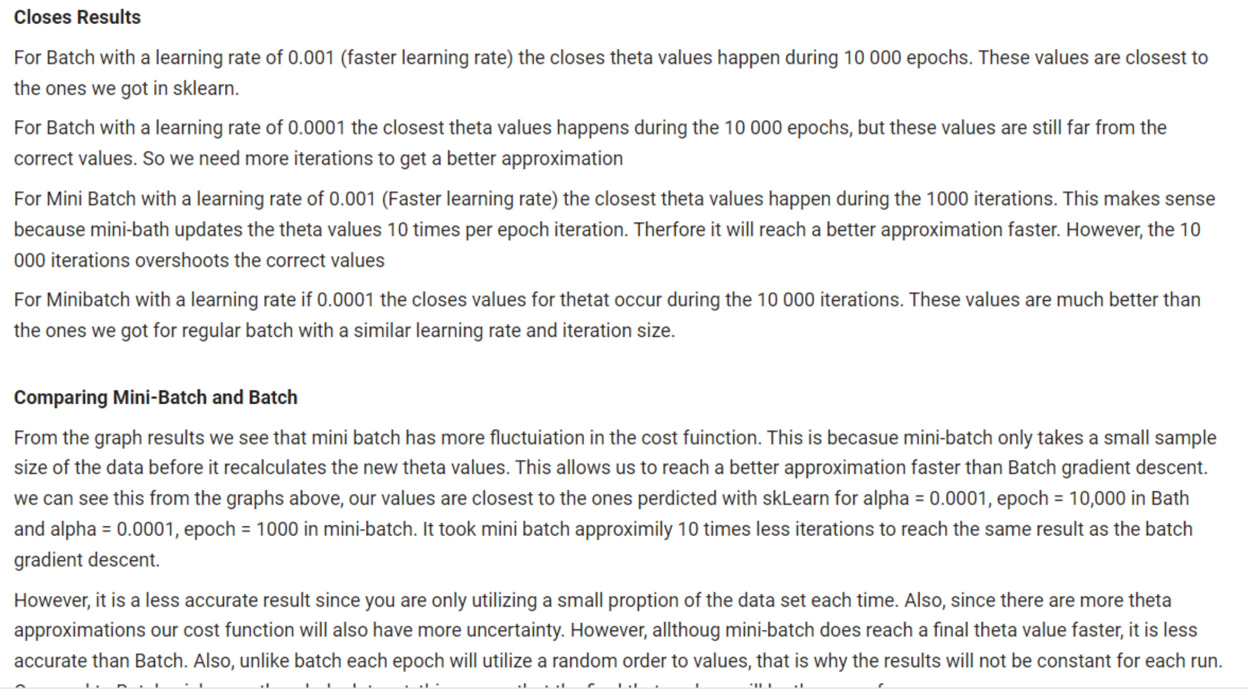
Comparing the Results of Different Batch sizes

Objective:

The objective of this experiment was to analyse the learning rate of logistic regression coupled with a sigmoid function. To do this, we chose three different step sizes (0.001, 0.0001 and 0.00001) along with three different iteration sizes (500, 1000 and 10 000). Initially we tried every combination of step size with iteration size using a total/full batch approach. Next, we tried each possible iteration with a mini batch approach. Unlike a full batch mini batch updates the variables multiple times in one epoch, this allows the system to reach a better approximation faster, however it is less accurate. The results we obtained from this analysis here coherent with our expected results.



Closes Results

For Batch with a learning rate of 0.001 (faster learning rate) the closes theta values happen during 10 000 epochs. These values are closest to the ones we got in sklearn.

For Batch with a learning rate of 0.0001 the closest theta values happens during the 10 000 epochs, but these values are still far from the correct values. So we need more iterations to get a better approximation

For Mini Batch with a learning rate of 0.001 (Faster learning rate) the closest theta values happen during the 1000 iterations. This makes sense because mini-bath updates the theta values 10 times per epoch iteration. Therfore it will reach a better approximation faster. However, the 10 000 iterations overshoots the correct values

