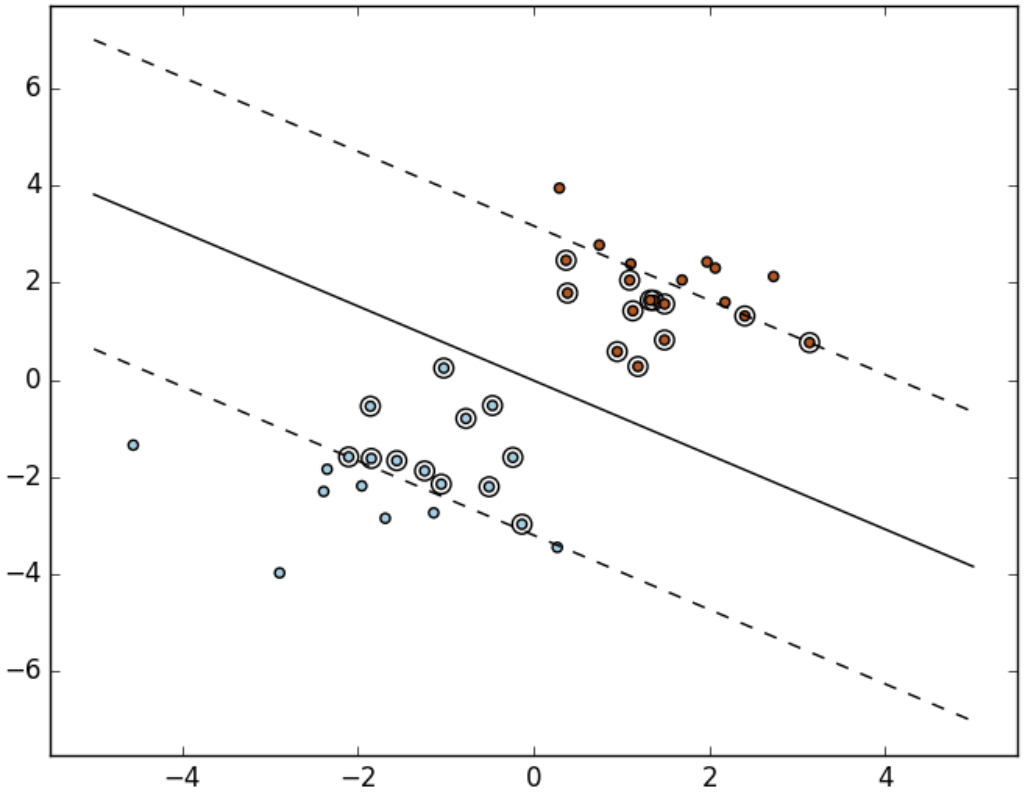
SENG 474 Assignment 3

Michael Reiter
V00831568

1. a) The plot made using C = 1 is identical to the one made using C = 10; they both have exactly 3 support vectors. This is because C is already large enough so there will be no slack variables. Increasing C makes the SVM choose a hyperplane with a smaller margin if that hyperplane better classifies the training points. In other words, increasing C changes how tightly fit the model is.

   b) With C = 0.01, there are exactly 2 support vectors. Support vectors are the points which define the hyperplane, so there will be only 2 unless there are more points that lie exactly on the hyperplane (as in the example for C = 1 and C = 10).

