

# K-Means Clustering

## Methodology:

### K-Means Algorithm:

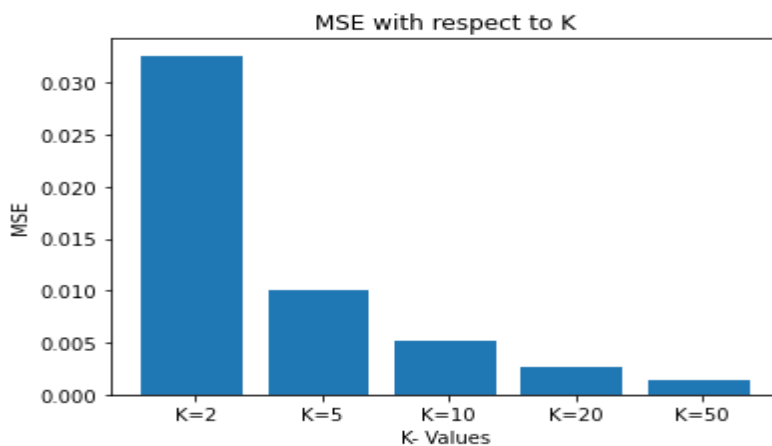
1. Initialise cluster centres,temp centres.
2. Randomly pick  $k$  vectors from the dataset and set them as cluster centres.
3. Assign temp centres with cluster centres
4. Iterate each point in the dataset, find the distance between cluster centres and add to the  $k$ th-set which cluster centre has minimum distance.
5. Now find the average points in each cluster and make them as cluster centres.
6. If  $\text{norm}(\text{cluster centres} - \text{temp centres}) < \text{epsilon}(10^{-6})$  , break the loop
7. Else goto step 3

Like this I iterated over  $k = [2,5,10,20,50]$  cluster size and updated the vectors in the data set to the nearest cluster centre.

Then i calculated Mean Square error with updated\_vec vs real\_vec by using `np.square(real_vectors-updated_vectors).mean()`

MSE Values = [0.032644, 0.009996, 0.0051141, 0.0027099, 0.0013567]

### Plot between K and MSE:



MSE Plot

Here when K values increase MSE value decreases due to more cluster centres with RGB values being present so that image captures better pixels compared to previous that's why MSE is decreased.

## Reconstruction of Image:

After getting the updates\_vectors reshaping into original image shape that is (512,512,3) for every k in [2,5,10,20,50]



**K = 2**



**K = 5**



**K = 10**



**K = 20**



**K=50**