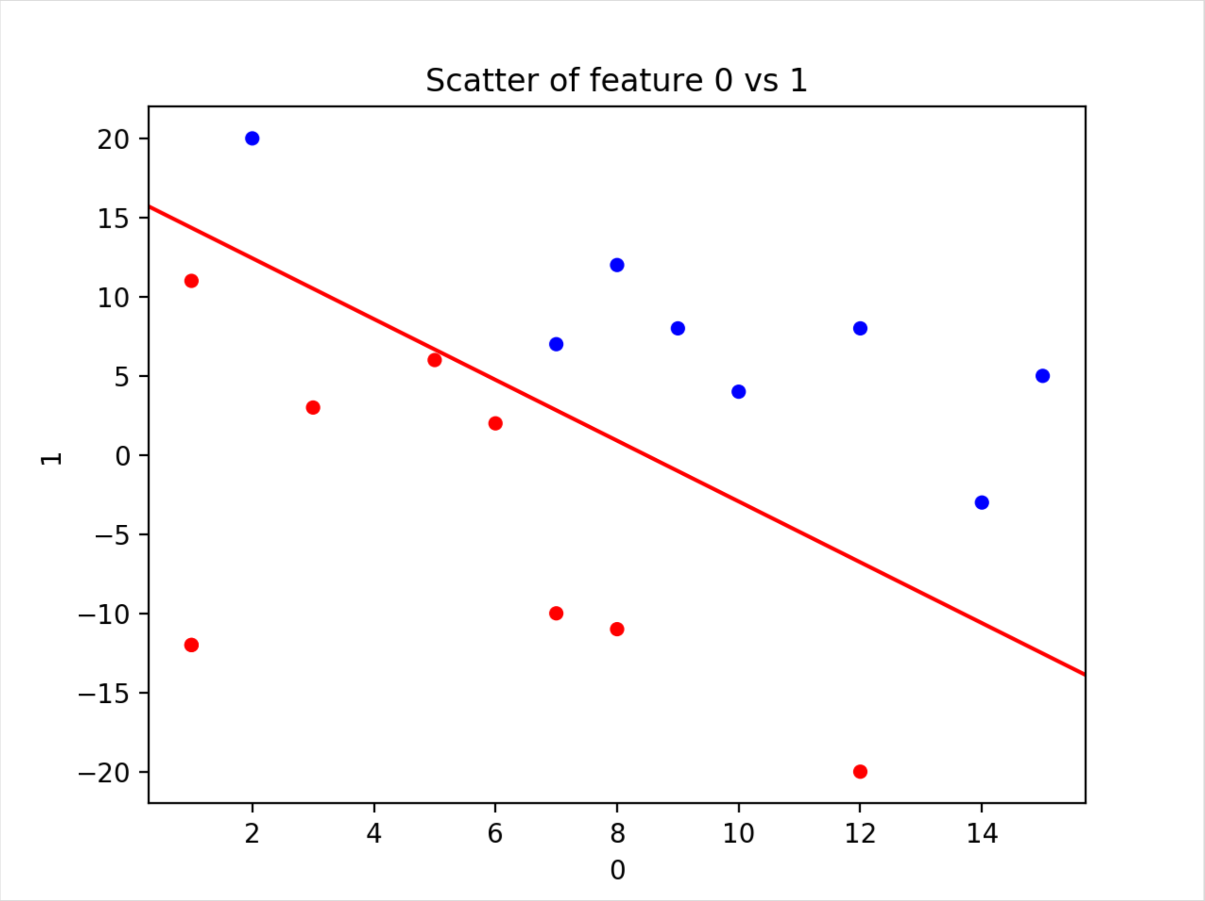
Readme

#Course/Artificial Intelligence#