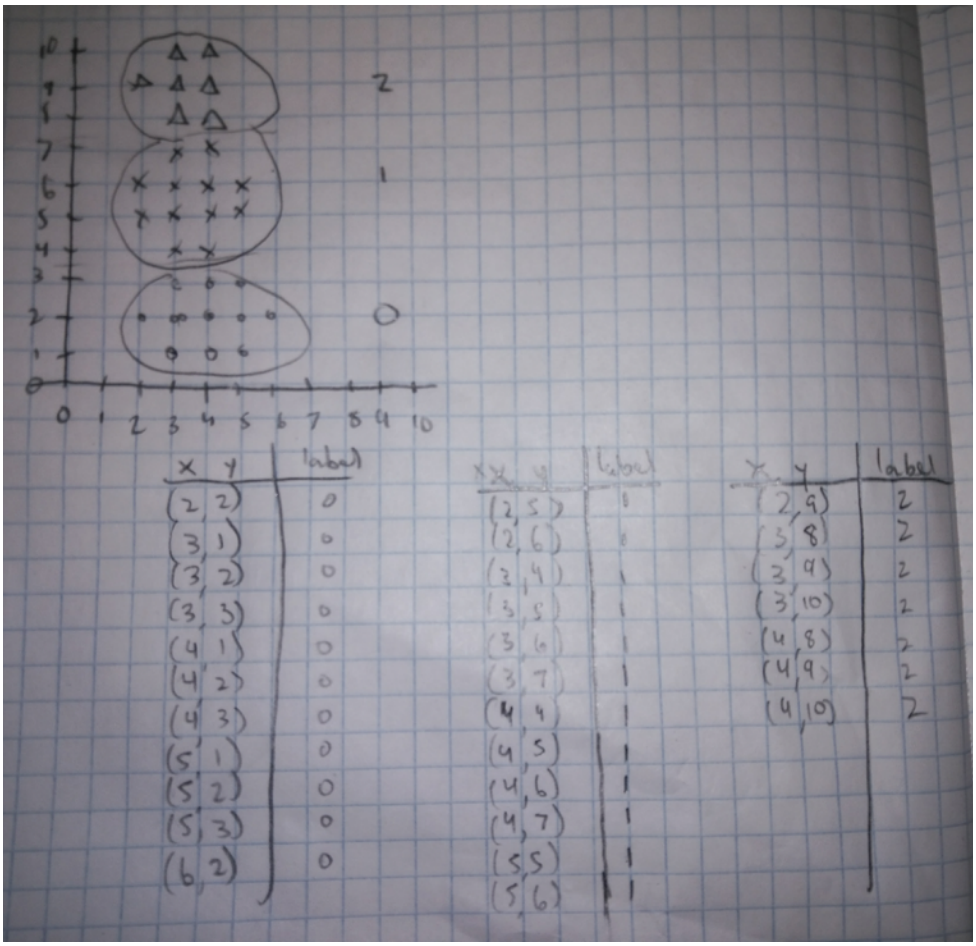
   c)



SENG 474 Assignment 3

Michael Reiter
V00831568

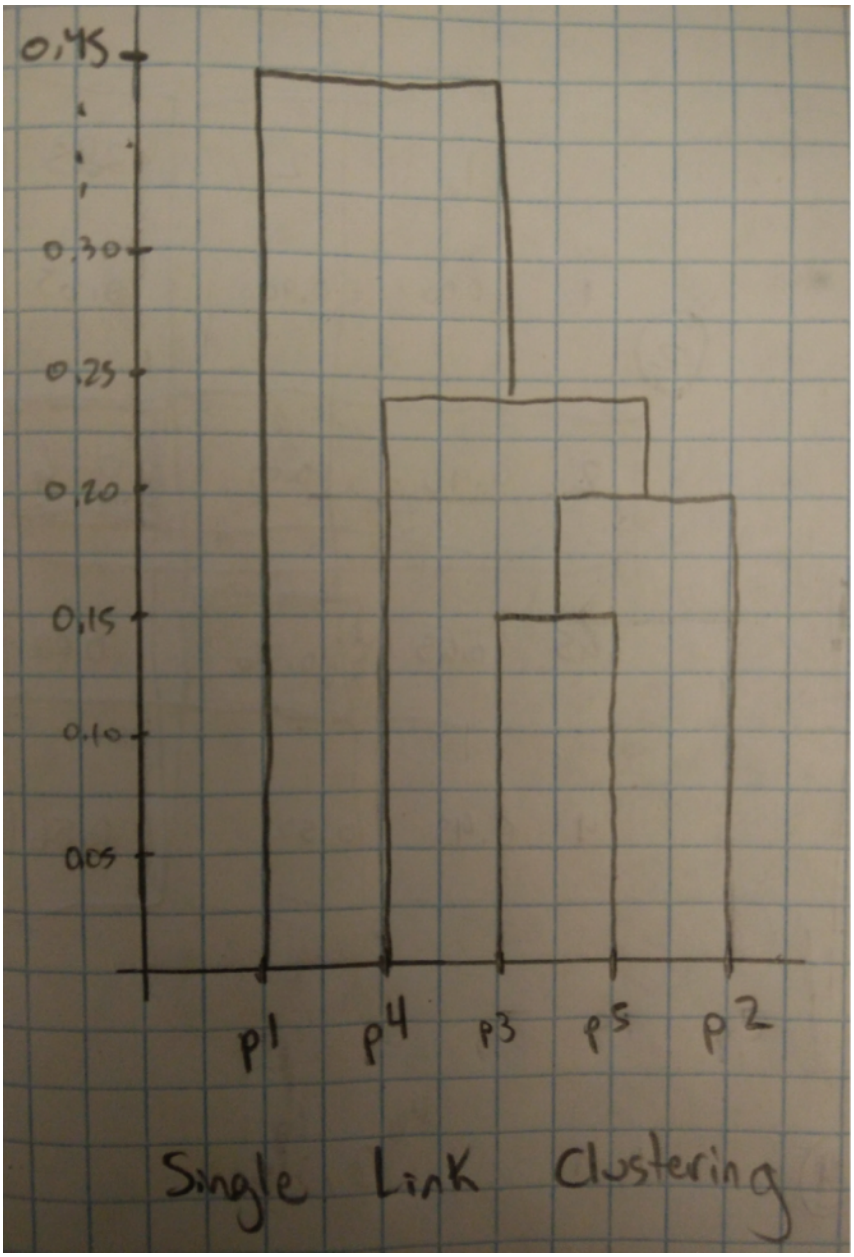
2.



