Topics in Unsupervised Learning HW 1

Shir Hamawie & Nadav Shoham

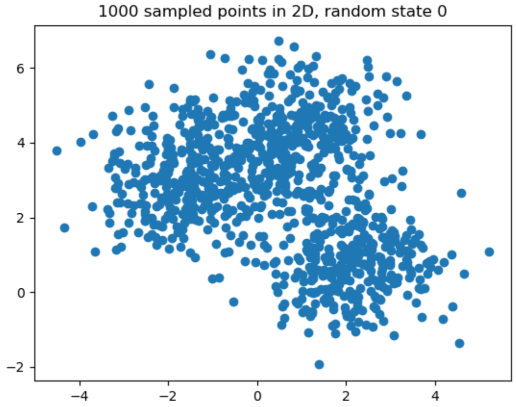
April 21, 2023

Introduction

In this assignment, we will implement the k-means and dc-dp-means clustering algorithms and evaluate their performance on different artificial datasets with different hyperparameters. We will compare the clustering results of the two algorithms and analyze their strengths and weaknesses.
We will generate synthetic 2D data of N = 1000 points with different random states to see how our algorithms perform at different situations.

Task 1

In this task we implemented both algorithms and checked how they performed on the following generated data:



The following images are our clustering results with k = 3 and l = 4:

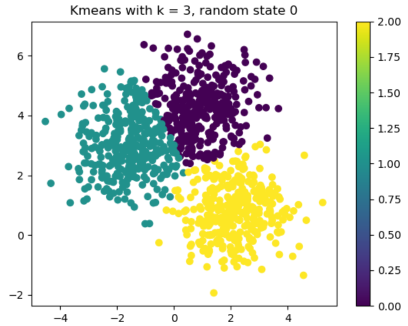


Figure 1: Kmeans

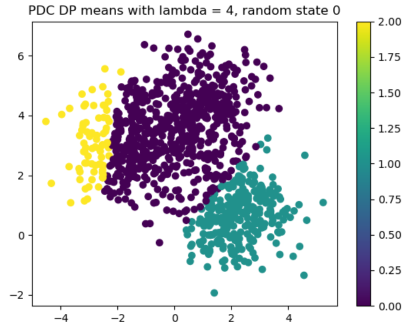


Figure 2: DC-DP Means

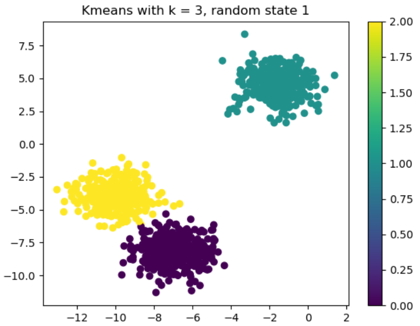
We can see we got fairly reasonable clusters, where the kmeans clustering looks like better separation.

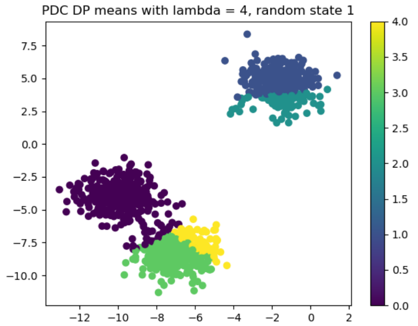
Task 2

The purpose of this part is to investigate the behavior of the algorithms in different scenarios and gain insights into their strengths and limitations. First, we will test the algorithms on different datasets with different random states and compare the results. Then we will test the algorithms on the same dataset with different hyper parameters and compare the results.

Different samples space initializations

In this section, we will present the experimental results of applying the clustering algorithms on four more randomly generated 2D datasets with varying degrees of complexity and structure. We will evaluate the performance of the algorithms using the same hyper parameters as before and analyze the resulting clustering outputs.