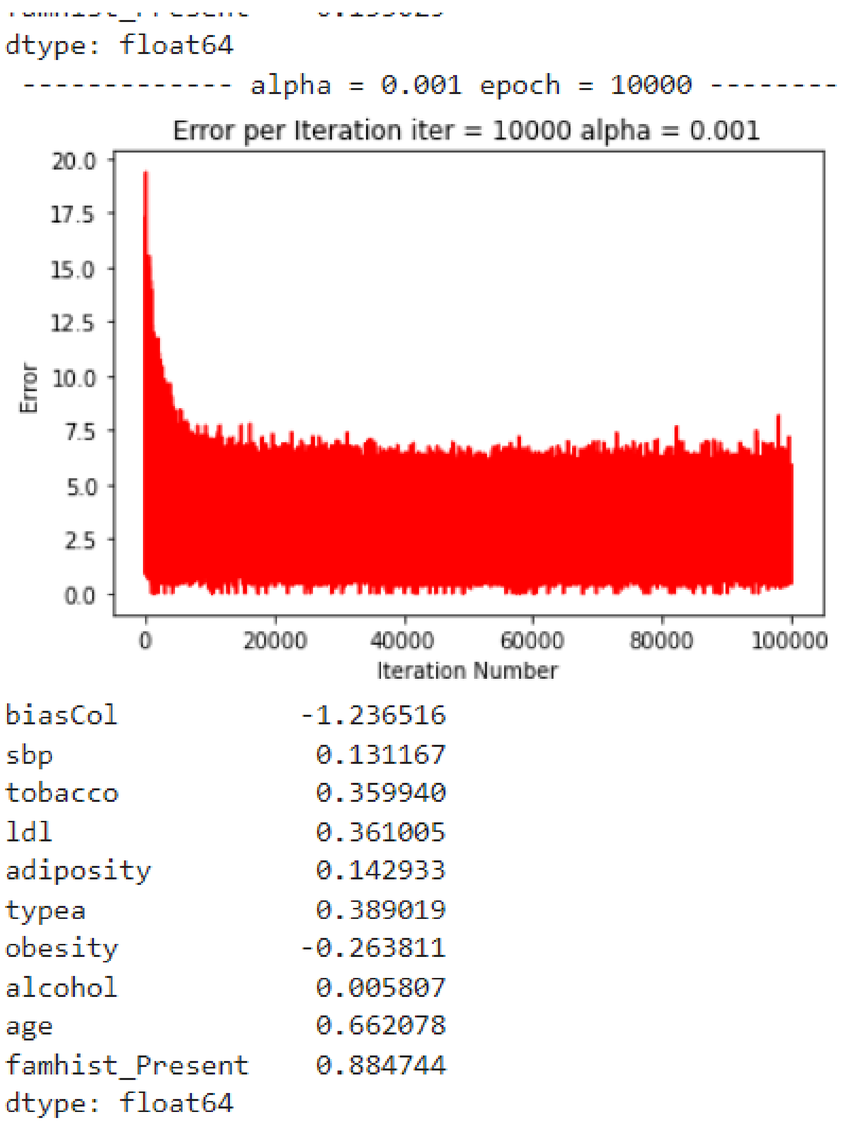
For Minibatch with a learning rate if 0.0001 the closes values for thetat occur during the 10 000 iterations. These values are much better than the ones we got for regular batch with a similar learning rate and iteration size.

Comparing Mini-Batch and Batch

From the graph results we see that mini batch has more fluctuiation in the cost fuinction. This is becasue mini-batch only takes a small sample size of the data before it recalculates the new theta values. This allows us to reach a better approximation faster than Batch gradient descent. we can see this from the graphs above, our values are closest to the ones perdicted with skLearn for alpha = 0.0001, epoch = 10,000 in Bath and alpha = 0.0001, epoch = 1000 in mini-batch. It took mini batch approximily 10 times less iterations to reach the same result as the batch gradient descent.

However, it is a less accurate result since you are only utilizing a small proption of the data set each time. Also, since there are more theta approximations our cost function will also have more uncertainty. However, allthoug mini-batch does reach a final theta value faster, it is less accurate than Batch. Also, unlike batch each epoch will utilize a random order to values, that is why the results will not be constant for each run.

Sample result (mini batch approach)



```
famhist_Present    [illegible]
dtype: float64
 ------------- alpha = 0.001 epoch = 10000 --------
```

```
biasCol           -1.236516
sbp                0.131167
tobacco            0.359940
ldl                0.361005
adiposity          0.142933
typea              0.389019
obesity           -0.263811
alcohol            0.005807
age                0.662078
famhist_Present    0.884744
dtype: float64
```