SENG 474 Assignment 3

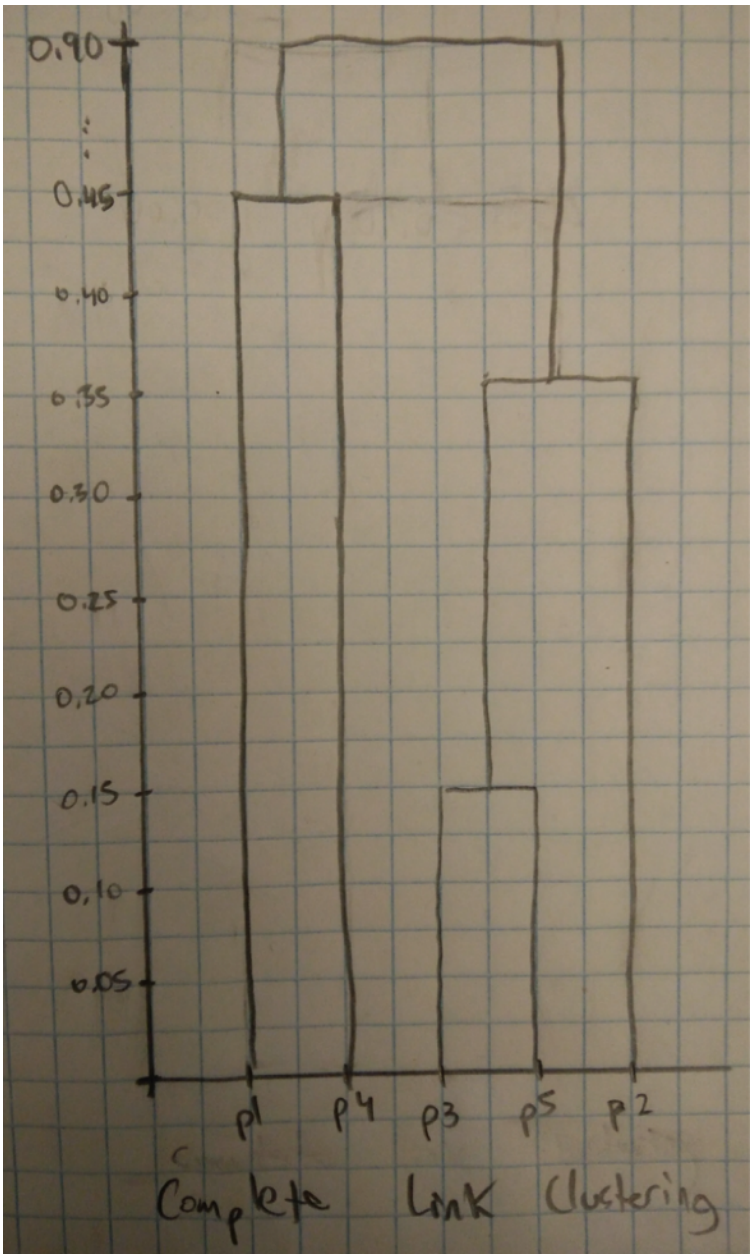
Michael Reiter
V00831568

3.



