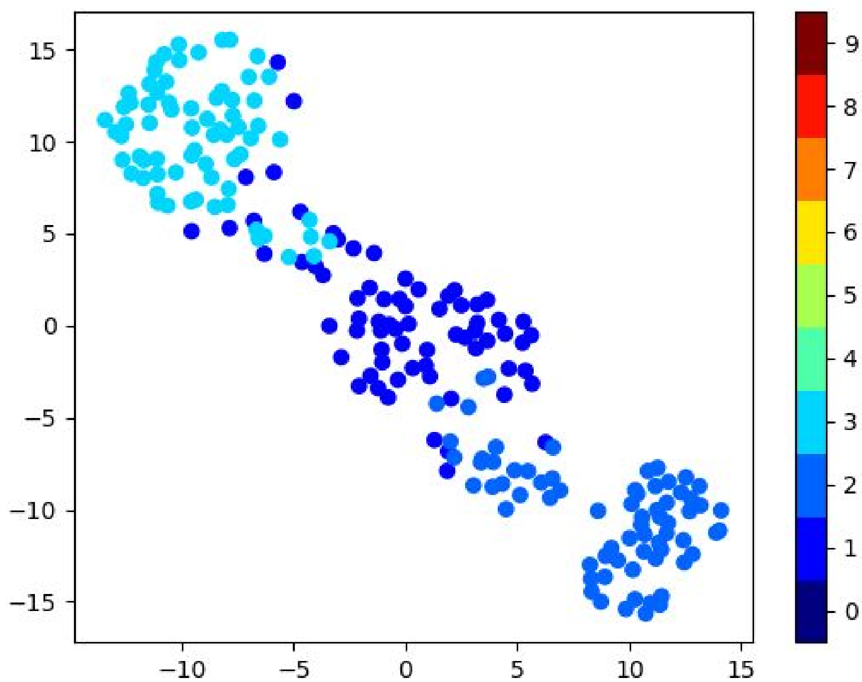
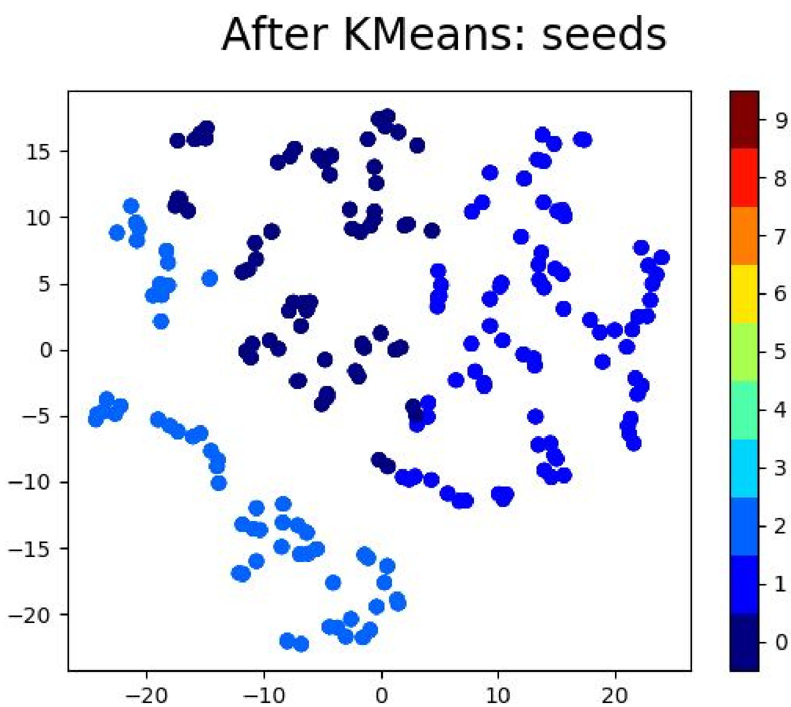
ML Assignment 4 Report
Neha Jhamb
MT16037

K Means

Note: The data is cleaned and shuffled before applying K means.
The following graphs are for true value of K.
The graphs below show the transition after applying K means to the four datasets:

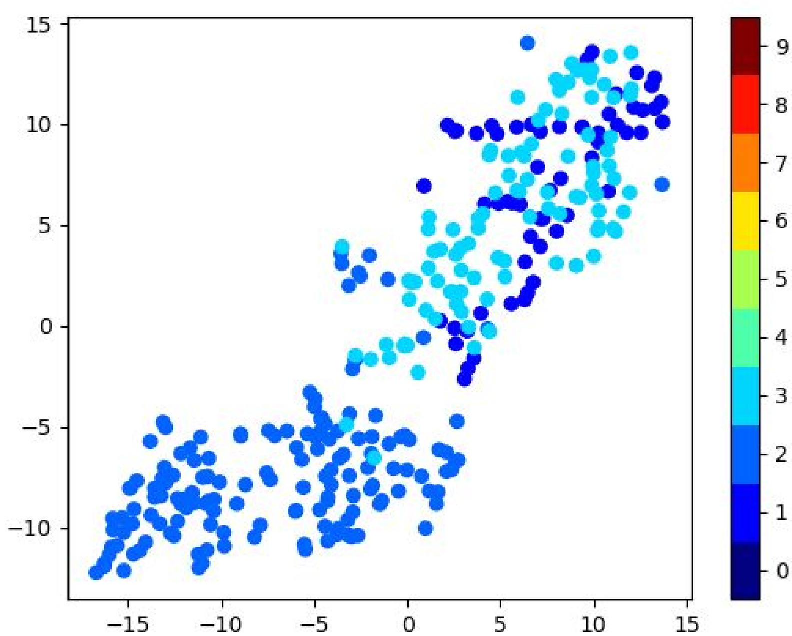
- **Seeds**

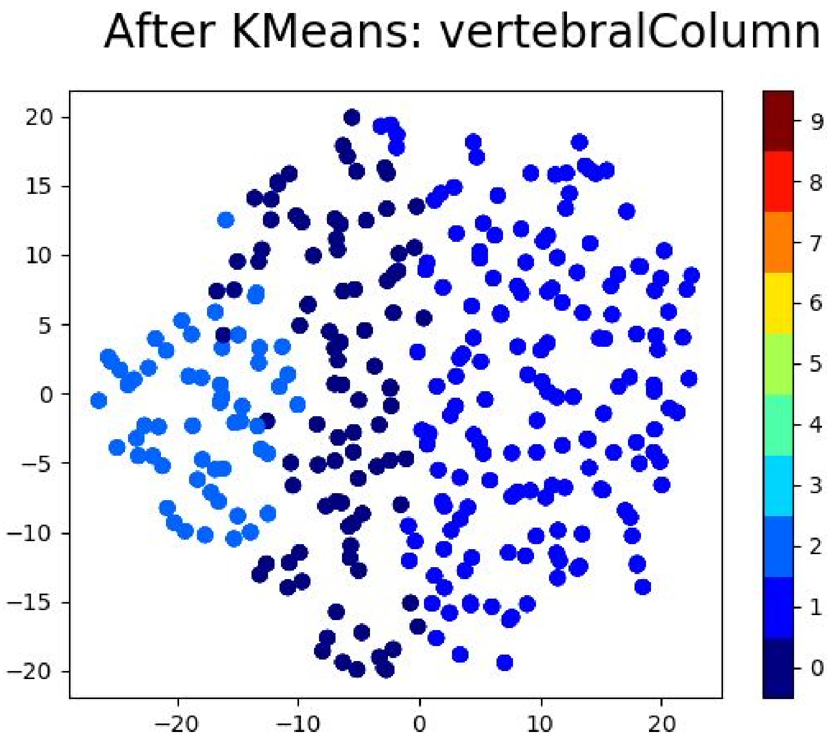




Before applying K means, the clusters are not very well defined. But after the application of K means, the three clusters, as plotted above, are clearly separable.

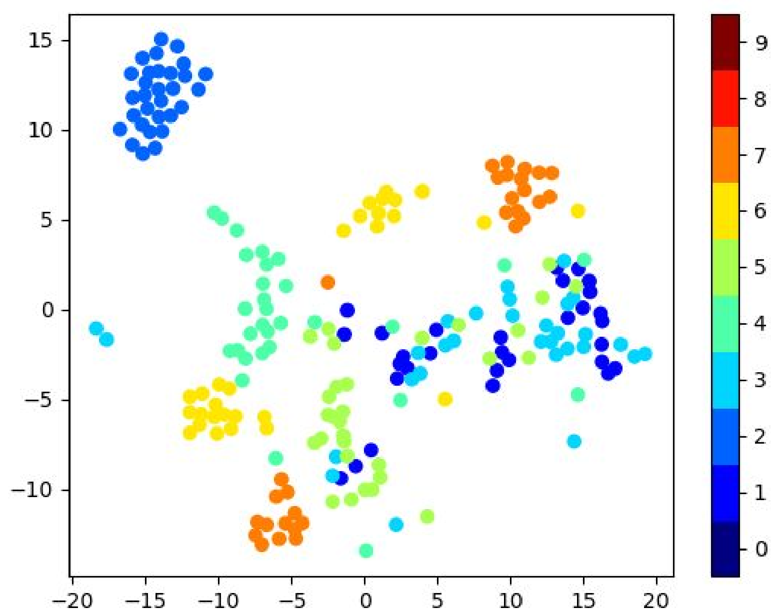
- **Vertebral Column**

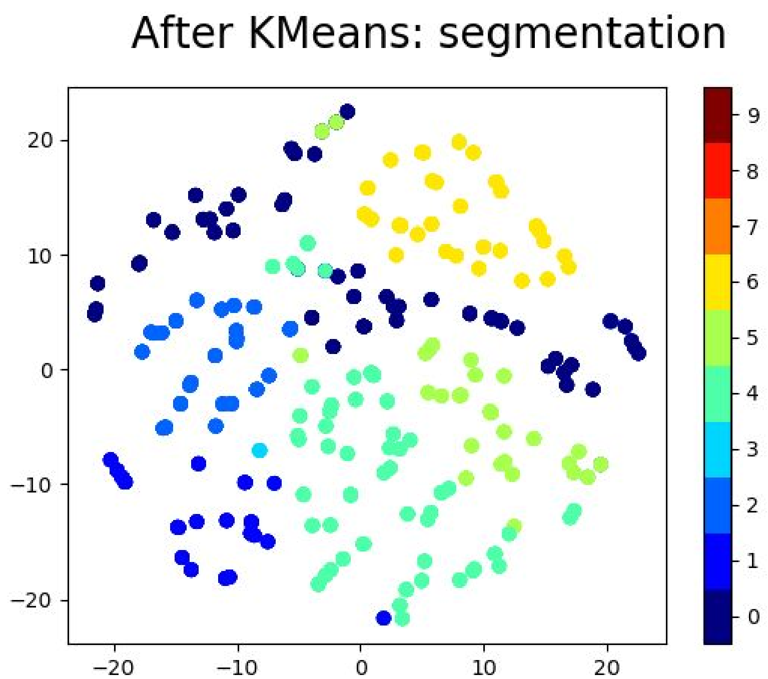




Before K means is applied, the classes 2 and 3 of the Vertebral Column dataset are mixed up. After K means is applied, all the three classes have formed three different clusters which are clearly separable.