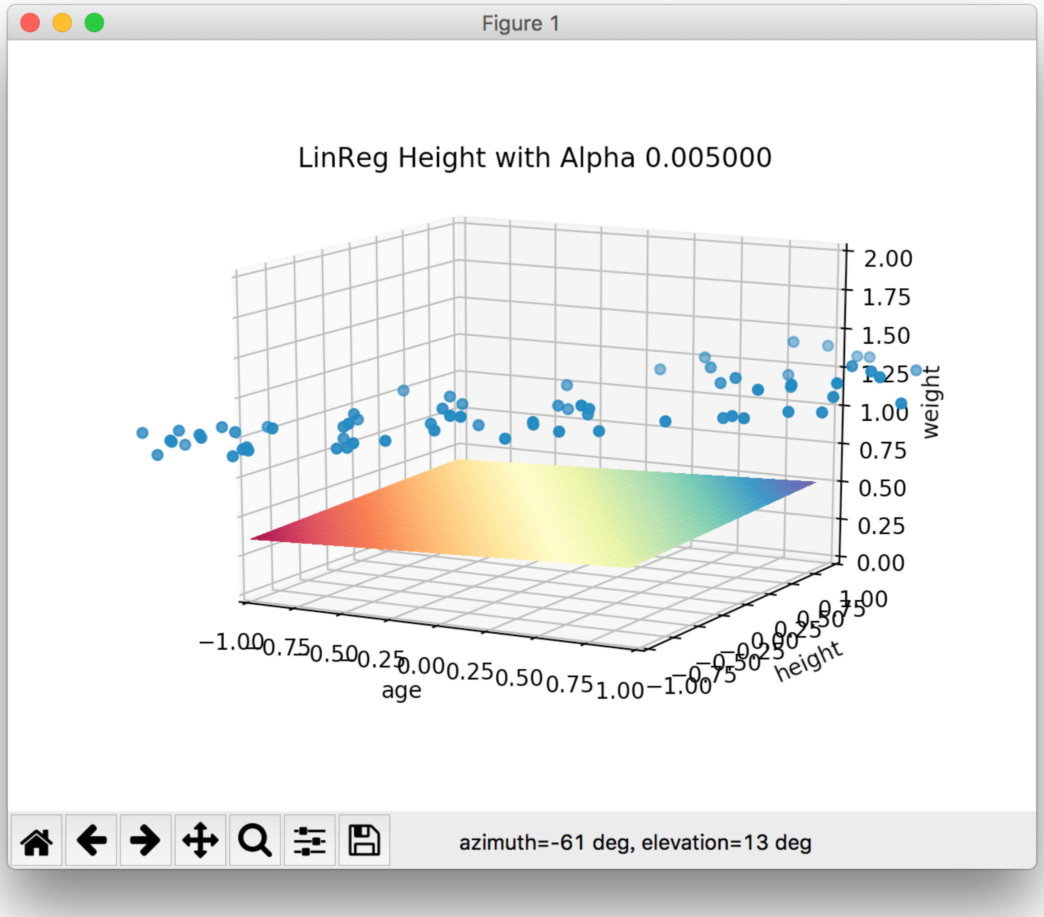
I.Perceptron:

weight_1, weight_2, b = [-5.0,-2.0,39.0]