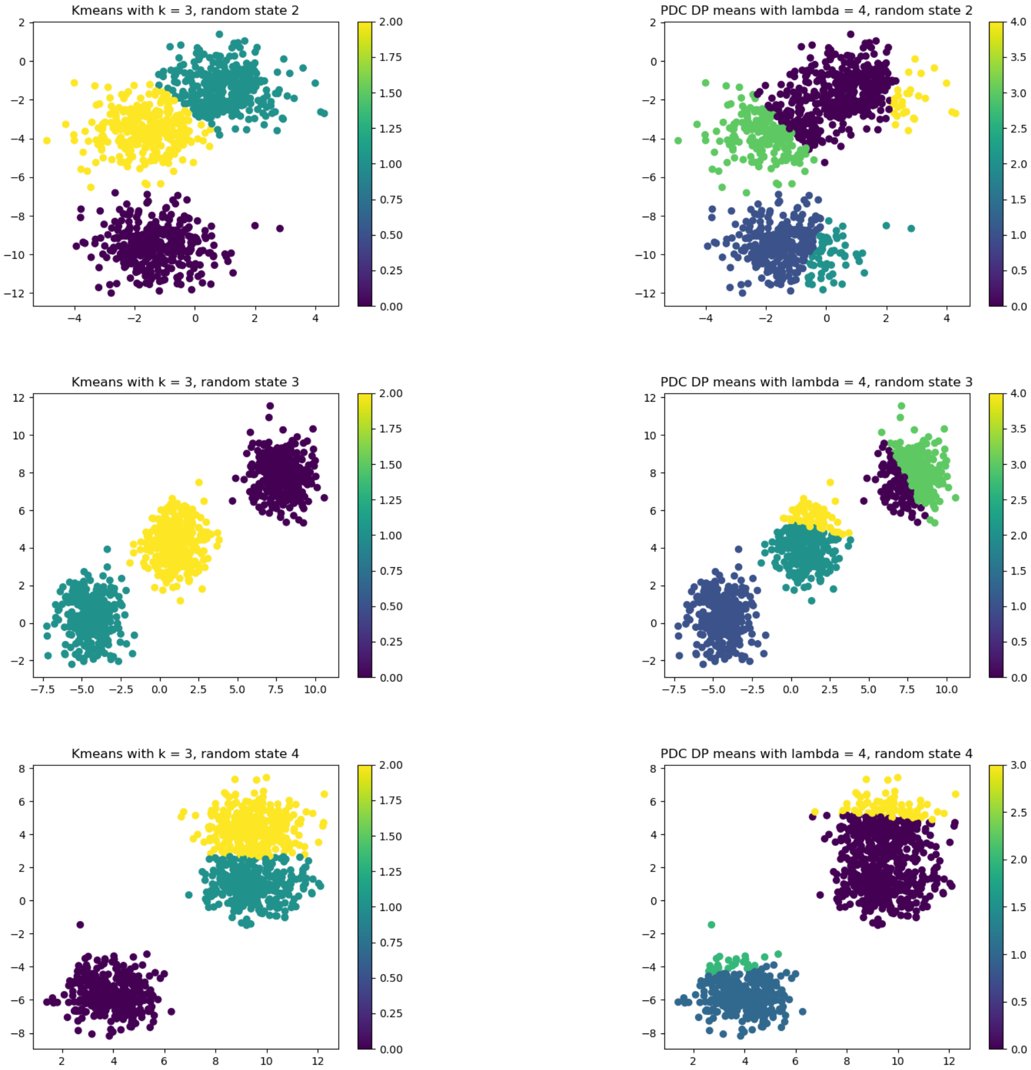


Figure 3: Diffrent samples space initializations

Based on the figures above, both algorithms performed reasonably well on the datasets. However, k-means seemed to perform slightly better in terms of producing visually distinct clusters with clear boundaries. This is likely due to the choice of the lambda in dc-dp means, which may have led to too many splits and resulting in the formation of more numerous, smaller clusters. Either way the results demonstrate the different behavior of the two algorithms.

Different hyper parameters

Now we will test our algorithms on the same dataset, with different values for their respective hyper parameters.

