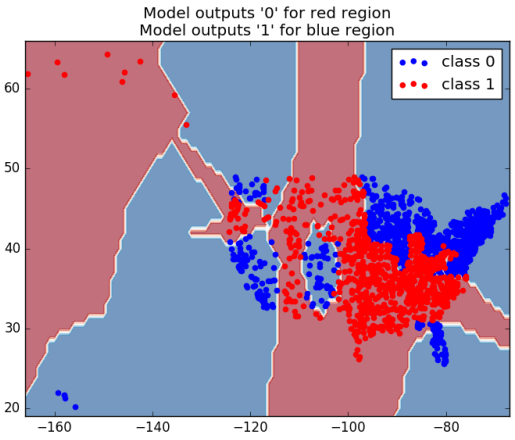
CPSC 340 Assignment 2

K-Nearest Neighbours, Random Forests, K-Means, Density-Based Clustering

1 K-Nearest Neighbours

1.1 KNN Prediction

1. See predict function in knn.py. `https://github.ubc.ca/cpsc340/jeanlam_nafis1_hw2/blob/master/code/knn.py`
2. The training and test errors:
   For K=1:Training error:0 Testing error:0.0645
   For K=3:Training error:0.0275 Testing error:0.066
   For K=10:Training error:0.072 Testing error:0.097

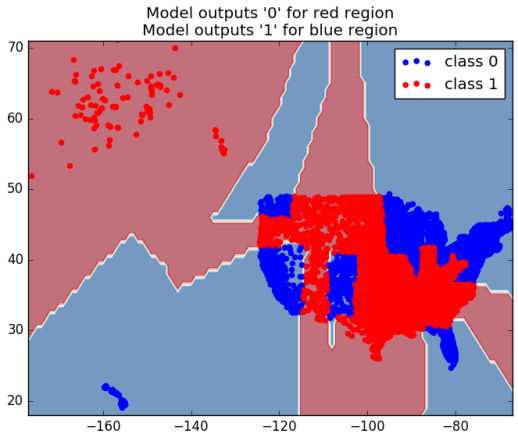


3.

4. For k=1 the training error is 0 because since k =1 it is 1-nearest neighbour. When we our function is predicting we are just taking the labels from the training data.The training examples are 1-nearest neighbour of itself.
5. We can use cross-validation like 5-fold or 10-fold to choose k.

1.2 Condensed Nearest Neighbours

1. See cnn.py. `https://github.ubc.ca/cpsc340/jeanlam_nafis1_hw2/blob/master/code/cnn.py`
2. Training error: 0.0075, Testing error:0.0175, Number of variables:455
3. See figures.



4. The examples that were correctly classified are no longer being stored as we took a subset of the training examples. As we add more examples the previous examples can become incorrectly classified
5. Now we are computing only the s distances to t test points.This causes us to have a runtime of O(dst).O(s) for s distances,O(t) for t points and O(d).
6. One reason for the test error being that high is some states might not be present in the training data.The training error is high because the order of the example affects the CNN method

2 Random Forests

2.1 Random Trees

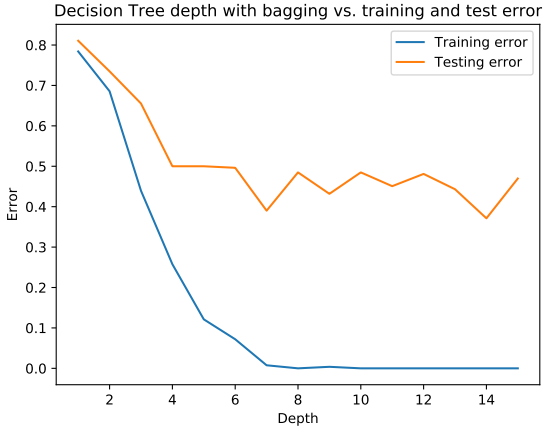
1.

2. The decision tree function terminates when the max depth is set to infinity as there are no more variables to split on. The fit function returns when either the max-depth is 1, or there is no split variable in the model.

3.

The random stump model can be found at `https://github.ubc.ca/cpsc340/jeanlam_nafis1_hw2/blob/master/code/random_stump.py`

4.



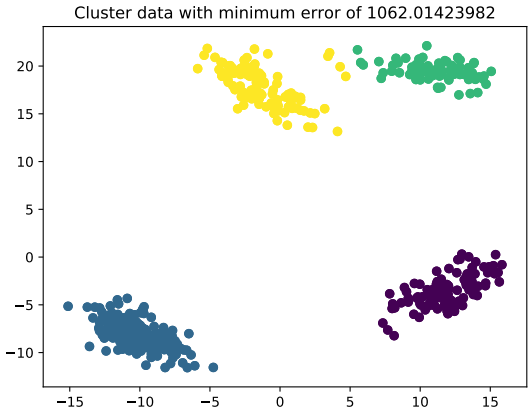
2.2 Random Decision Forests

1. The test error for 50 trees and infinite depth is 0.367 and the training error is 0.0
2. The test error for 50 trees and infinite depth with bootstrapping is 0.610 and the training error is 0.473
3. The test error for 50 random trees with infinite depth is 0.178 and the training error is 0.0
4. The test error for 50 random trees with infinite depth and bootstrapping is 0.568 and the training error is 0.458
   The random forest model can be found at `https://github.ubc.ca/cpsc340/jeanlam_nafis1_hw2/blob/master/code/random_forest.py`
5. From the above test errors, it is evident that using a random tree without bootstrapping works better than a decision tree without bootstrapping.