SENG 474 Assignment 3

Michael Reiter
V00831568



4. See also included code (recommender.py)

```
   *******User User similarity*******
Test instance 0 Recommend joke: 42
Test instance 1 Recommend joke: 75
Test instance 2 Recommend joke: 96
Test instance 3 Recommend joke: 62
Test instance 4 Recommend joke: 42
RMSE for all predictions: 10.5629878586
```

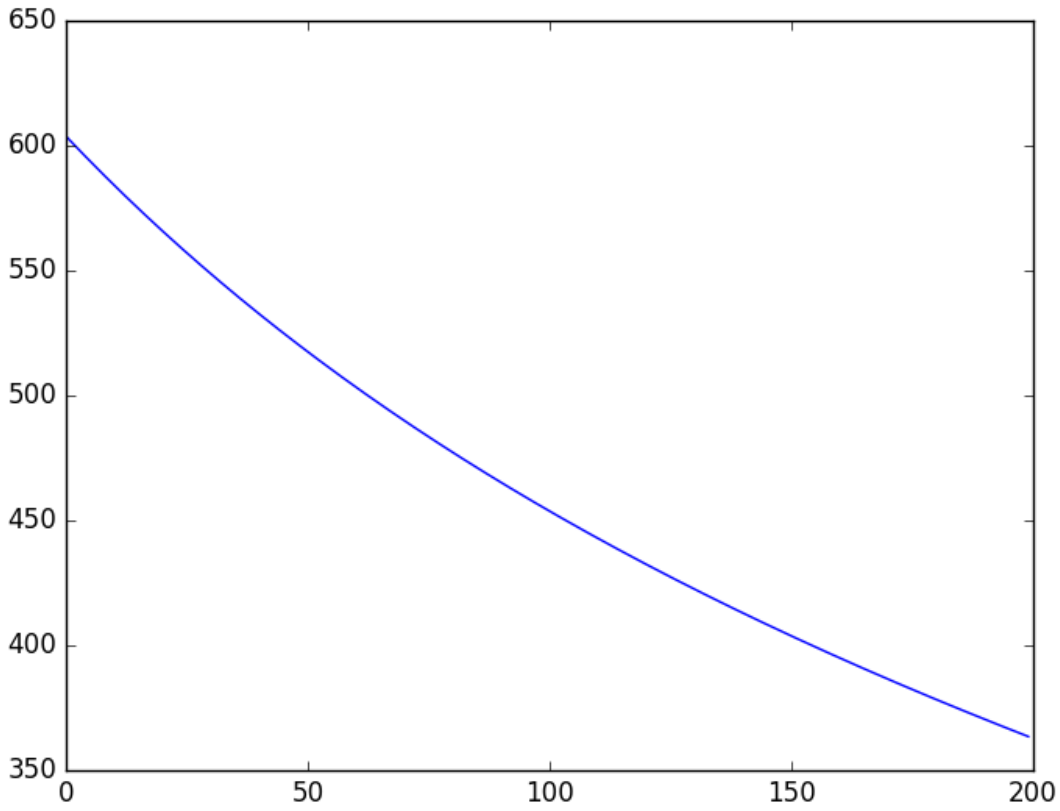
Michael Reiter
V00831568

```
    *******Item Item similarity*******
Test instance 0 Recommend joke: 97
Test instance 1 Recommend joke: 93
Test instance 2 Recommend joke: 88
Test instance 3 Recommend joke: 97
Test instance 4 Recommend joke: 85
RMSE for all predictions: 4.28008523232
```

5. See also included code (linreg.py)

   a) For Batch Gradient Descent, at each iteration the weights vector w is updated based on the gradient of the error function taking into account all n rows of the dataset X. It can be seen clearly that increasing the learning rate K causes the cost function to converge more quickly (falling more sharply).