II.Linear Regression

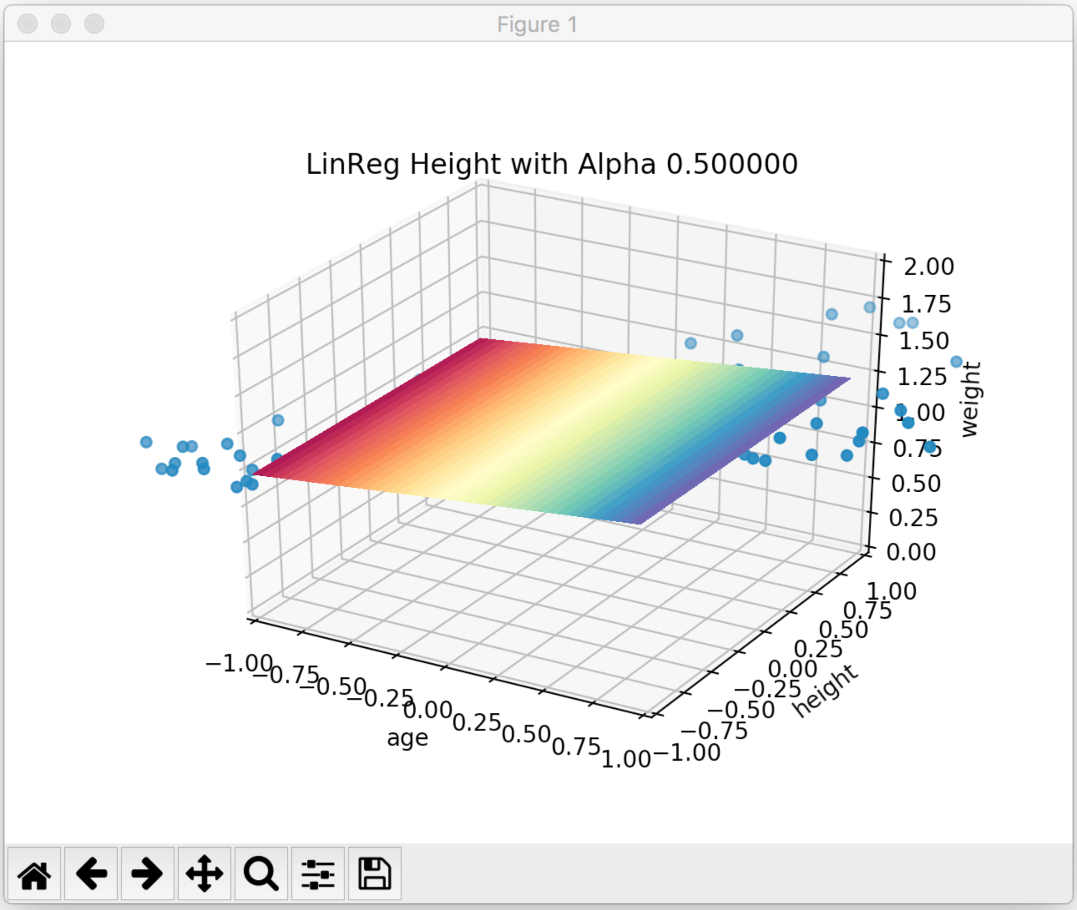
A sample 3d graph when alpha = 0.005, iteraton = 100, bias, b_age, b_weight = [0.4323, 0.0457, 0.0298]



azimuth=-61 deg, elevation=13 deg

My choice:

a, num_iters, bias, b_age, b_weight = [0.5, 170, 1.0964, 0.1286, 0.0014]