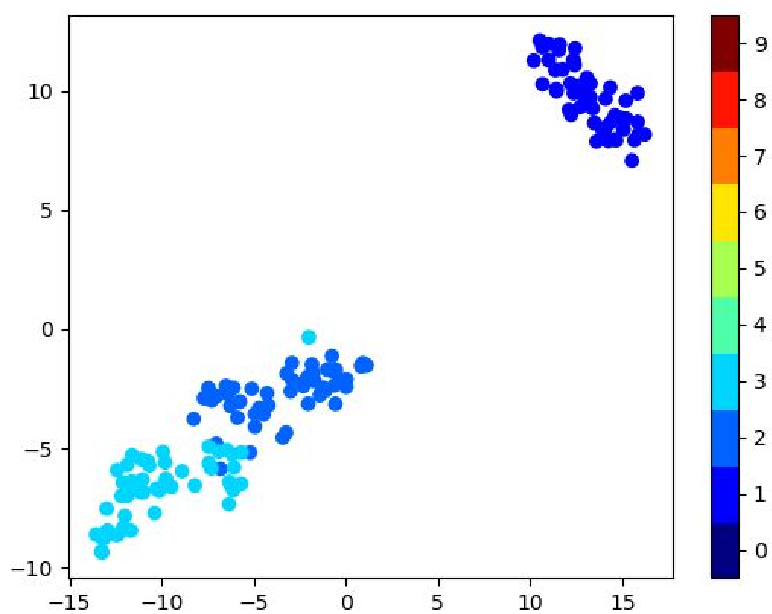
- **Image Segmentation**

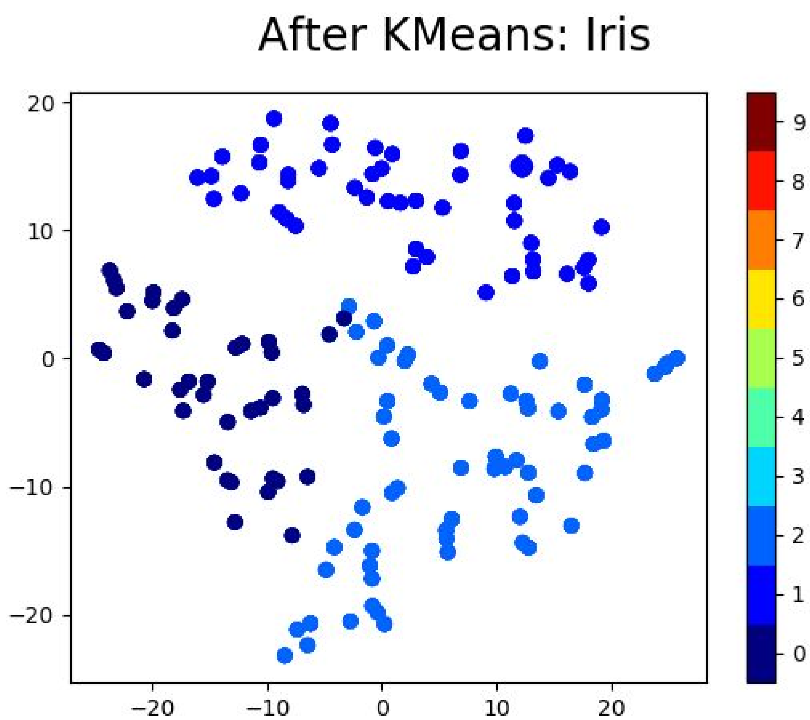




Since the Image Segmentation dataset has data points belonging to multiple classes mixed up with each other, K means works very well in clustering the data. After K means is applied, the clusters are well defined (apart from the few outliers).