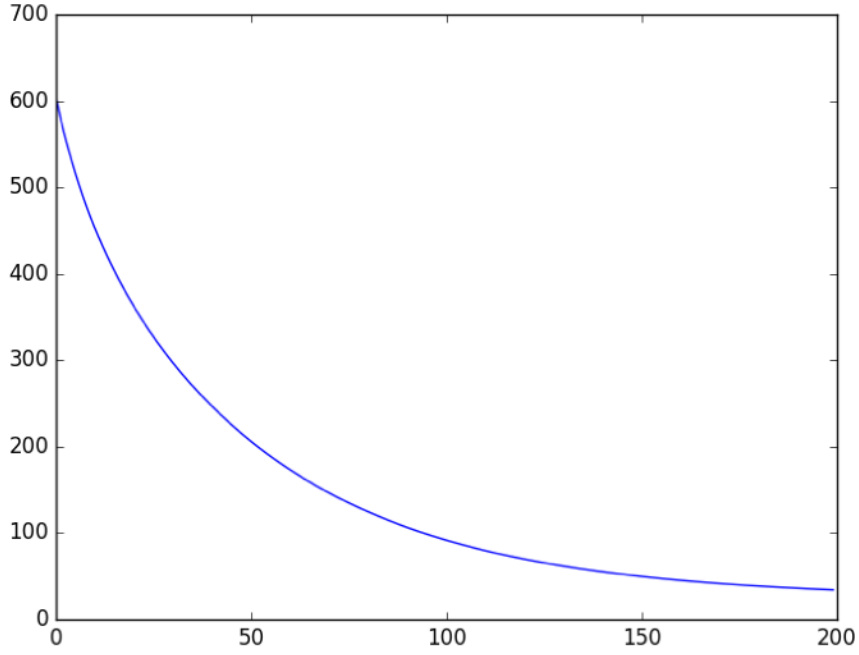
   K = 0.001



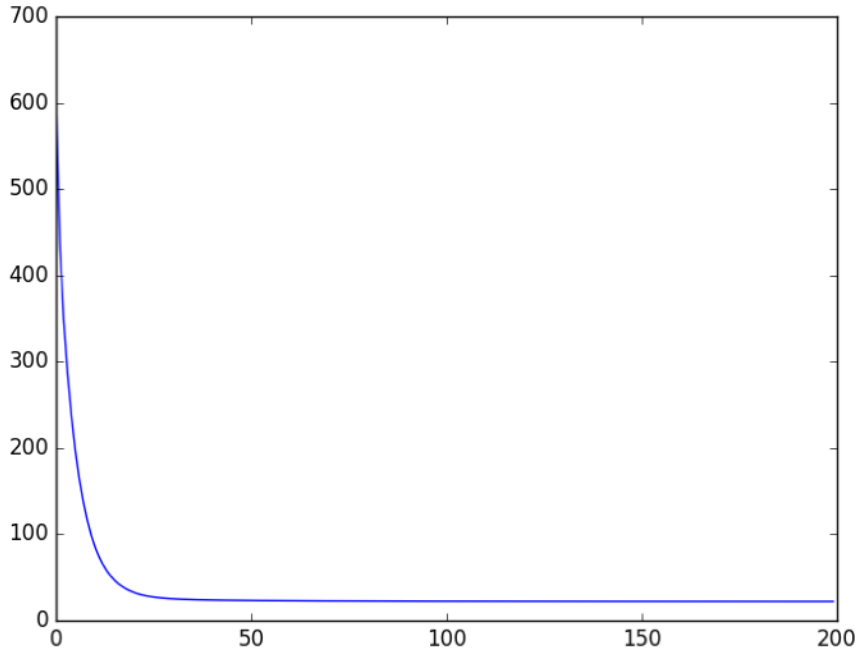
SENG 474 Assignment 3

Michael Reiter
V00831568

K = 0.01



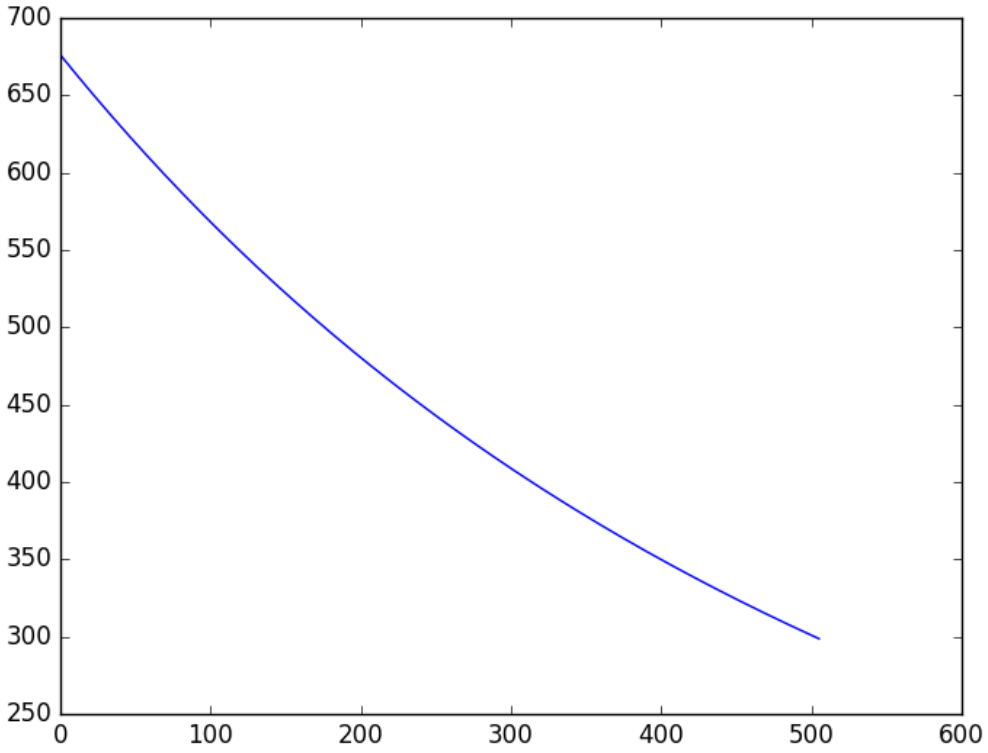
K = 0.1



Michael Reiter
V00831568

b) For Stochastic Gradient Descent, rather than taking into account all rows of the dataset when updating the weights vector (as in Batch Gradient Descent), only one row is considered at a time. One pass through of the each row in