Kmeans:

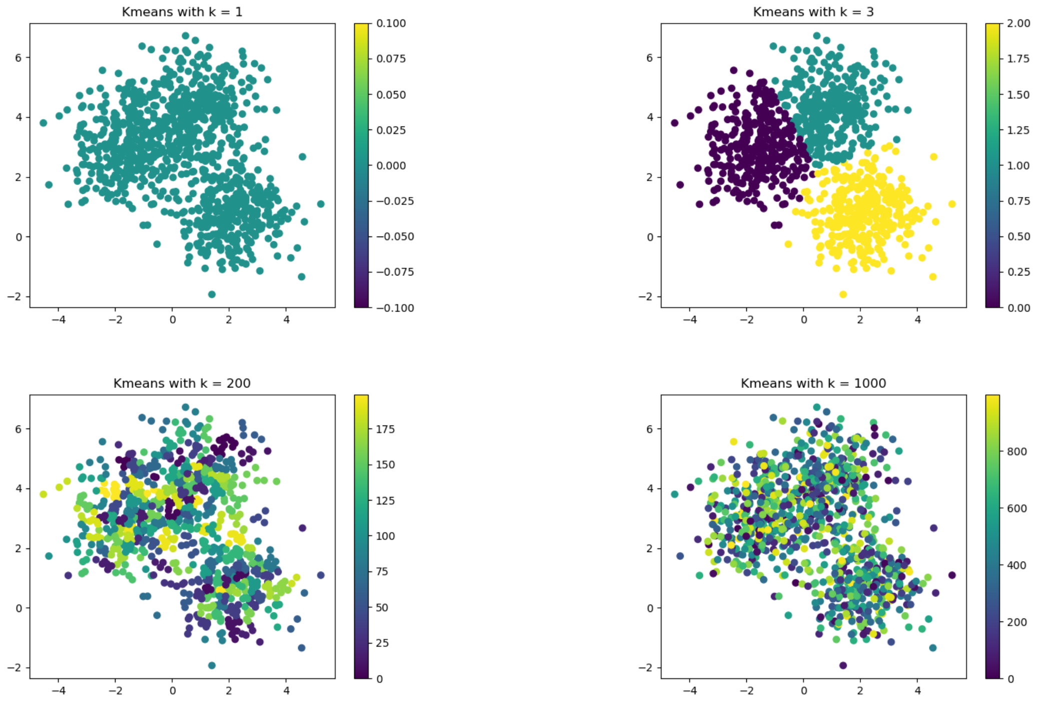
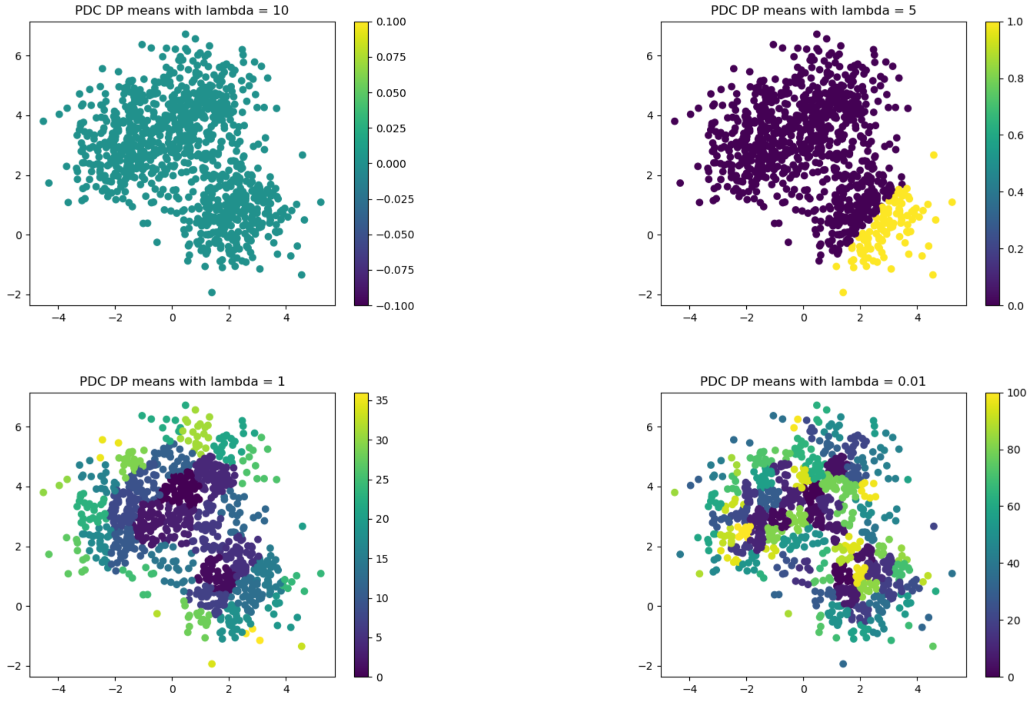