- **Iris**

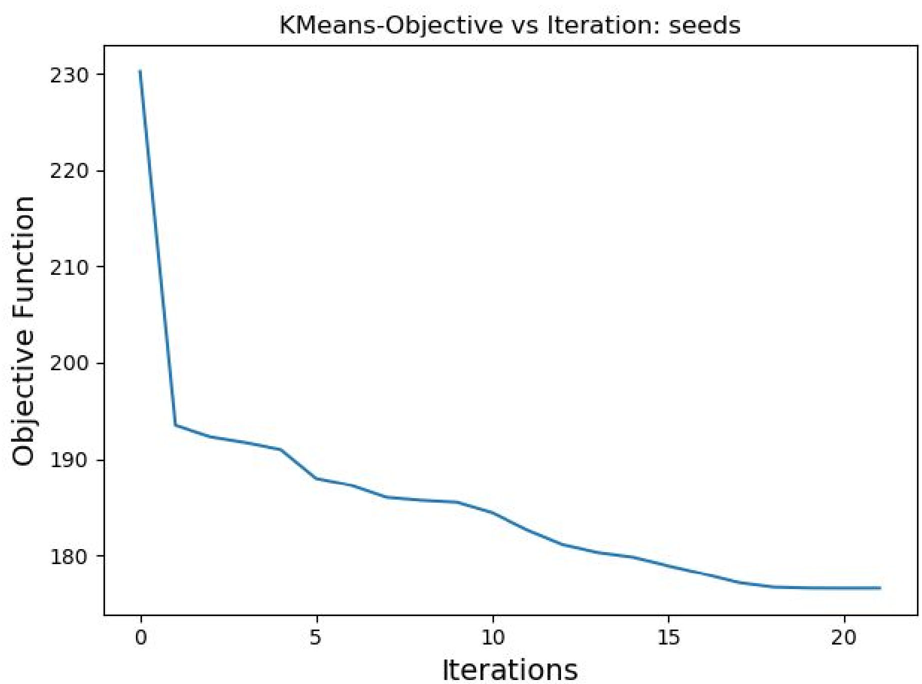




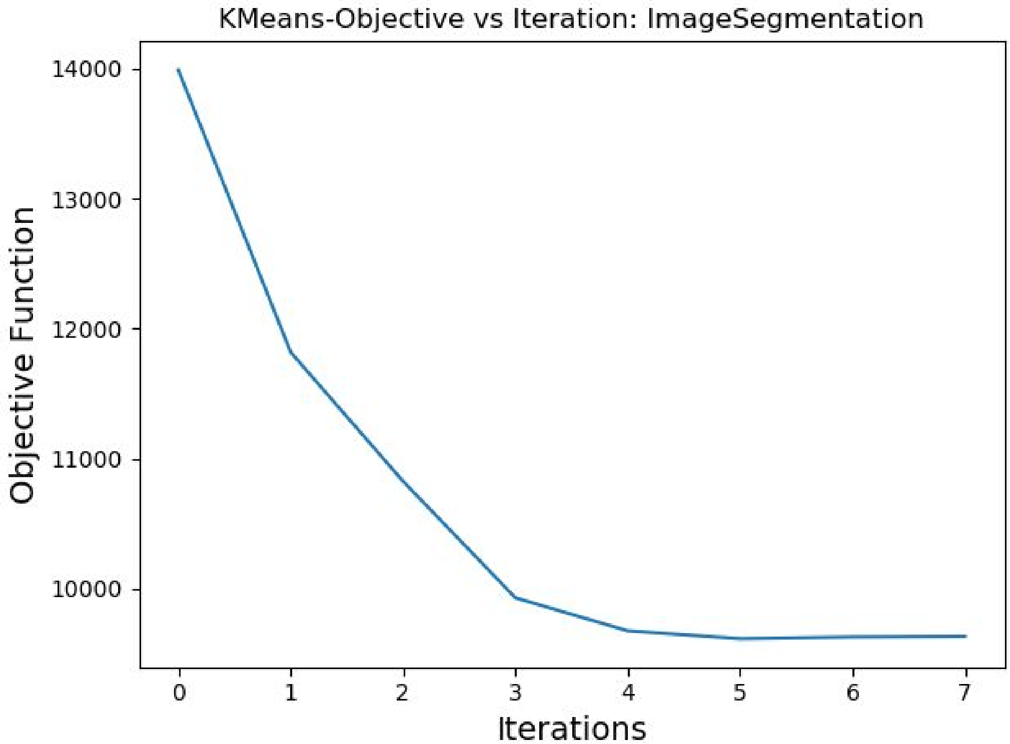
In the Iris dataset also, applying K means makes the clusters for the three classes well defined.

Objective Function vs Iteration Number :