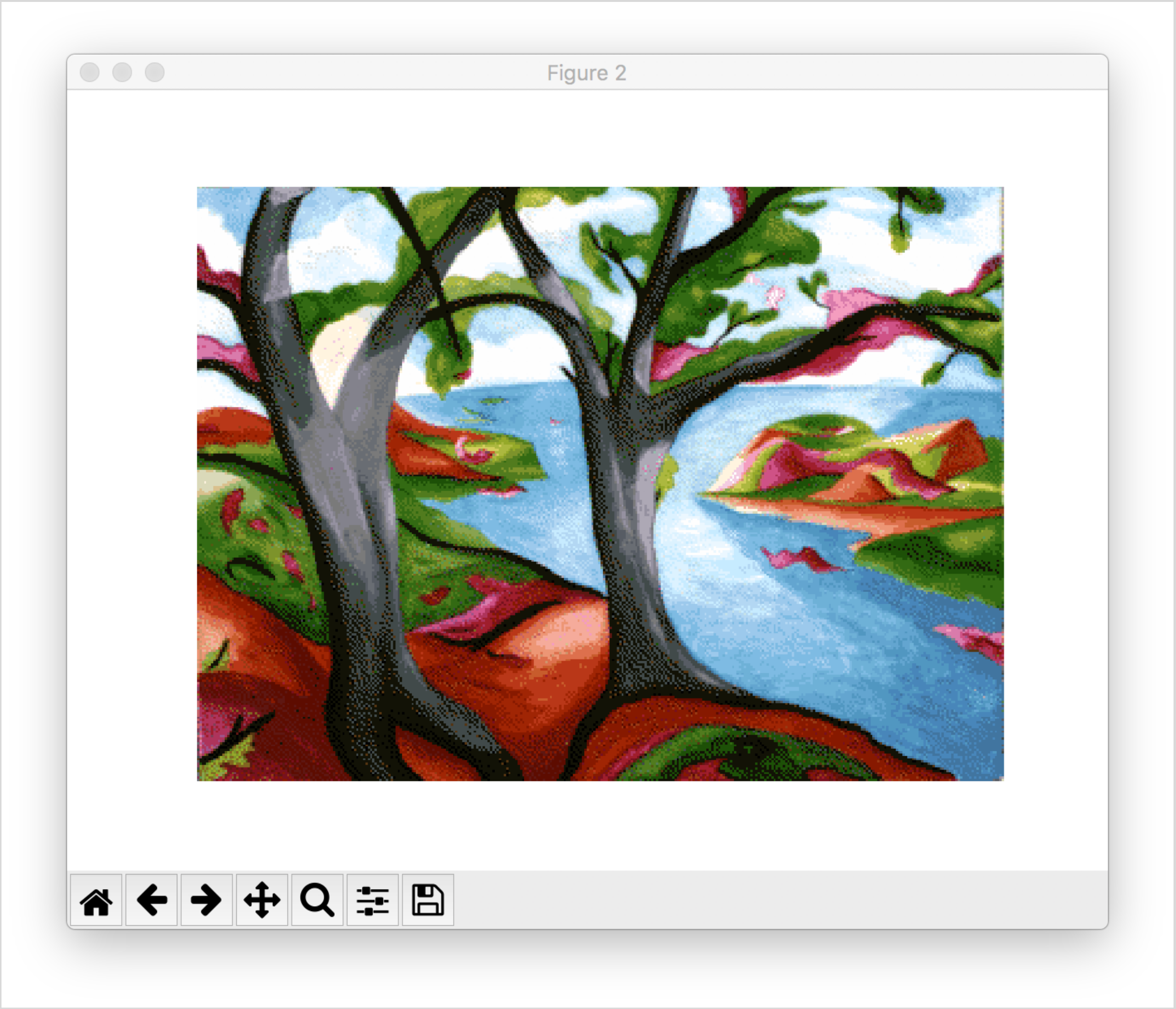
Use iteration over num_iters [from 100 to 1000] and α [from 0.1 to 2.0] to find the best configuration [1.0964,0.1286,,0.0014] which produces minimum loss in this range. This will return the best alpha with the smallest iteration within the range I choose.

- About why choosing α =0.5:
  As we can see from the 9 learning rate in this problem: α ∊ {0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5,10} with 100 iterations, α = 0.5, 1.0 produce better performance with lower loss function, so we should choose a α in this range with more iterations to help it converge.
- About why choosing numbers of iteration = 170:
  170 is the number that make the loss function converge to the minimum value under α = 0.5, so 170 is enough. Adding iterations using gradient descent will cost time without improving performance.

III. Clustering

The 3 representative k value is 64,123,300:

- K = 64
  Sum of squared distances of samples to their closest cluster center = 3768834.82377