3 K-Means Clustering

3.1 Selecting among Initializations

1. The error function can be found at `https://github.ubc.ca/cpsc340/jeanlam_nafis1_hw2/blob/master/code/kmeans.py`

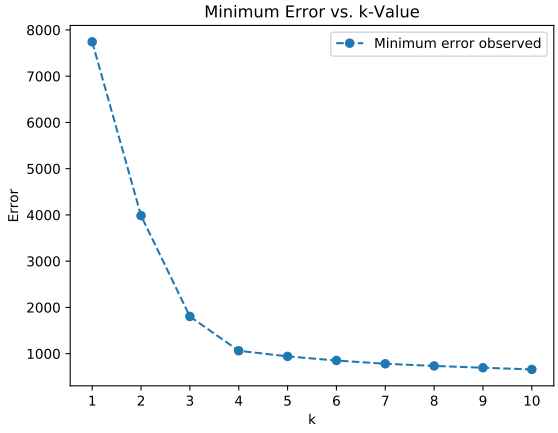


2.

The minimum error obtained is 1062.014

3.2 Selecting k

1. We cannot choose k by minimizing the sum of squares distance as we have many possible w_c closest means. We must consider N-1 different k values, and for each k value, determine the k means. For large datasets, this is not feasible as it is computationally heavy. Also the function would choose the largest value of k as the function decreases with k.
2. As mentioned above, evaluating the objective function on test data results in the same problem. As well, we would not have a good model to predict on if we are only using test data.As k increases so does the number of clusters.

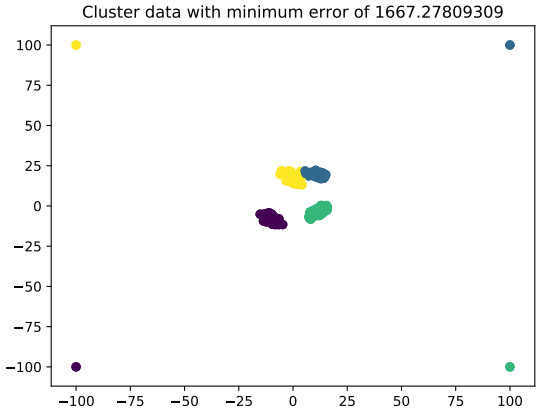


3.

Dashes were used to connect the plot to more clearly view the minimum points at each k value for the next question.

4. The largest change in slope is found between k=1 and k=2. According to this method, k=2 is a suitable k value for this dataset as it provides the most "benefit". The slope between k=1 and k=2 is -3757.885, signifying a large improvement in test error with the addition of one more cluster. As well, the slope between k=2 and k=3 is large, at 2181.363. Beginning at k=4, changes to the error are not drastic, and vary by less than 50. This means that adding one more cluster does not change the overall error by very much. Overall, choosing k=4 is most appropriate as the change in error for subsequent cluster numbers is minimal.

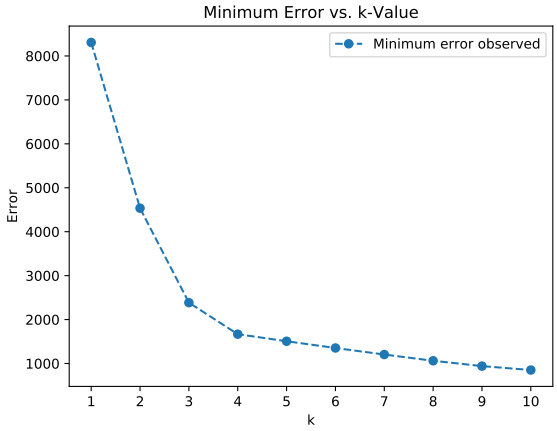
3.3 k-Medians



1.

The lowest error obtained over 50 runs is 1667.278.

2. Approaching the same methodology as in 3.2.4, I ran k-means 50 times, varying k between [1,10]. The following is the plot obtained showing the minimum error obtained in each k.

