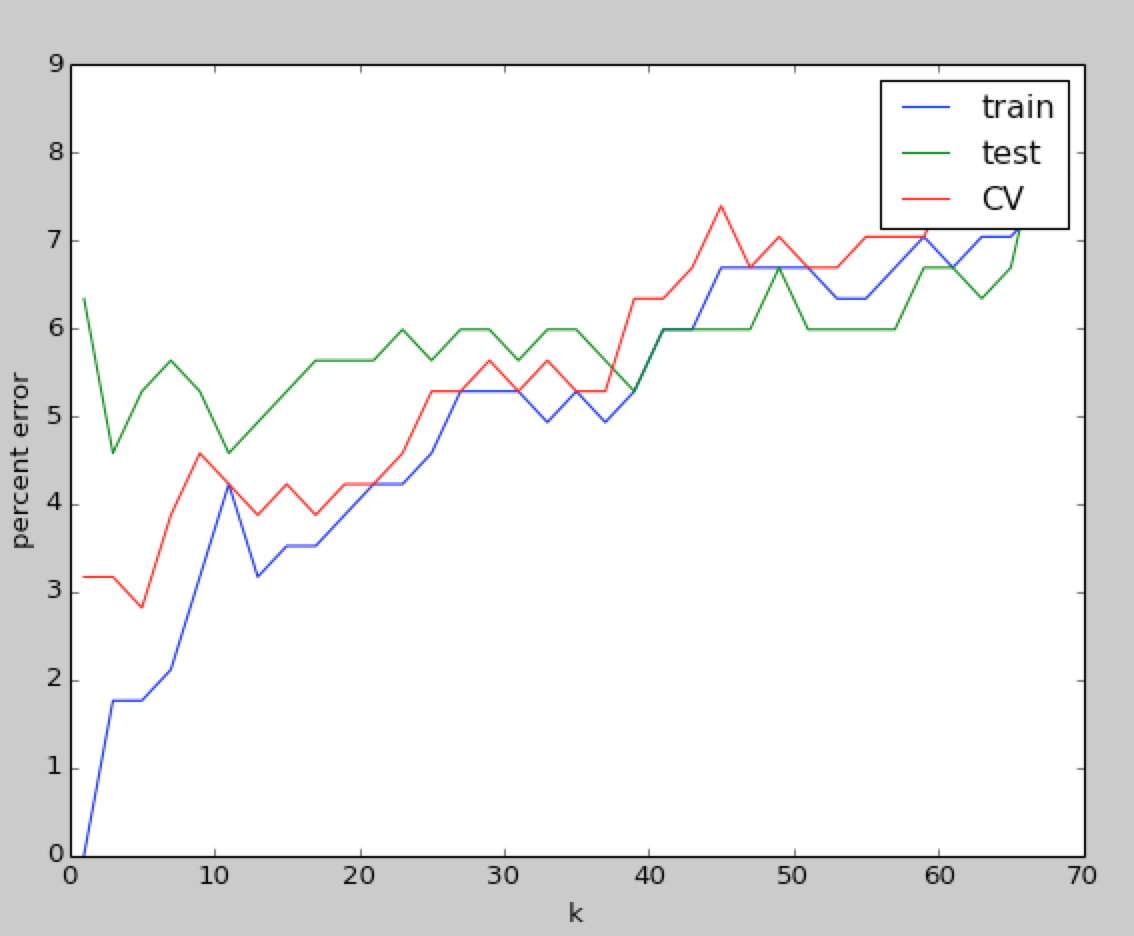
Alex Nguyen

Michael Lee

CS434

Assignment 2 - Logistic regression with L2 regularization

1. Part 1 – KNN
   1. Implemented algorithm – done.
   2. Graph of K vs. Error of Training, Test, Cross-validation error
   3. We observe that as K get larger that the error rate rises. Our training data is perfect when compared to itself and with a K of 1. The error rate rises pretty rapidly there after. Our test data's error rate stays approximately 6 percent error for the first 70 values of K. Our leave-one-out cross-validation error rate has a somewhat linear climb similar to the training set. Looking at the graph, a acceptable K value would be approximately 35 values of K.
2. Decision Trees
   1. Decision stump:
      1. Feature: 12
      2. Train Correct Percentage: 64.4366197183
      3. Test Correct Percentage: 60.9154929577
   2. Decision tree with depth 6:
      1. Tree with info gain per depth level is in tree.txt file
      2. Train correct percentage: 64.4366197183
      3. Test correct percentage: 60.9154929577
      4. Compared to the decision stump rates, the decision tree performs the same. These error rates could probably be caused by over fitting to the training data at depth six. From our results, the amount of -1 flags overruns both datasets after the split at depth one. This causes our data to predict very similar to each other.
      5. Our biggest difficulty was having the data split at certain thresholds. We followed the slide on sorting the attribute data and calculating entropy whenever the class changes. However, we had some bugs of wrong entropy values from bad data splits.
3. Using decision tree rates to improve KNN
   1. By knowing which feature splits the data, we can use that feature to improve KNN algorithm to increase the division of neighbors in neighborhoods. With this, we can run KNN on the smaller datasets and improve the classification from the decision tree alone. By improving neighborhoods we can achieve a higher prediction.