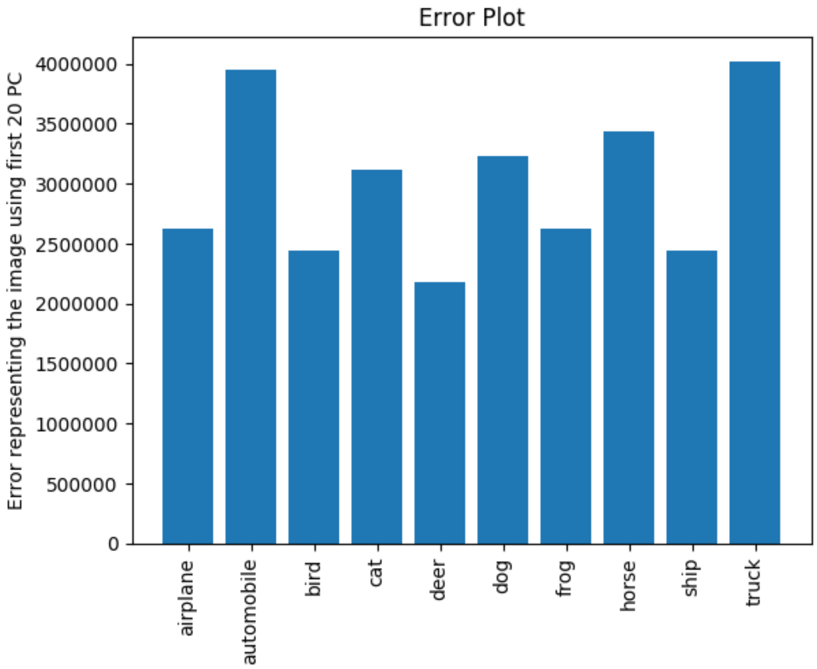
HOMEWORK 3

This homework involves working with CIFAR-10 dataset which contains some 60000 images each having 3072 features and the images are splitted across 10 categories such that each one has 6000 images each. The data is imported in python using pickle and merged all batches in one data frame with index as the labels.

Part 1:

The data is sub-stetted into individual categories each having 6000 examples. Each example is then centered by subtracting each instance from its mean and this centered image is applied to PCA algorithm using 20 PC.

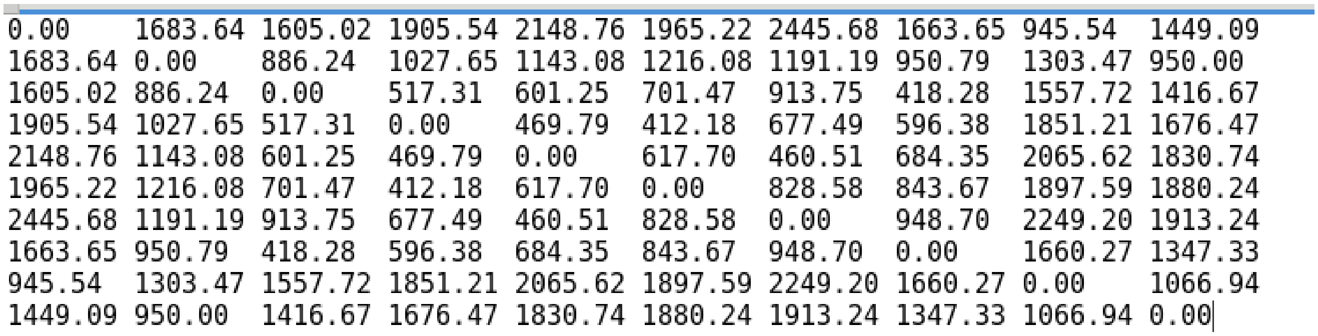
Error is calculated by subtracting the original image and the image reconstructed by applying inverse_tranform over the 20 Eigen vectors. We then square this difference, sum them by for each image and take the mean of that such that we get a scalar quantity. We get some error for each category and then plot it on a bar graph. Following is the error reported:



Part 2:

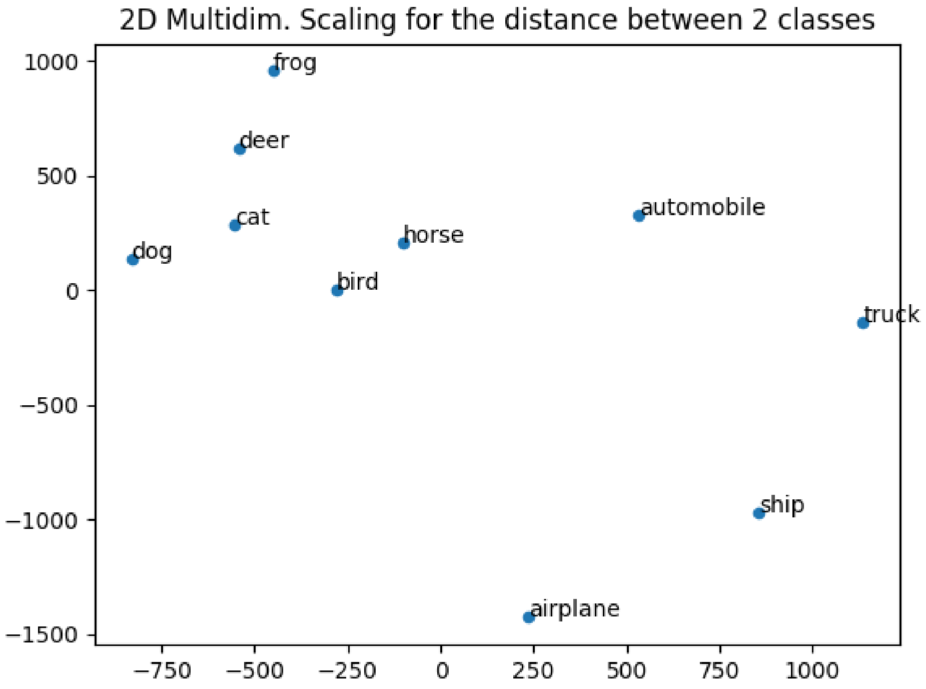
So here we have the mean for each category and we then calculate a pair wise Euclidean distance between each category mean pair and put this in a matrix of 10*10 such that each row represents the distance of one category with all the other categories. We then do a Multi-Dimensional Scaling of this symmetric matrix such that we project all the points on to a 2D map.

The distance matrix is as follows:



```
0.00    1683.64 1605.02 1905.54 2148.76 1965.22 2445.68 1663.65 945.54  1449.09
1683.64 0.00    886.24  1027.65 1143.08 1216.08 1191.19 950.79  1303.47 950.00
1605.02 886.24  0.00    517.31  601.25  701.47  913.75  418.28  1557.72 1416.67
1905.54 1027.65 517.31  0.00    469.79  412.18  677.49  596.38  1851.21 1676.47
2148.76 1143.08 601.25  469.79  0.00    617.70  460.51  684.35  2065.62 1830.74
1965.22 1216.08 701.47  412.18  617.70  0.00    828.58  843.67  1897.59 1880.24
2445.68 1191.19 913.75  677.49  460.51  828.58  0.00    948.70  2249.20 1913.24
1663.65 950.79  418.28  596.38  684.35  843.67  948.70  0.00    1660.27 1347.33
945.54  1303.47 1557.72 1851.21 2065.62 1897.59 2249.20 1660.27 0.00    1066.94
1449.09 950.00  1416.67 1676.47 1830.74 1880.24 1913.24 1347.33 1066.94 0.00
```

Here is the plot for that:

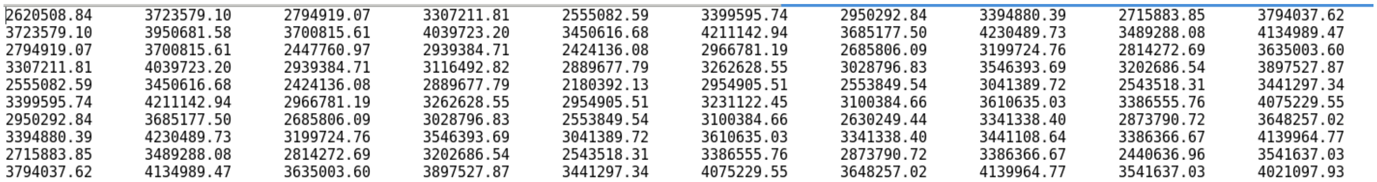


Reasoning: In the above plot we can see that there is a cluster of items formed where each item is some kind of animal (dog, cat, deer, frog, bird, horse) and the rest of the item categories are scattered across in the plot, such kind of behavior is expected since the cluster item share some common attributes and hence are placed closely in the vector space where-as the other item do not share much in common and are randomly lying in the cluster space. The above plot is just a projection of the vector space on to 2 dimensions where the distance shows us the similarity between items. Thus we can conclude from the above plot that the items that form a cluster have some similarity with each other.