the data set is called an epoch. Like in part a, increasing the learning rate causes it to converge faster. Compared to Batch Gradient Descent, the cost function is initially larger (~675 vs ~600), but it converges to a smaller minimum (~25 vs ~33). It appears Stochastic Gradient Descent converged slower than Batch Gradient descent in my testing.

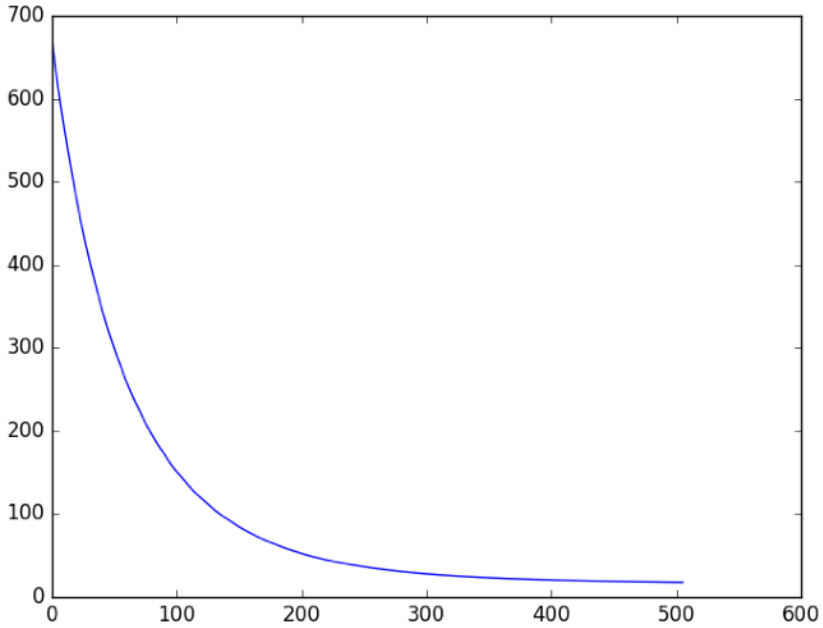
K = 0.001



SENG 474 Assignment 3

Michael Reiter
V00831568

K = 0.01



K = 0.1