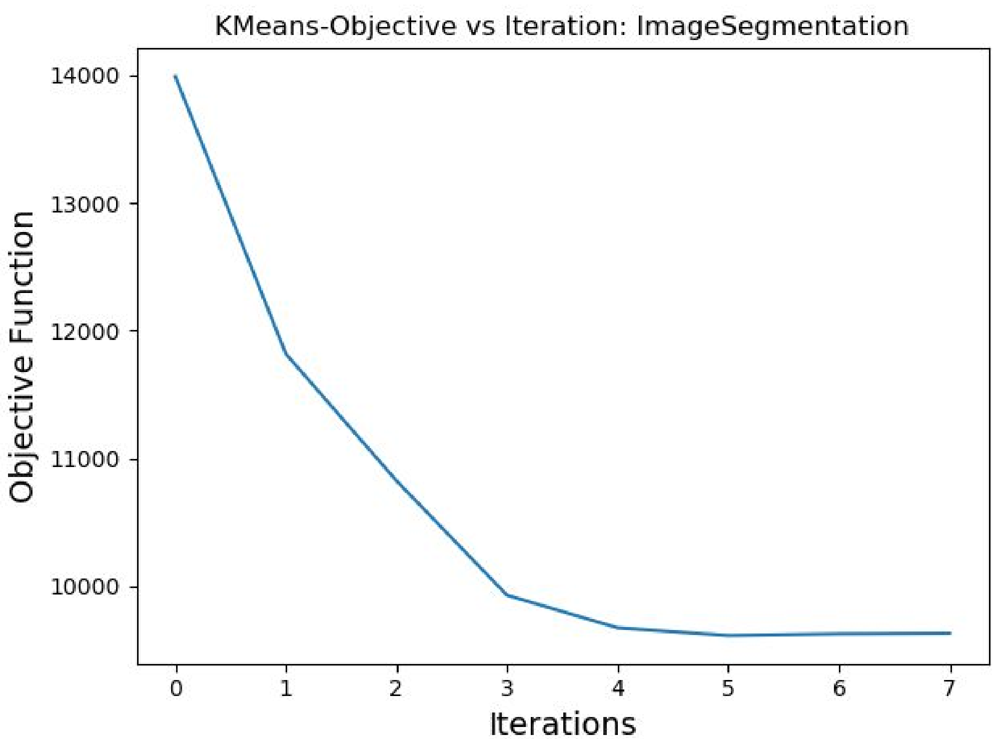
- **Seeds**



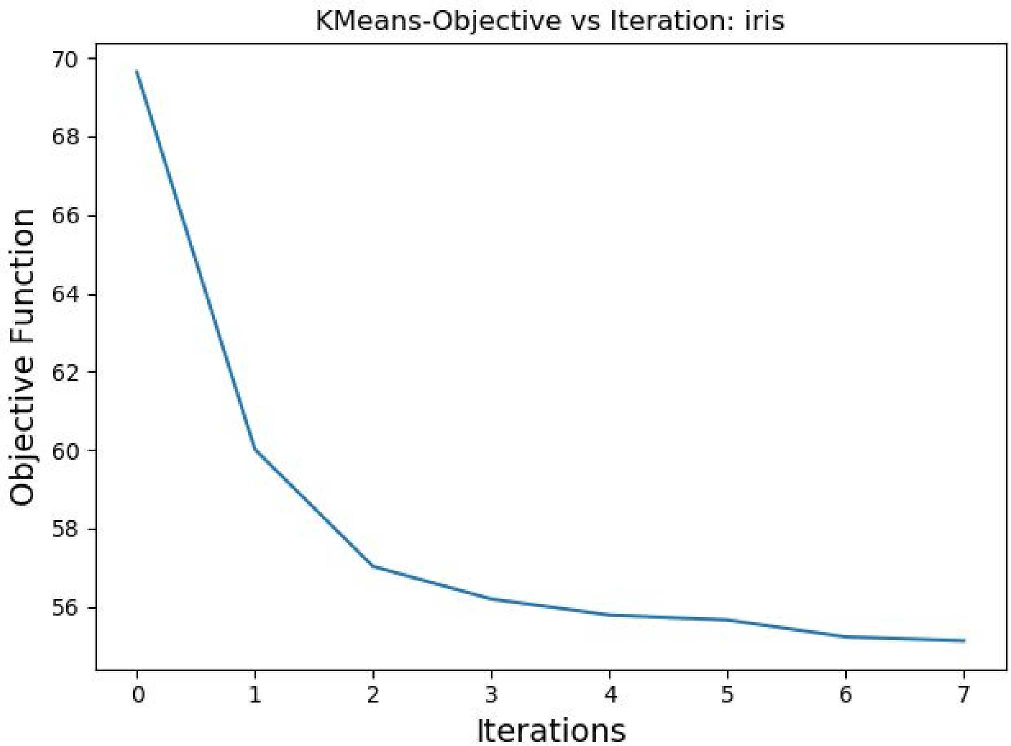
- **Vertebral Column**



- **Image Segmentation**



- **Iris**



We notice that the value of the objective function decreases with the number of iterations and then becomes constant. This is because the objective function is defined as:
The sum of (sum of distances between the centroid and the data points in that cluster) for all the K clusters. This keeps decreasing until the centroids stop changing. Convergence refers to the centroids staying constant after a certain number of iterations.

Note that this trend remains same for all the datasets.

Analysis and Inference (Qualitative or visual):
The K means clustering works very well for the Image segmentation and the Vertebral Column datasets as the clusters become separable only after applying K means. However for the seeds and iris datasets, the clusters even before applying K means are good enough.

Averaged over 5 iterations:

Data	K=2			K = true value			K = 12		
	ARI	NMI	AMI	ARI	NMI	AMI	ARI	NMI	AMI
Iris	0.539921829421	0.679322701116	0.519360805606	0.665919394275	0.720617426702	0.700680096063	0.326631504285	0.632024742075	0.410544820606
Segmentation	0.099636499070	0.398441022257	0.186743579709	0.329367022487	0.505623188144	0.454053486082	0.385023245259	0.608623207363	0.526771277667
Seeds	0.466200647778	0.55179198735	0.425780721317	0.712852967855	0.704011611083	0.699262400369	0.289916129166	0.538682600295	0.350413967866
Vertebral	0.298846071044	0.424966802408	0.334934951236	0.30953686047	0.418114605177	0.409368183433	0.184148027347	0.422613431438	0.272431967745

Analysis and Inference (Quantitative):
For seeds and Iris datasets, the values for ARI, NMI and AMI are around 0.7 for the true values of K. However for Image Segmentation and Vertebral Column datasets, the ARi, NMI and AMI values are around 0.4 for the true values of K. This means that there is a larger difference in the true and the predicted labels.

Comparison: The qualitative and quantitative analysis are consistent for the true values of K as is clear from the graphs and table above.