Part 3:

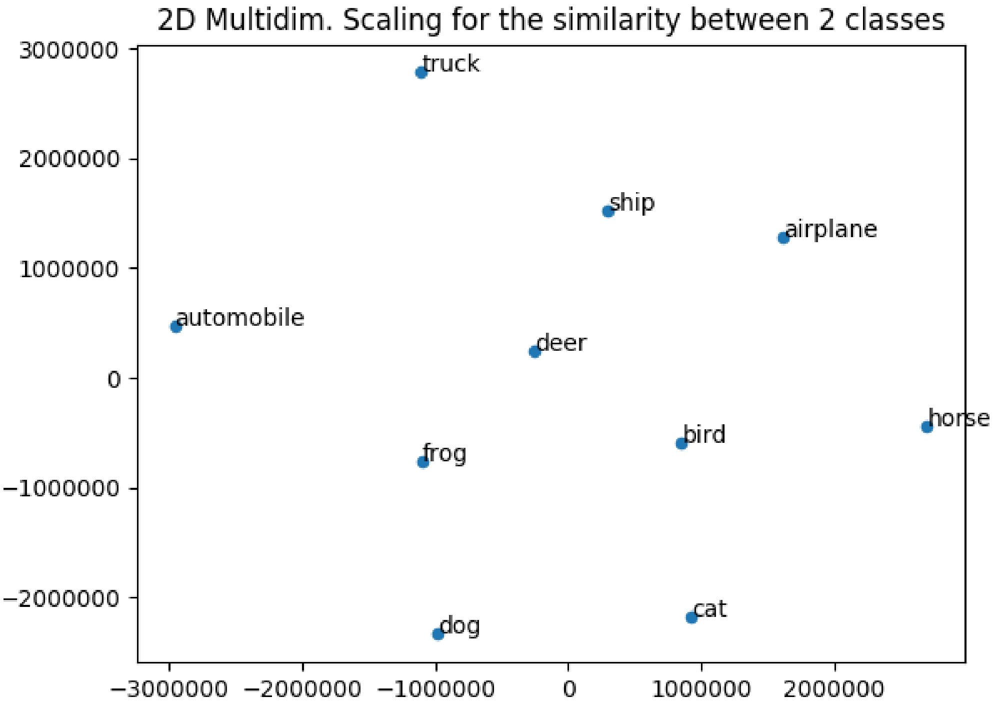
Here we tend to define the similarity between each category pair by subtracting the original image with the approximate image constructed by doing inverse transformation of the principal components of other category and the mean of given category. After that we calculate the similarity for any two given categories by doing (1/2)(E(A|B) + E(B|A)).

The matrix is as follows:



```
2620508.84     3723579.10     2794919.07     3307211.81     2555082.59     3399595.74     2950292.84     3394880.39     2715883.85     3794037.62
3723579.10     3950681.58     3700815.61     4039723.20     3450616.68     4211142.94     3685177.50     4230489.73     3489288.08     4134989.47
2794919.07     3700815.61     2447760.97     2939384.71     2424136.08     2966781.19     2685806.09     3199724.76     2814272.69     3635003.60
3307211.81     4039723.20     2939384.71     3116492.82     2889677.79     3262628.55     3028796.83     3546393.69     3202686.54     3897527.87
2555082.59     3450616.68     2424136.08     2889677.79     2180392.13     2954905.51     2553849.54     3041389.72     2543518.31     3441297.34
3399595.74     4211142.94     2966781.19     3262628.55     2954905.51     3231122.45     3100384.66     3610635.03     3386555.76     4075229.55
2950292.84     3685177.50     2685806.09     3028796.83     2553849.54     3100384.66     2630249.44     3341338.40     2873790.72     3648257.02
3394880.39     4230489.73     3199724.76     3546393.69     3041389.72     3610635.03     3341338.40     3441108.64     3386366.67     4139964.77
2715883.85     3489288.08     2814272.69     3202686.54     2543518.31     3386555.76     2873790.72     3386366.67     2440636.96     3541637.03
3794037.62     4134989.47     3635003.60     3897527.87     3441297.34     4075229.55     3648257.02     4139964.77     3541637.03     4021097.93
```

Here is the plot:



<u>Reasoning:</u> The above plot shows us how much dissimilar one category is with respect to another category in terms of constructing one category using the prominent features of other category. Here a category is generated using the 20 principal components of other category and the mean of its own category. In this plot we see a small cluster tendency as compared to what we saw in part 2 where we calculated how far 2 given mean vectors are. In this case the distances between category would be effected by the notion of how far away are the principal components from one another.

Here we take the most evident approximation(mean) of a category and the most variable directions of the other category and try generating the original image , then we subtract these 2 images square them sum them over and take the mean which tells us the error occurred in generating one category from the other .While the second plot accounts for difference between all the features of the 2 categories this plot takes into account the variation between categories based on the other category principal components.

Thus to conclude in plot 2 we saw clustered item sharing some similarity measures whereas in plot 3 we see cluster item sharing some dissimilarity, since objects that are similar will have less dissimilarity the above plots justifies that behavior.

Citation:

We looked up for some sklearn pca functions in stack overflow and had some discussions with TA(Eric) on Slack.

https://stackoverflow.com/questions/25192093/what-is-the-correct-input-to-scikit-learns-mds.

https://stackoverflow.com/questions/32857029/python-scikit-learn-pca-explained-variance-ratio-cutoff