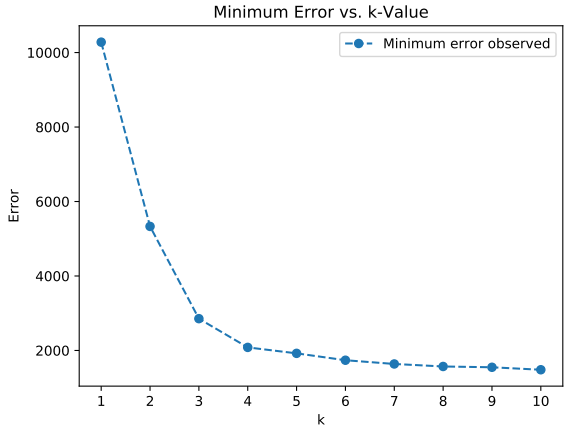


It is apparent, just like in 3.2.4, that the largest slope occured between k=1 and k=2, with value -3773.338. The same is true between k=2 and k=3, and k=3 and k=4. Beginning at k=4, however,

the difference in error is minimal (<10) when adding one more cluster. Thus, a k-value of 4 may be most appropriate for this dataset.

3. The k-medians algorithm can be found at `https://github.ubc.ca/cpsc340/jeanlam_nafis1_hw2/blob/master/code/kmedian.py`. The error obtained is 2082.490.

4. Under the elbow method with running k-median 50 times each over varying k-values of [1-10], the following graph was obtained:



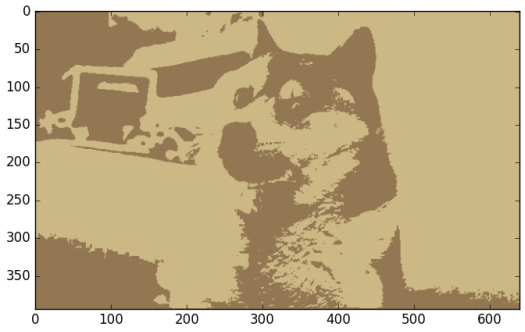
Following the trend, the largest change in slope appears between k=1 and k=2, with a change of -4948.097. K=4 is the most appropriate choice as when k=5 was run, there was a larger error than k=4.

4 Vector Quantization and Density-Based Clustering

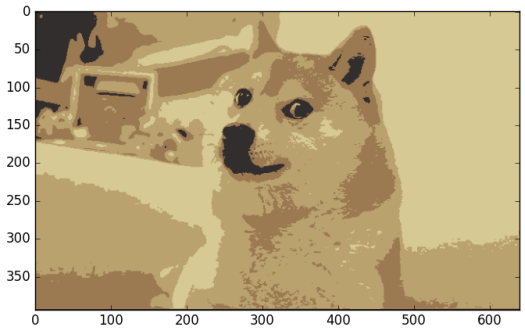
4.1 Image Colour-Space Compression

1. Link to code: `https://github.ubc.ca/cpsc340/jeanlam_nafis1_hw2/blob/master/code/quantize_image.py`

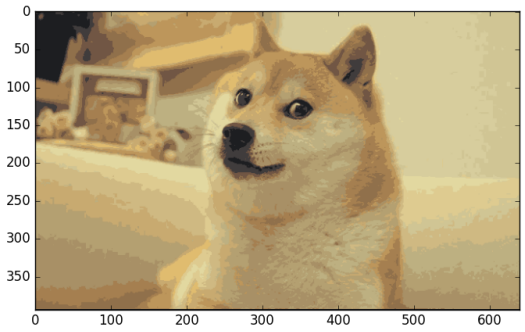
2. Please see doge images here: `https://github.ubc.ca/cpsc340/jeanlam_nafis1_hw2/tree/master/figs` or embedded below
One bit:



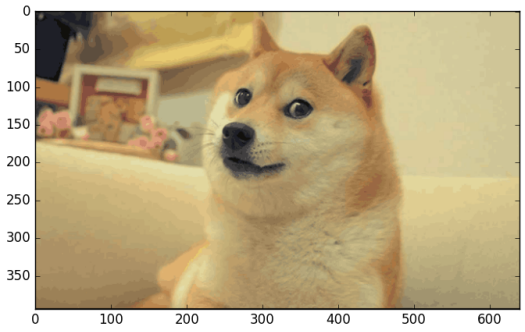
Two bits:



Four bits:



Six bits:



4.2 Effect of Parameters on DBSCAN

Keeping minPts = 3
The 4 true cluster, radius =11
3 clusters,radius =18
2 clusters, radius =180
1 clusters, radius =800

4.3 K-Means vs. DBSCAN Clustering

1. I have chosen r = 15 and kept min =3pts to get the following clusters:
   Cluster 1: antelope horse moose ox sheep giraffe buffalo zebra deer pig cow
   Cluster 2: dalmatian persian+cat german+shepherd siamese+cat mole tiger leopard fox hamster squirrel rabbit wolf chihuahua rat weasel bobcat mouse collie
   Cluster 3: hippopotamus elephant rhinoceros
   Cluster 4: blue+whale humpback+whale seal walrus dolphin
   Cluster 5: spider+monkey gorilla chimpanzee