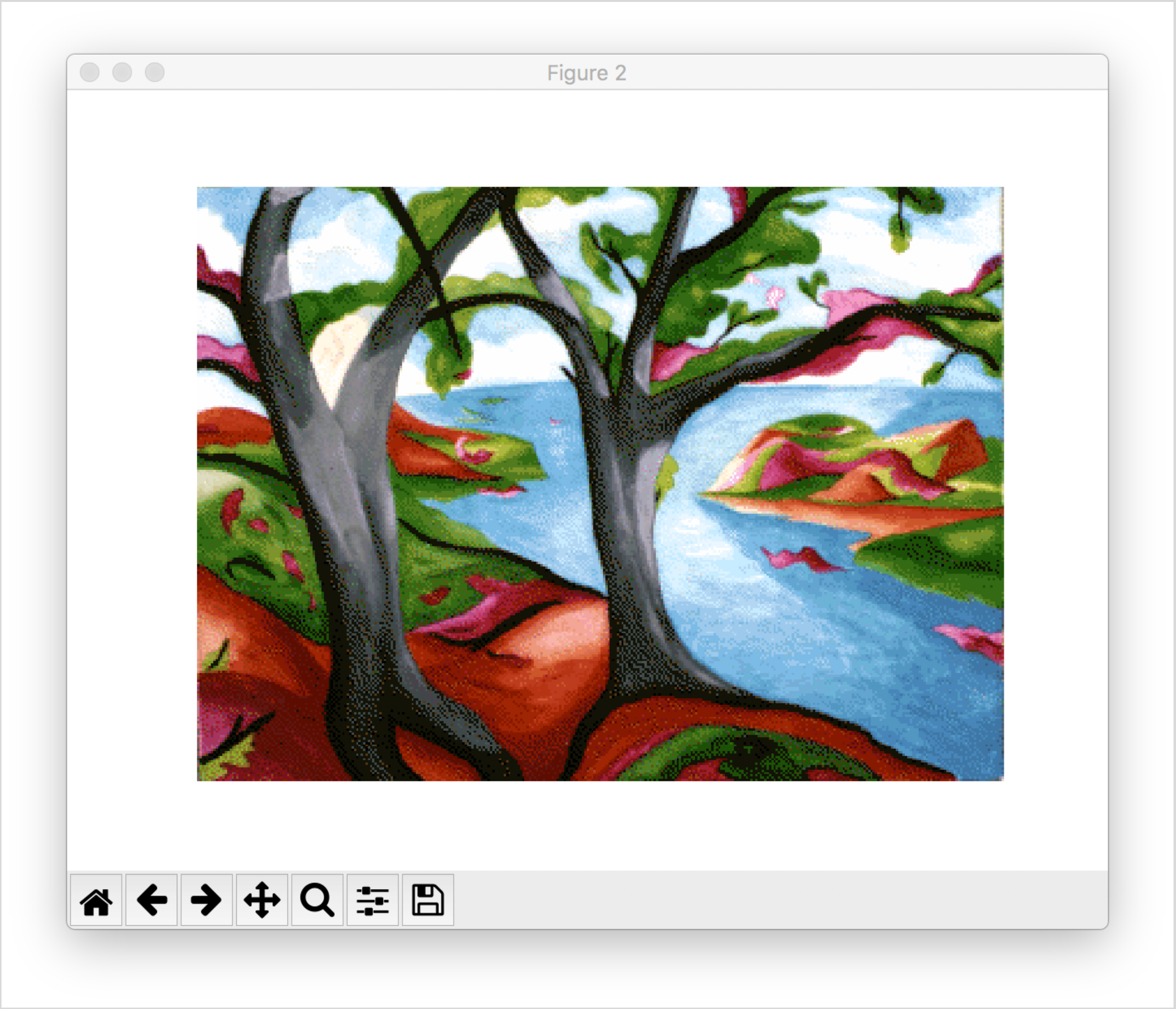


Pros: Quick-—small number of k can significantly save time to get the cluster centers

Cons: The 64 cluster centers cannot well represent all pixel values because the Sum of squared distances of samples to their closest cluster center is too big and the image segmentation reproduced is visually different from the original one: more edges and un-smoothy parts