Figure 4: Kmeans with different values of k

The initial experiment with k=1 resulted in all the samples being assigned to a single cluster, which is not useful for data analysis. However, as the value of k was increased to 3, the algorithm produced a clustering result that appeared to be the most reasonable, with distinct clusters formed around the data's structure. As k was further increased, the number of clusters gradually increased,but the resulting clusters became less cohesive, and the boundaries between them became blurred. This highlights the trade-off between the number of clusters and the cohesiveness of the clusters, which is a critical factor in choosing the optimal value of k for k-means clustering.

DC-DP-Means:


PDC DP means with lambda = 10

PDC DP means with lambda = 5

PDC DP means with lambda = 1

PDC DP means with lambda = 0.01

Figure 5: DC-DP-Means with different values of l

We started with l=10, which produced a single cluster encompassing all the samples in the dataset this makes sense because l indicate the maximal distance between points in a cluster and 10 captures all the space. As we gradually decreased the value of l, the number of clusters increased, and the resulting clusters became more cohesive and distinct. Yet around 1 the number of clusters increased dramatically, and the clusters became less cohesive and more blurred. The best clustering result is probably when l is set between 2-4.

Task 3

Now we will apply the algorithms we have developed to a real-world dataset. We have taken the following image of a mandrill monkey from the web and compressed it to reduce calculation time.

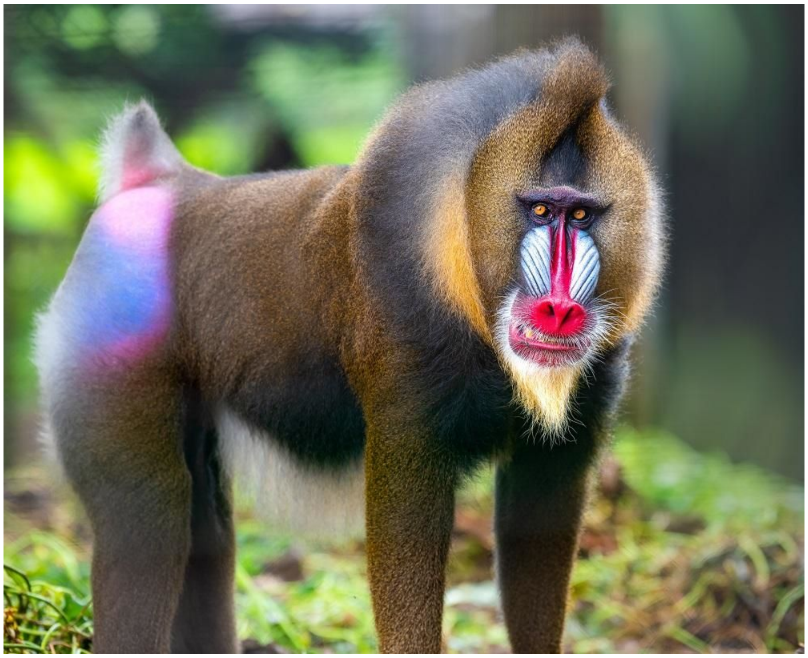


Figure 6: Real image (scaled down)