Also, for K=2, we notice that the values for the three metrics decreases mostly for all the datasets. For K=12, the values decrease for Iris, seeds and the Vertebral column dataset but increase for Image Segmentation dataset as the number of actual classes in this dataset is more.

==

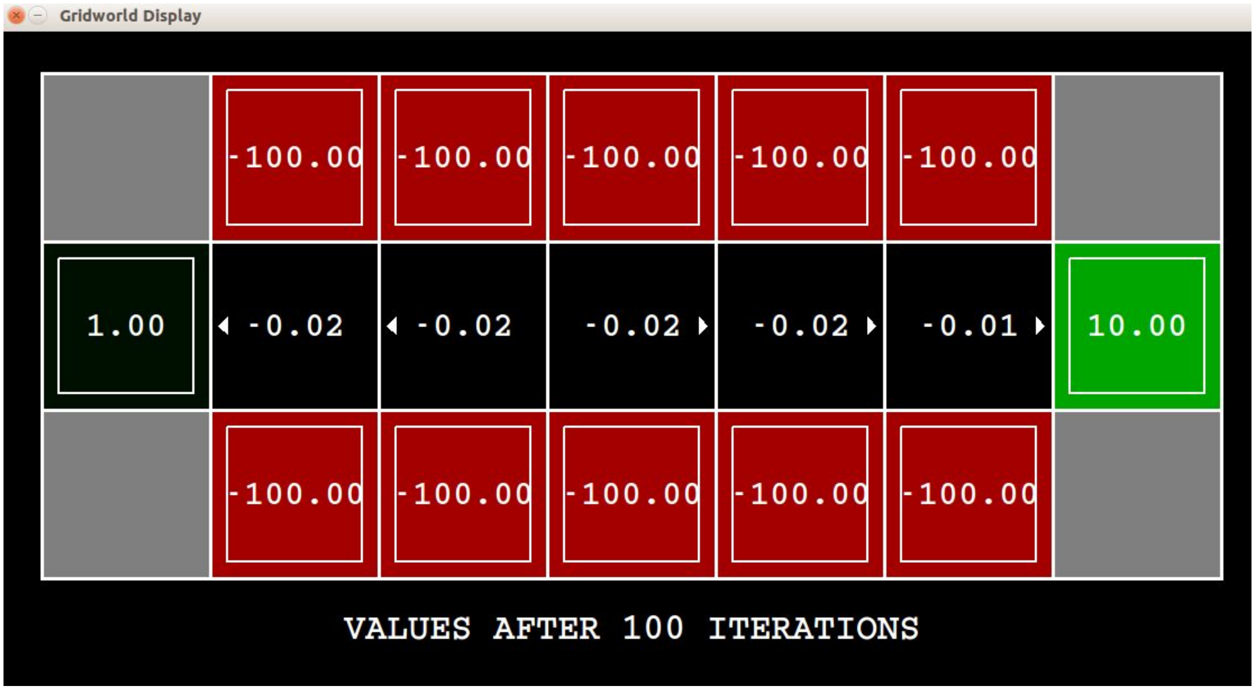
REINFORCEMENT LEARNING

Q2: The screenshots for changed values are:

Tried reducing the discount first.
Discount = 0.5, Noise = 0.2

Discount = 0.3, Noise = 0.2

Discount = 0.001, Noise = 0.2



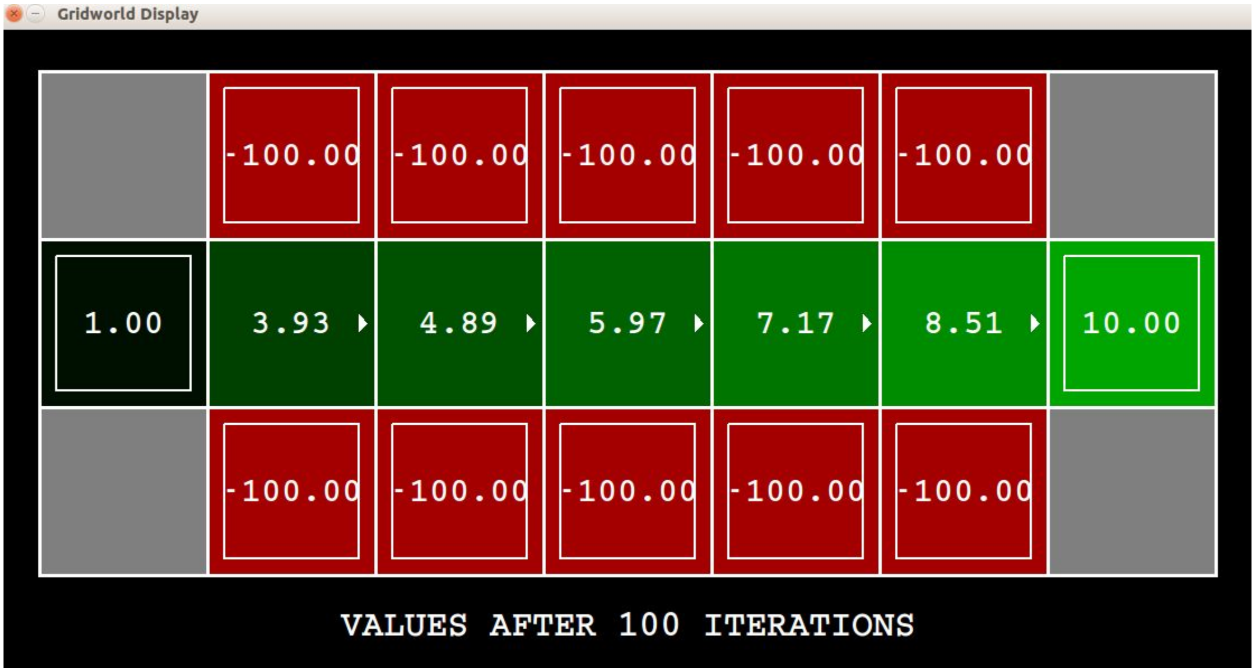
Then, tried changing the noise.

