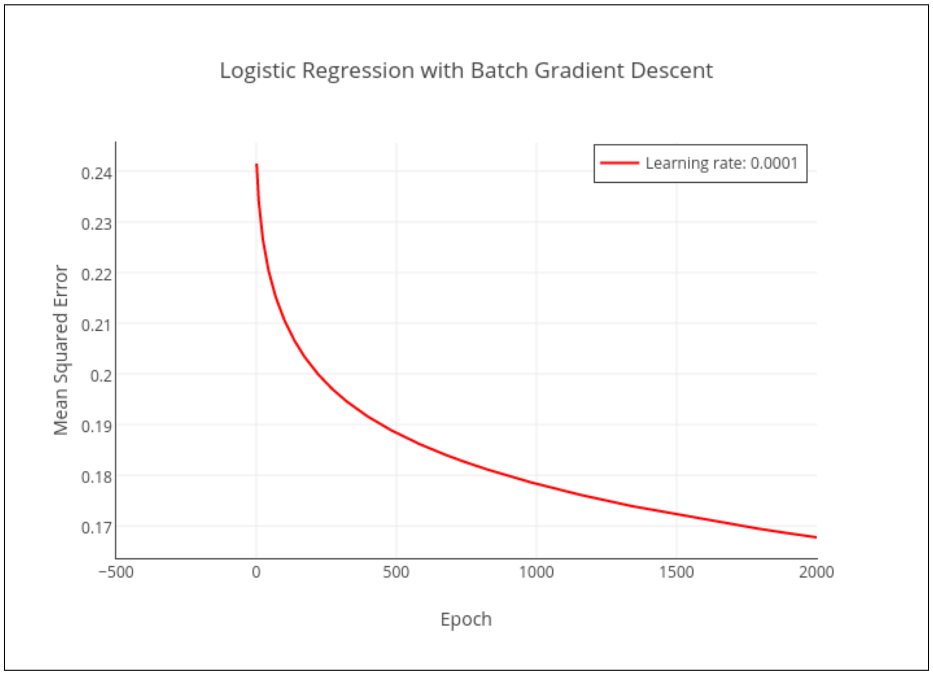

Figure 15: Logistic Regression with BGD trained with 0.0001 as the learning rate for 2000 epochs.

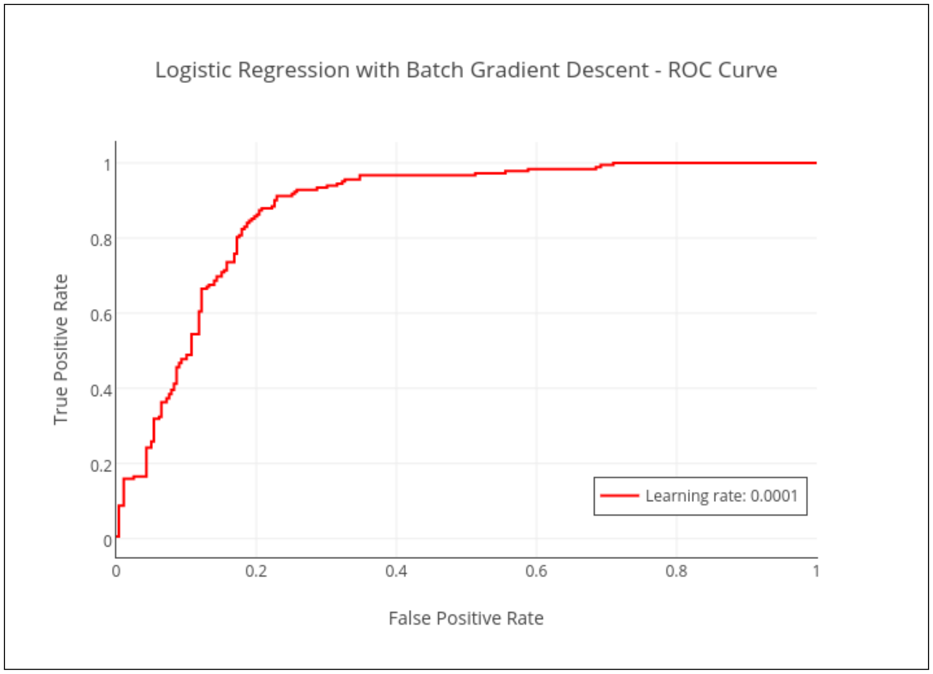

Figure 16: ROC Curve of Logistic Regression with BGD trained with 0.0001 as the learning rate for 2000 epochs.

References

[1] Andrew Ng. "CS229 Lecture notes". In: *CS229 Lecture notes* 1.1 (2000), pp. 1–30.

[2] The Yhat Blog. *ROC Curves in Python and R*. URL: `http://blog.yhat.com/posts/roc-curves.html` (visited on 12/03/2016).

[3] Python Docs. *Built-in Functions: List Comprehensions*. URL: `https://docs.python.org/3.5/tutorial/datastructures.html#list-comprehensions` (visited on 01/03/2016).

[4] Python Docs. *Built-in Functions: zip()*. URL: `https://docs.python.org/3.5/library/functions.html#zip` (visited on 03/03/2016).

[5] Python Docs. *Built-in Functions: sum()*. URL: `https://docs.python.org/3.5/library/functions.html#sum` (visited on 05/03/2016).

[6] Plotly. *Visualize Data, Together*. URL: `https://plot.ly/` (visited on 12/03/2016).