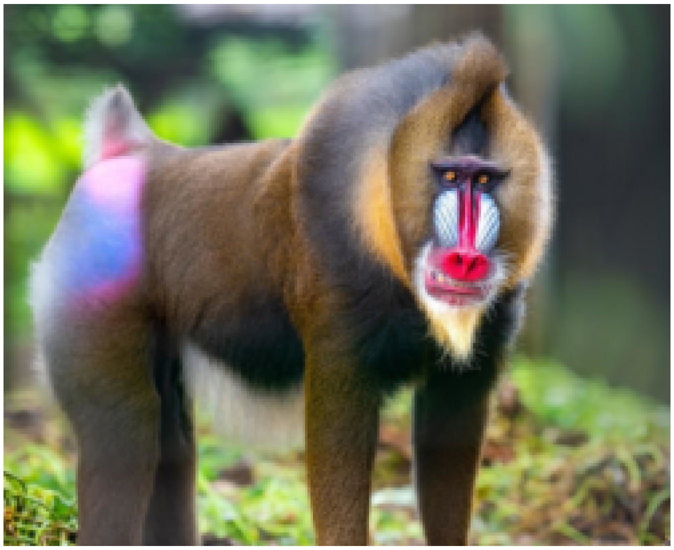


Figure 7: Compressed image

K-means:

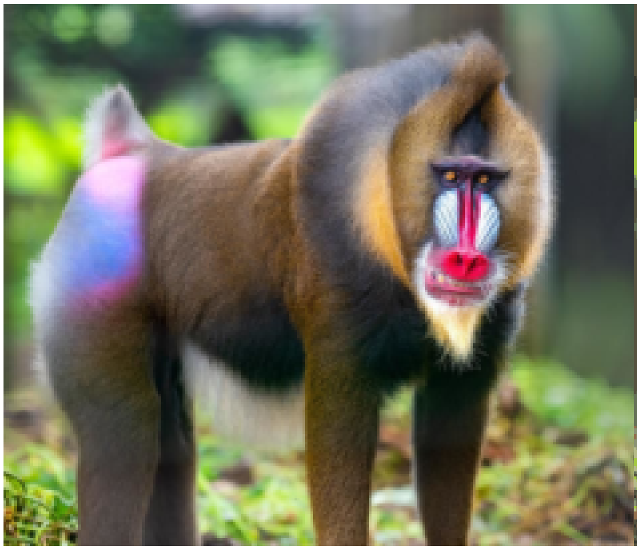


Figure 8: Original

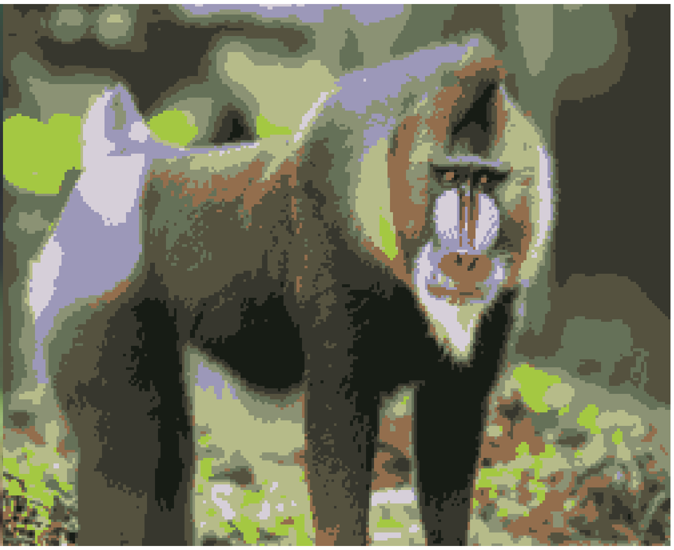


Figure 9: k = 10

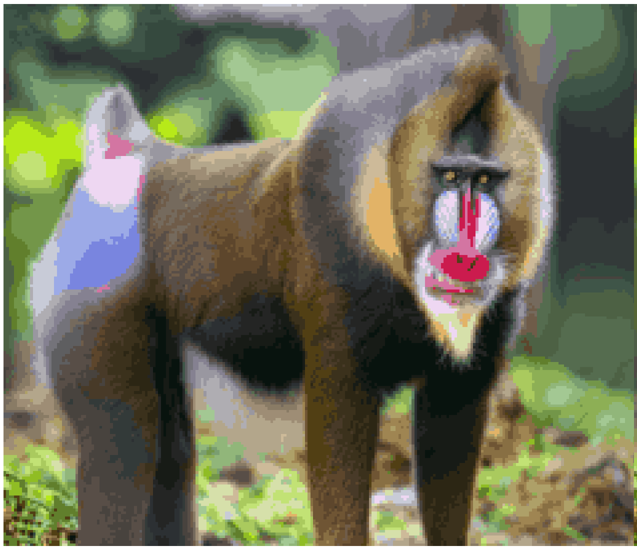


Figure 10: k = 50

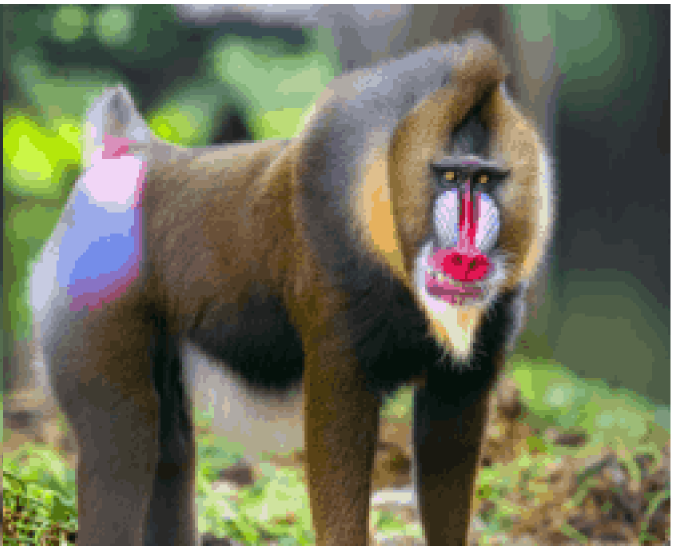


Figure 11: k = 100

As seen above, when choosing k=10, the algorithm produced a clustering result that appears lacking in the distinctiveness of the mandrill's colors. This is expected, as 10 colors is not enough to capture the full range of colors in the image. As k was increased to 50, the resulting clusters became more cohesive and distinct, and the colors and structure of the mandrill became more apparent. At k=100 we get a pretty good clustering result, looking almost identical to the original image at most places.

DC-DP-Means:

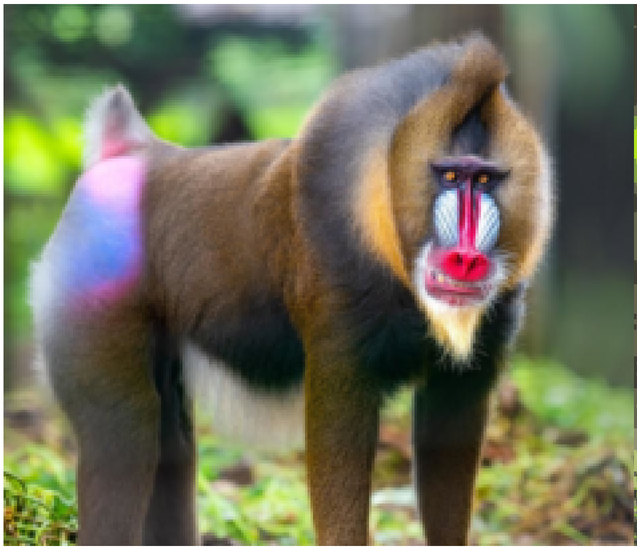


Figure 12: Original

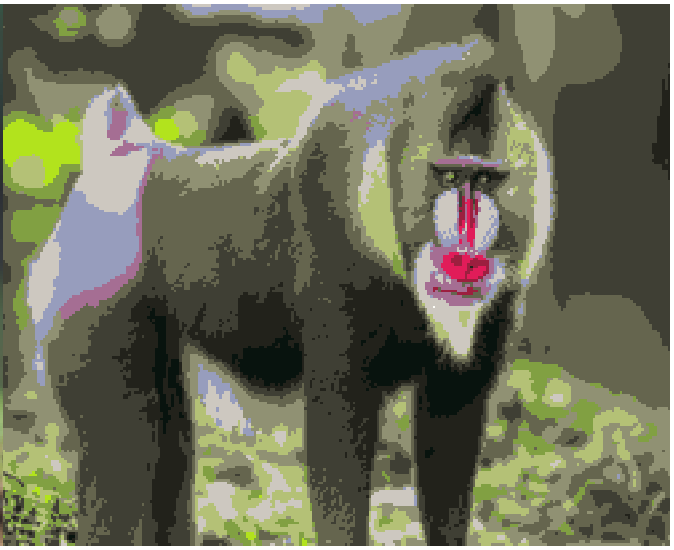


Figure 13: l = 100

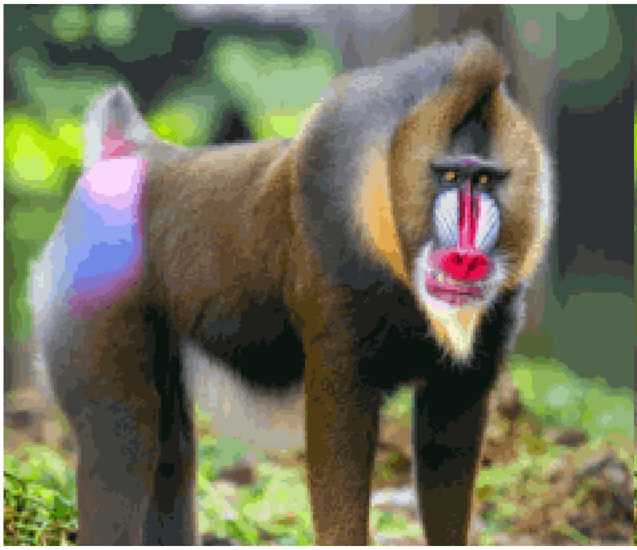


Figure 14: l = 20

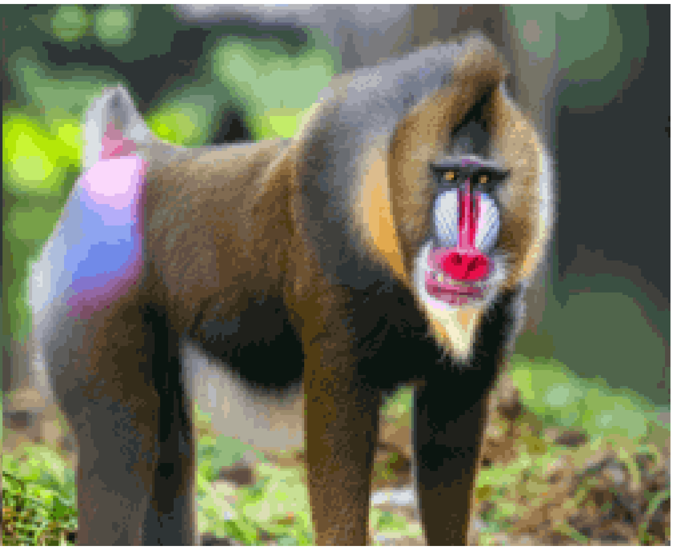


Figure 15: l = 5

As seen above, when choosing l=100, the algorithm produced a clustering result that appears lacking like when we chose k = 10. This is also expected, as 100 is a large distance required to split a cluster, therefore we end up with a few large clusters. As l was decreased to 20, the results improved dramatically, and the colors and structure of the mandrill became more apparent. decreasing l further to 5 didn't seem to improve much on the already impressive result.