Discount = 0.9, Noise = 0.3

As I increased the noise, the values became more negative. So, then I tried reducing the noise.
Discount = 0.9, Noise = 0.1

Discount = 0.9, Noise = 0.05
Here, I got one positive value. So, tried reducing the noise further more.

Discount = 0.9, Noise = 0.005



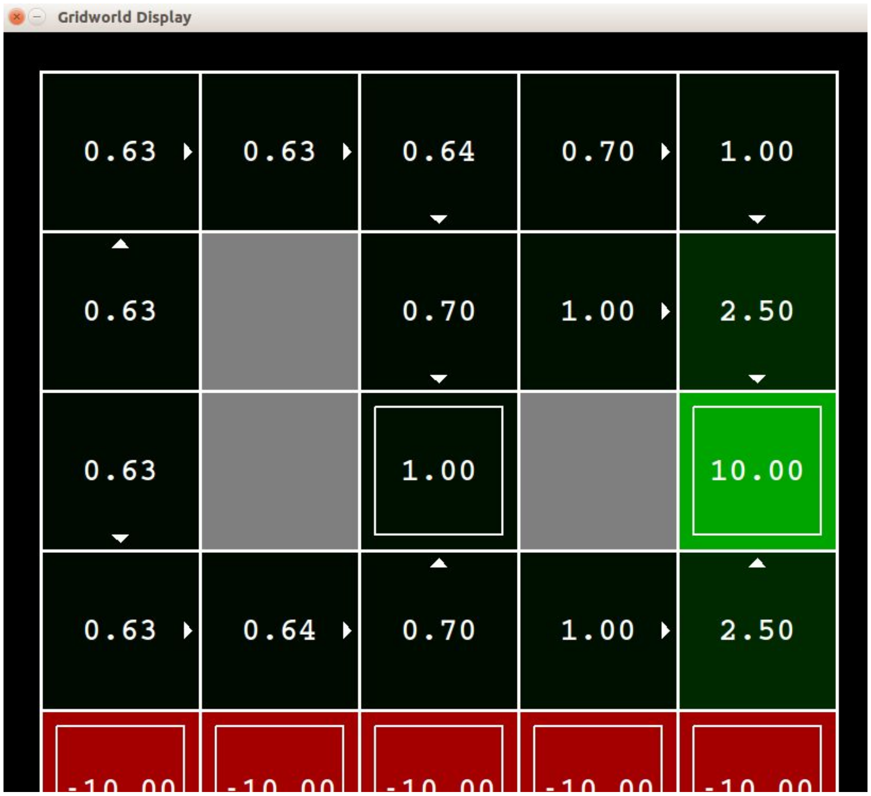
Discount = 0.9, Noise = 0.001

Since, we want the agent to cross the bridge, so we need positive values to move towards the high reward end. We know that on decreasing the noise, the values might increase. The noise refers to the probability of moving in an unintended direction, so, it should be low in this case. Also, the discount factor determines the importance of future rewards. A larger value approaching 1 makes it strive for a long term higher reward.

Q3:
Noise: The probability of moving in an unintended direction.
Discount: The discount factor determines the importance of future rewards
LivingReward: The (negative) reward for exiting "normal" states.

A. **Prefer the close exit (+1), risking the cliff (-10)**

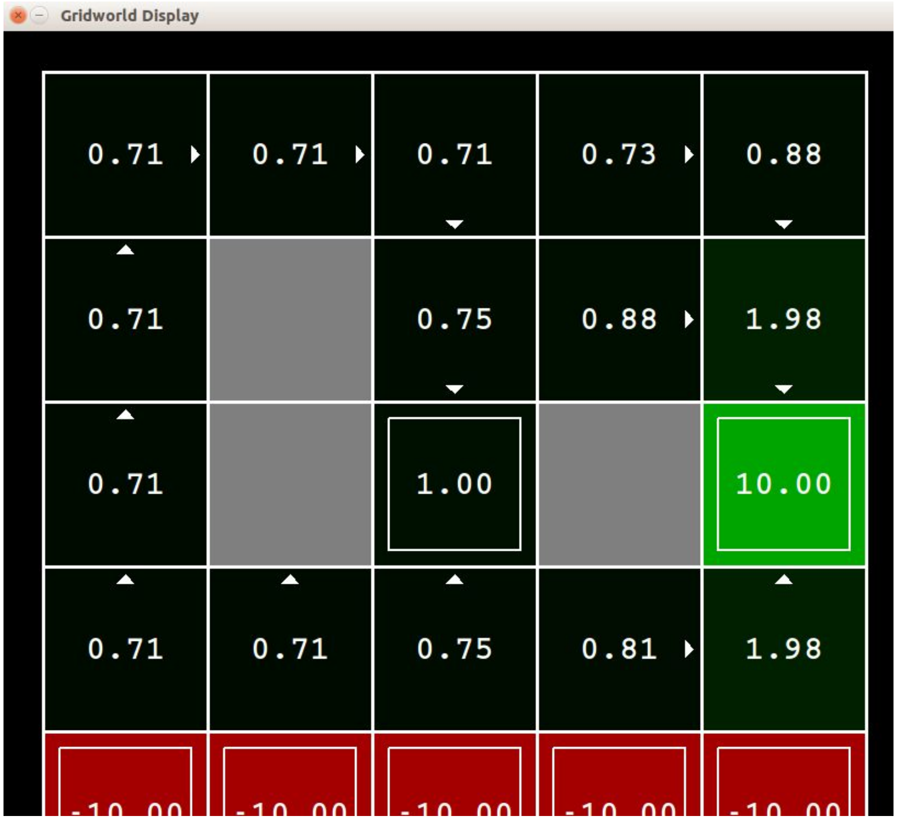
--discount 0.2 --noise 0.001 --livingReward 0.5



