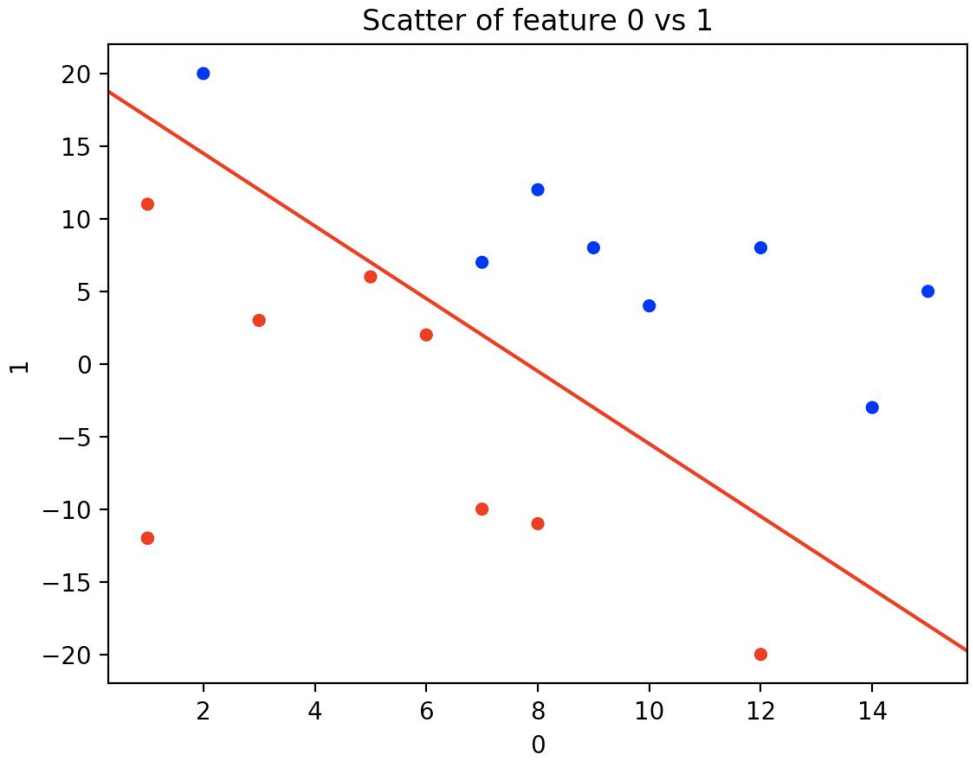
I. Perceptron



The decision boundary is shown above and the command to execute my code using Python3 is:

```
python3 problem1.py input1.csv output1.csv
```