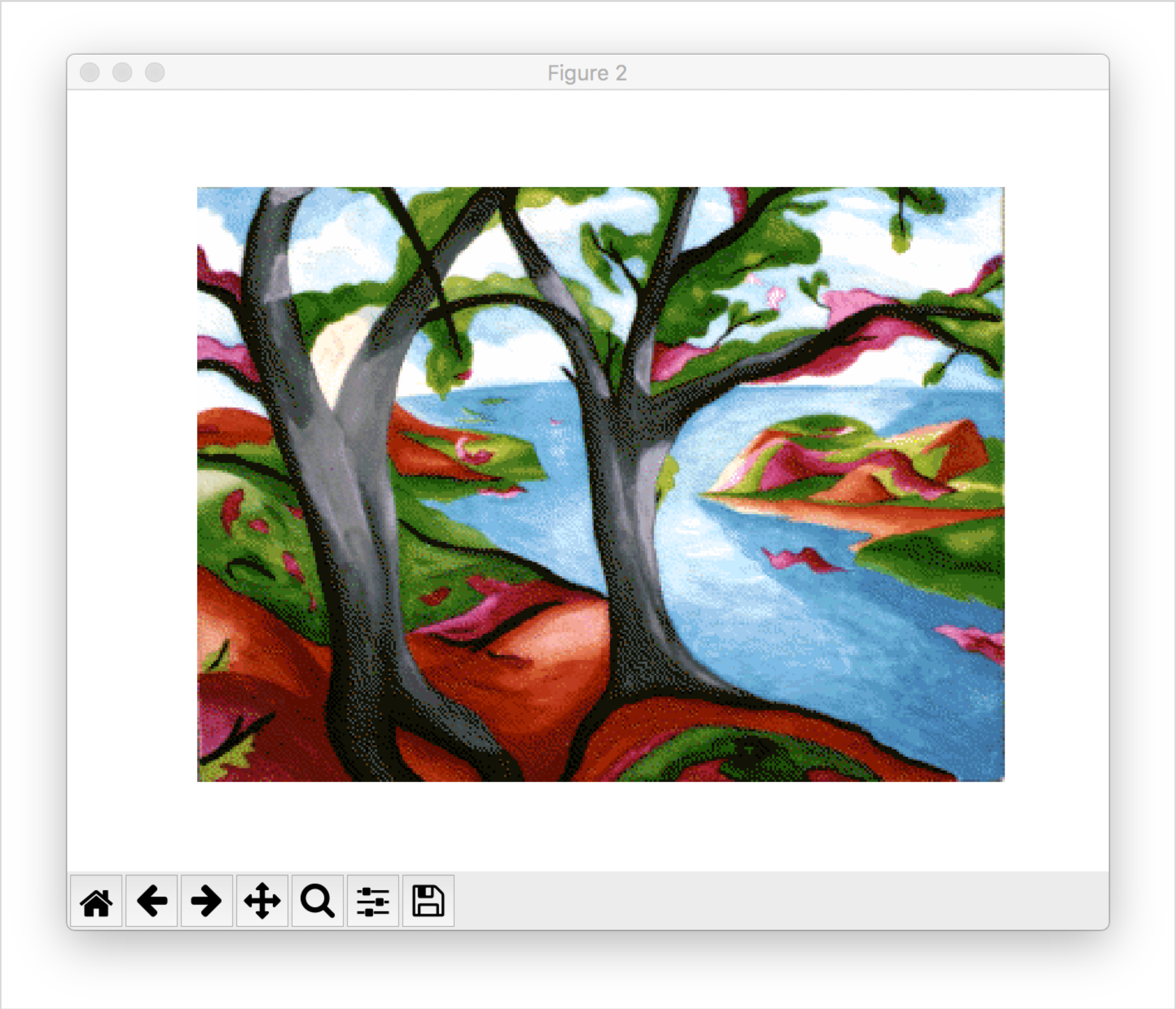
- K= 122

Sum of squared distances of samples to their closest cluster center = 9917.86941581



K = 123

Sum of squared distances of samples to their closest cluster center = 1.61453718003e-18