Since we want to reach the lower reward state, so we choose a small value of discount as a higher value of discount leads to higher reward. Also, we want to risk the cliff, so we take a average value of livingReward and that does the job for us. The noise value is chosen to be low as we want to avoid the unintended states.

B. **Prefer the close exit (+1), but avoiding the cliff (-10)**

--discount 0.15 --noise 0.09 --livingReward 0.6



Since we want to reach the lower reward state, so we choose a small value of discount as a higher value of discount leads to higher reward. Also, we want to avoid the cliff, so we take a larger value of noise that increases the probability of moving in an unintended direction. Going North is unintended as it is a longer route to reach a positive reward state.

C. Prefer the distant exit (+10), risking the cliff (-10)

--discount 0.4 --noise 0.001 --livingReward 0.6

Since we want to reach the higher reward state, so we try increasing value of discount as a higher value of discount leads to higher reward. Also, we decrease the value of noise to reach the high reward state faster (through the shorter path). Note that we decrease the probability of going to an unintended state this way. The living reward value is kept the same or it can be decreased to reach to high reward state faster.

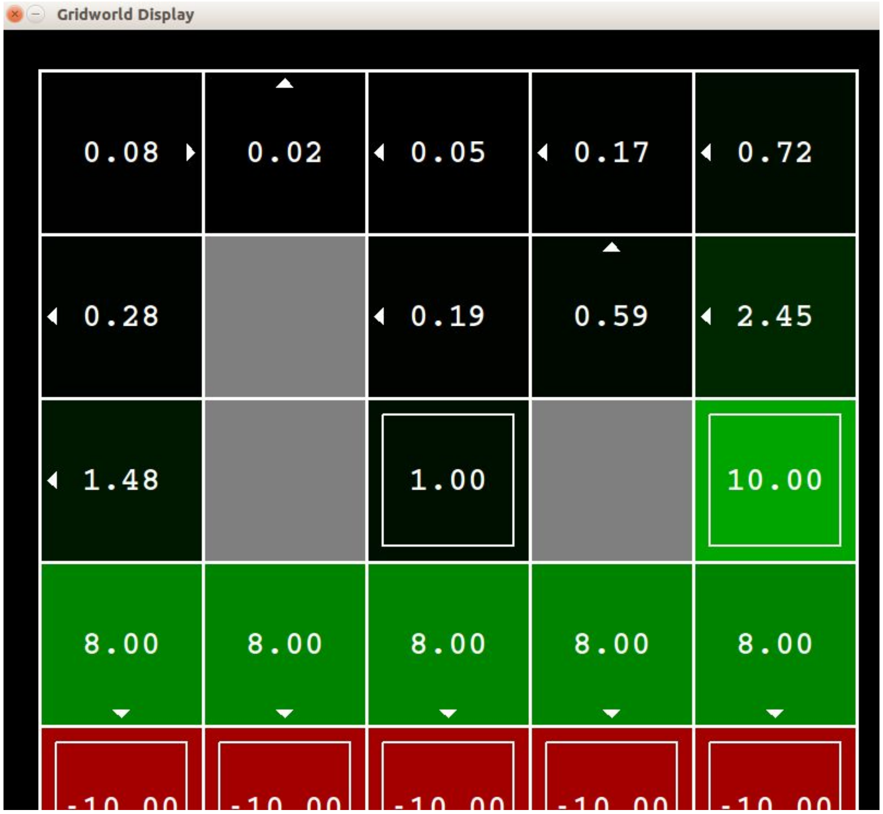
D. Prefer the distant exit (+10), but avoiding the cliff (-10)

--discount 0.7 --noise 0.2 --livingReward 0.6

Since we want to reach the higher reward state, so we try increasing value of discount as a higher value of discount leads to higher reward. Also, we increase the value of noise to reach the high reward state slower (through the longer path). Note that we increase the probability of going to an unintended state this way and so the policy becomes moving to North. The living reward value is kept the same or it can be increased to reach to high reward state slower.

E. Avoid both exits and the cliff (so an episode must never terminate)

--discount 0.1 --noise 5 --livingReward 0.0



The values of discount and living reward close to 0 lead to a random policy generation which does not lead to any of the high reward states. Higher noise so that the transition is to an unintended state.