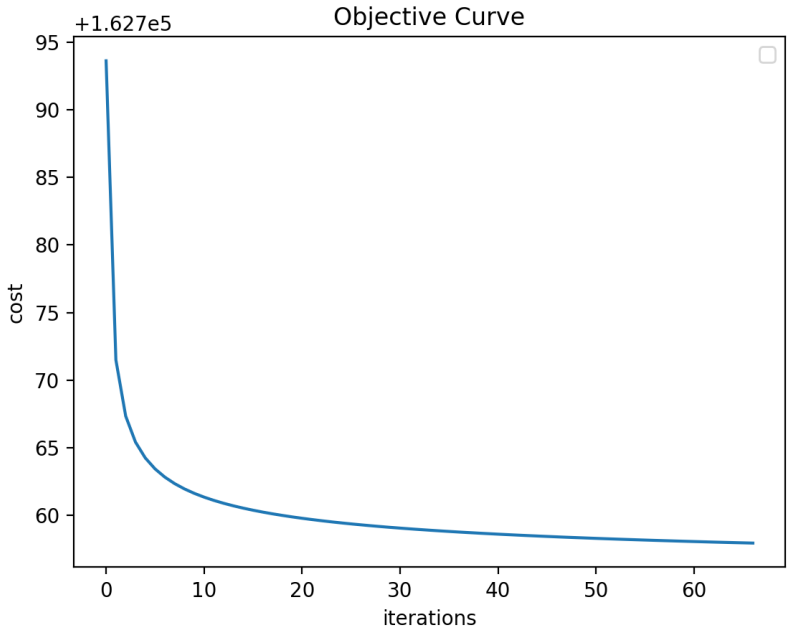
# ML Project 1 Experiments

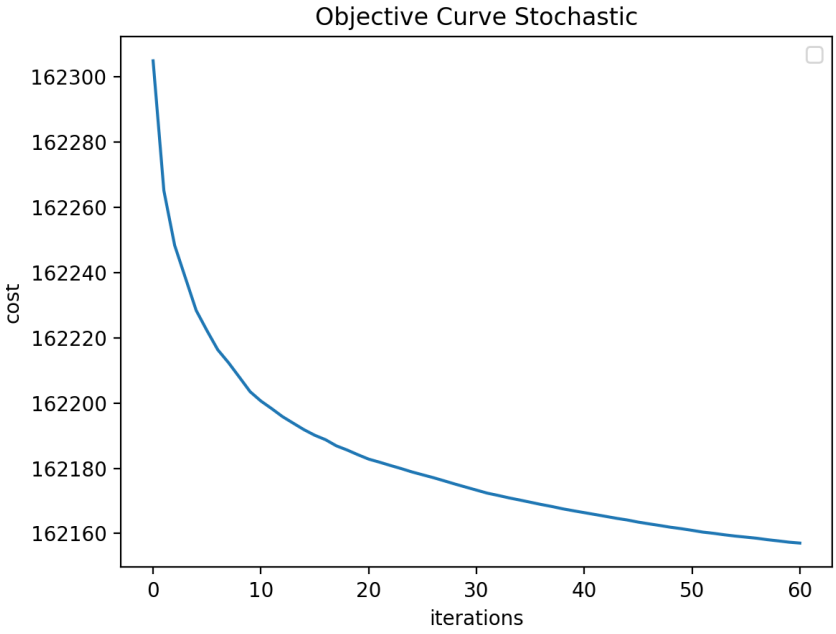
Thomas Klimek

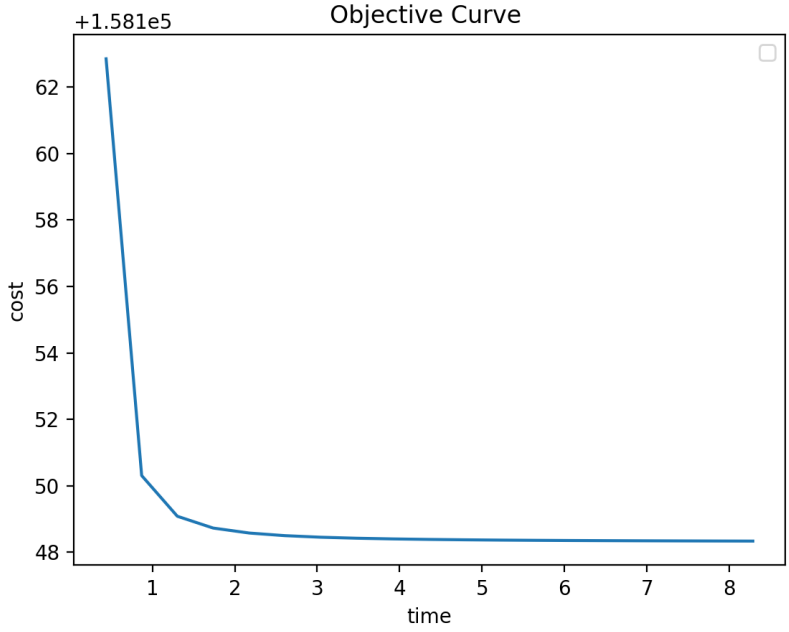
October 2018

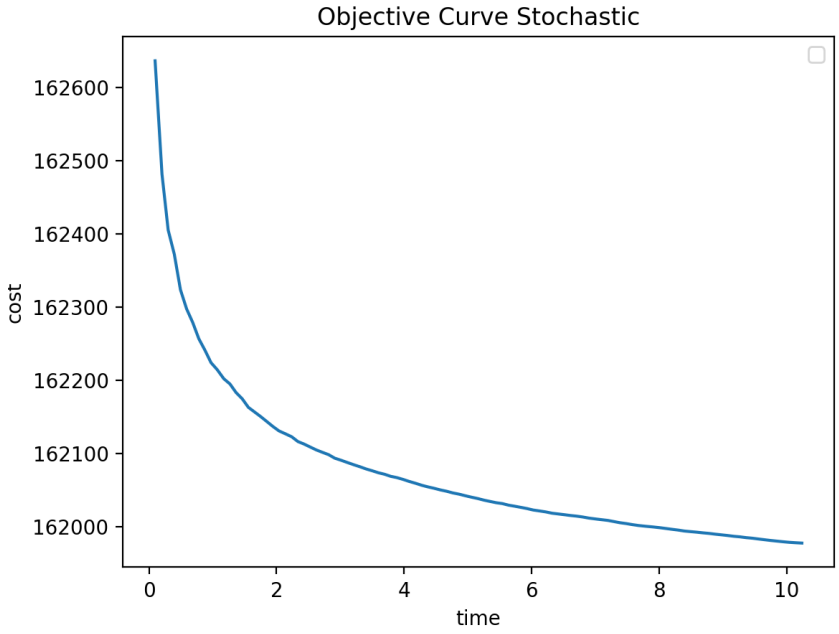
## 1 4.1 Full Gradient v.s. Stochastic

When plotting regular gradient descent v.s. stochastic gradient descent I observed the following curves for cost function v.s. iteration and cost function v.s. time.




Objective Curve Stochastic
cost
iterations
162300
162280
162260
162240
162220
162200
162180
162160
0
10
20
30
40
50
60

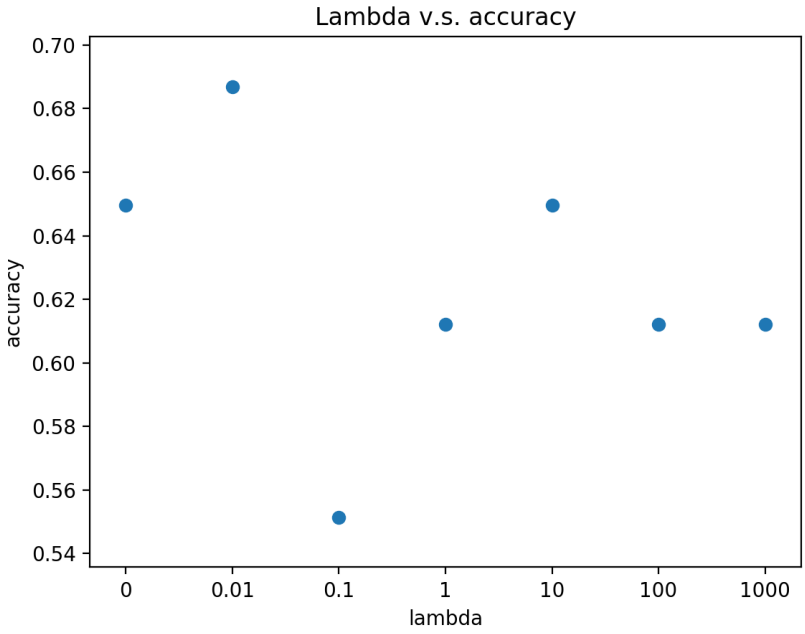

Objective Curve
+1.581e5
cost
time
62
60
58
56
54
52
50
48
1
2
3
4
5
6
7
8




You can see that the the objective curve converges faster for the normal gradient descent, however the stochastic gradient descent will overall have less time complexity because it is not necessary to calculate the gradient on the entirety of the training data, but rather just a random batch. Also it is expected for the stochastic gradient descent to have more noise and variation in the cost function over time, because the batch is randomly sampled and thus will produce different costs. This is not observed in my graph perhaps because of an implementation error.

## 2 4.2 Effects of Regularizer

Plotting different test accuracies v.s. values of the regularizer $\lambda$, I generated the following plot.




We can see that the highest value was achieved for $\lambda = 0.1$. I suspect there are some implementation errors within my code, so the chart is not entirely precise, however the sense is that a lambda of zero will overfit the data to the training data, while a very high lambda will make the regularizer dominate the weights and underfit the data. This trend is still partially observed in my graph.