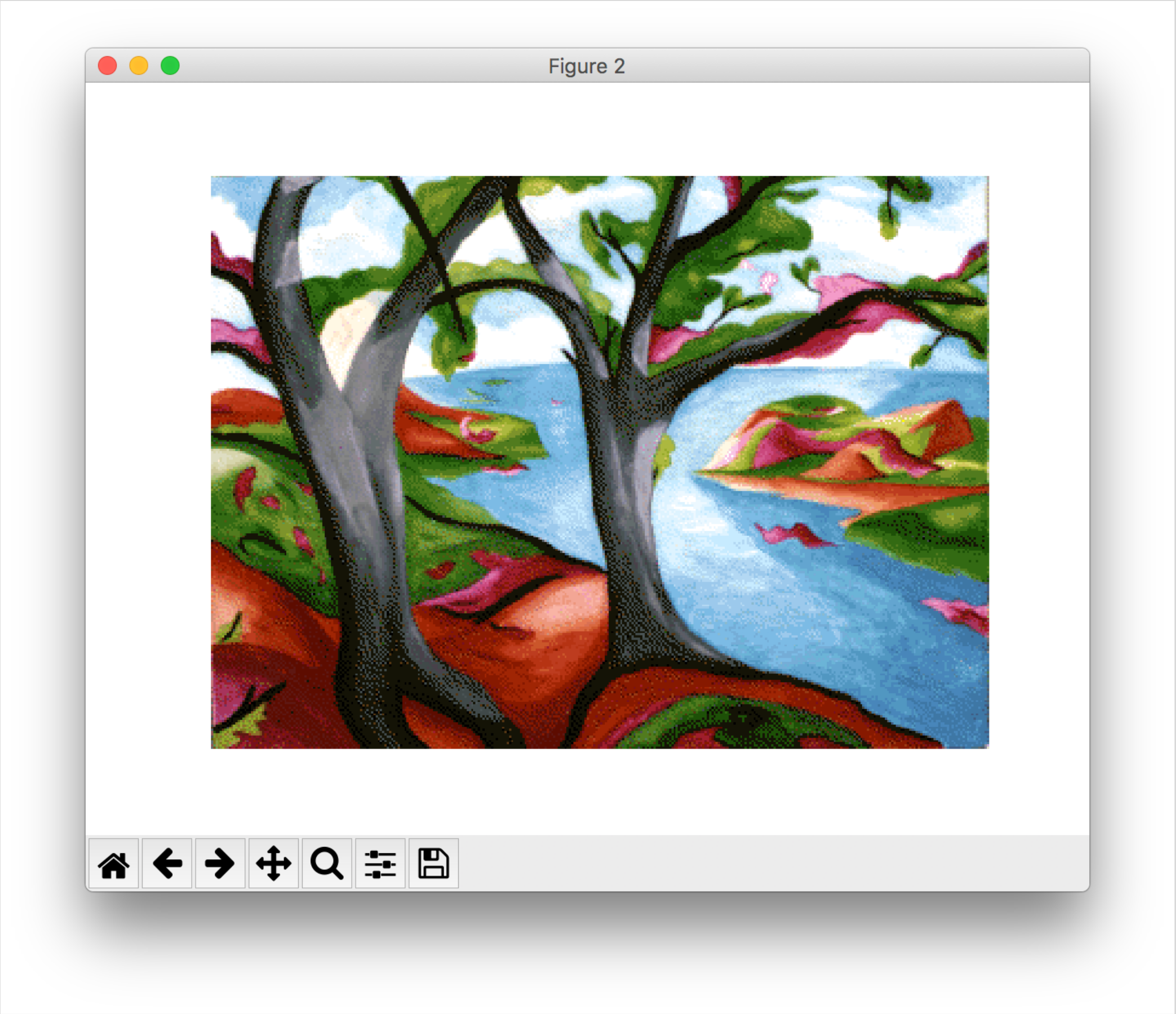


Pros: very little Sum of squared distances of samples to their closest cluster center means 123 clusters can well represent all pixel colors and the reproduced images segmentation is almost the same as original one.

Cons: little more time to compute the clusters

- K= 300

Sum of squared distances of samples to their closest cluster center = 6.91794762962e-19



Pros: smaller Sum of squared distances of samples to their closest cluster center, better representation and segmentation reproduction

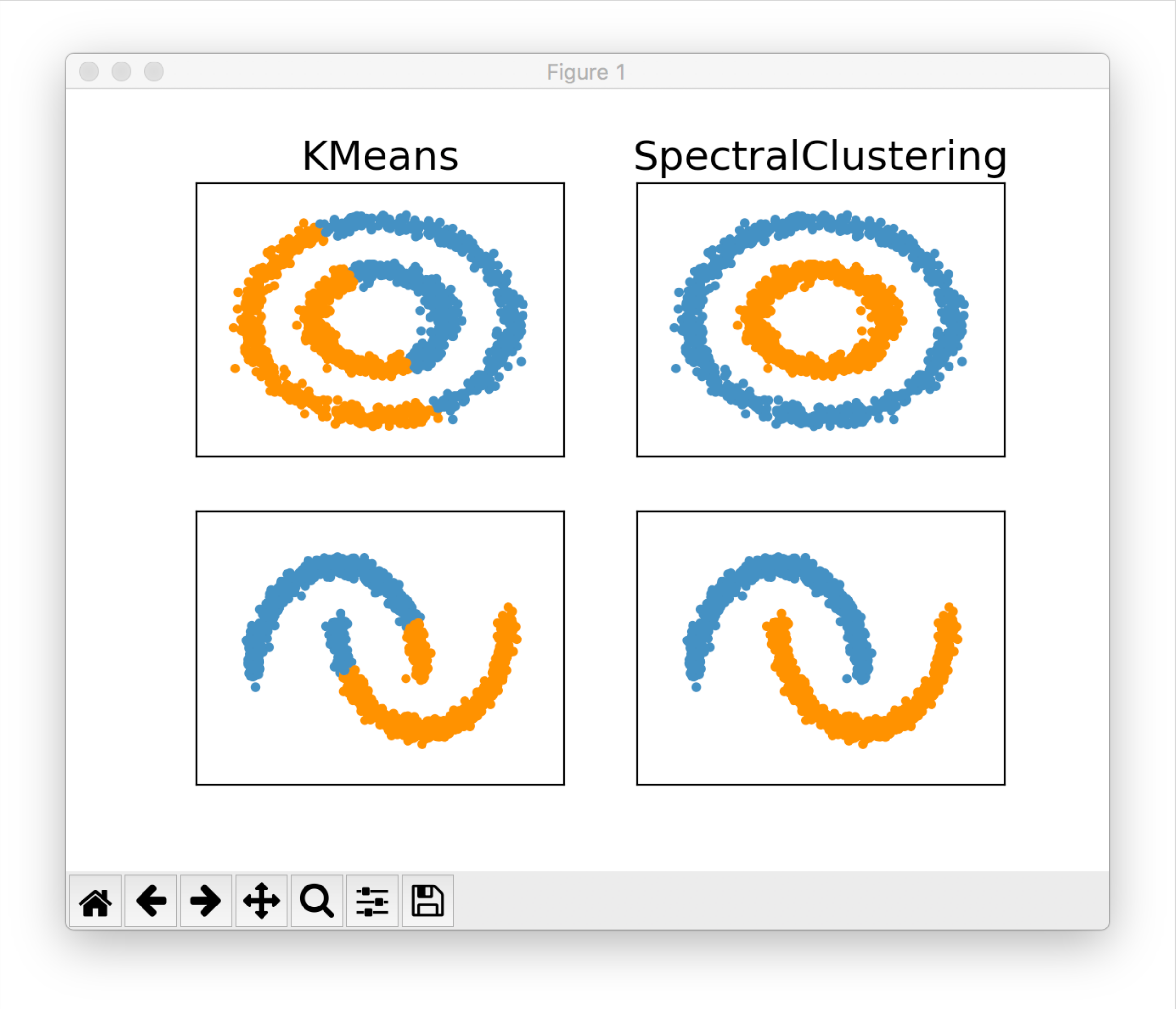
Cons: far more time for computation.

- Conclusion: k =123 is the best because it has very small (Sum of squared distances of samples to their closest cluster centers) and increasing k cannot reduce the error significantly but will cost far more time.

Kmeans_Bonus

The dataset is series of circles/semicircles on same center.

The feature is the x-y position of the points.



Spectral clustering performs far better than kmeans because:

- K-means is ideal for discovering globular clusters where all members of each cluster are in close proximity to each other.
- Spectral clustering forms a similarity matrix where the (i,j) entry is some similarity distance between the i_th and j_th data points in dataset.
- So, spectral clustering is more general because whenever K-means is appropriate for use then so too is spectral clustering. Kmeans use a fixed similarity measurement in the form of Euclidean distance while spectral clustering is more general.