II. Linear Regression

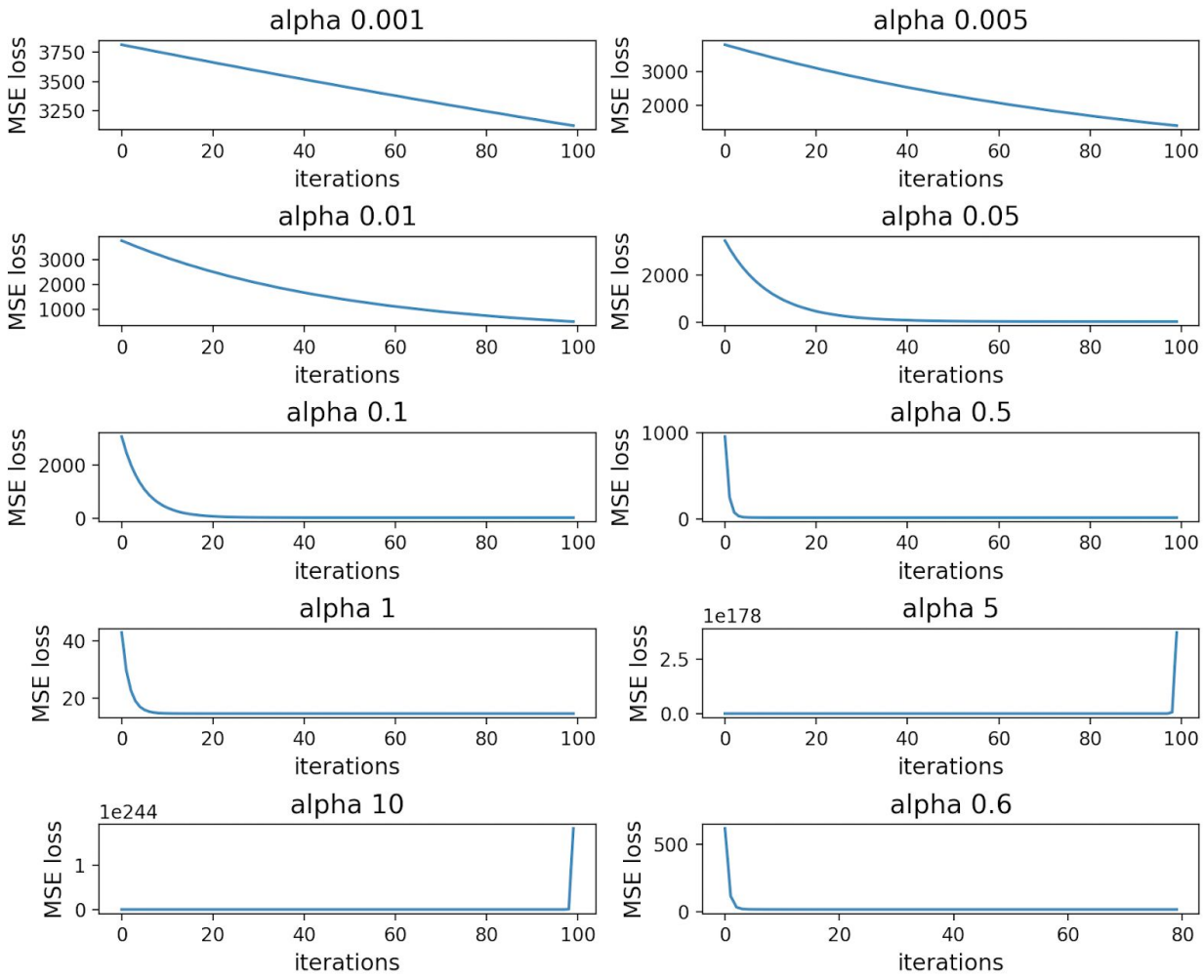
The comparison plots below show the MSE loss versus the number of iterations under different alpha values, and it can clearly illustrates how alpha affects the rate of convergence in the first nine plots using the nine given learning rates and 100 iterations.

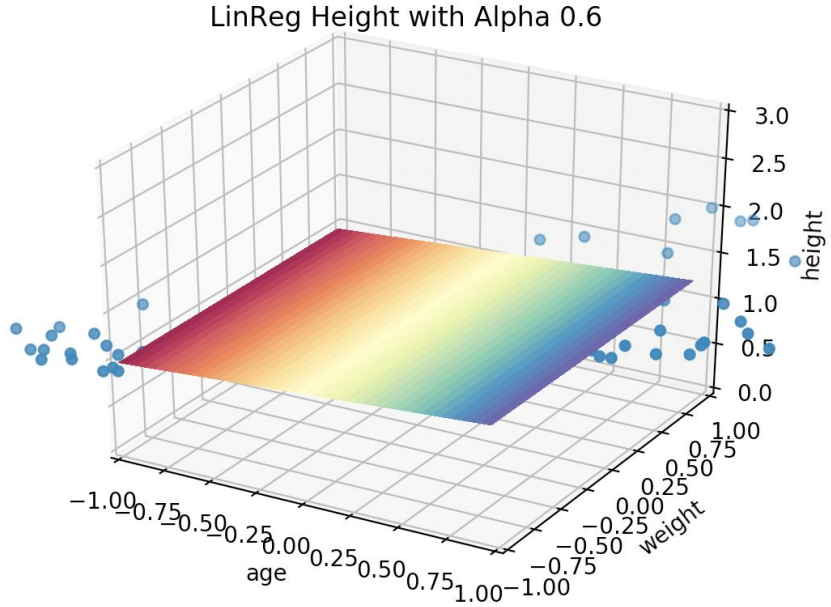
From these plots we can see that with the increase of alpha, MSE converges faster. But when alpha becomes too large (greater than 1), it will cause the step size to be too large and result in unconvergence. Thus, my choice for the tenth rate and number of iterations is 0.6 and 80 because I observe that with alpha in the range of [0.5, 1], the MSE converges fast within 100 iterations with the same stable final MSE value and the results are shown in output2.csv.

The 3d visualization plot with scaled features and decision plane is also shown below.

The command to execute my code using Python 3 is:

```
python3 problem2.py input2.csv output2.csv
```

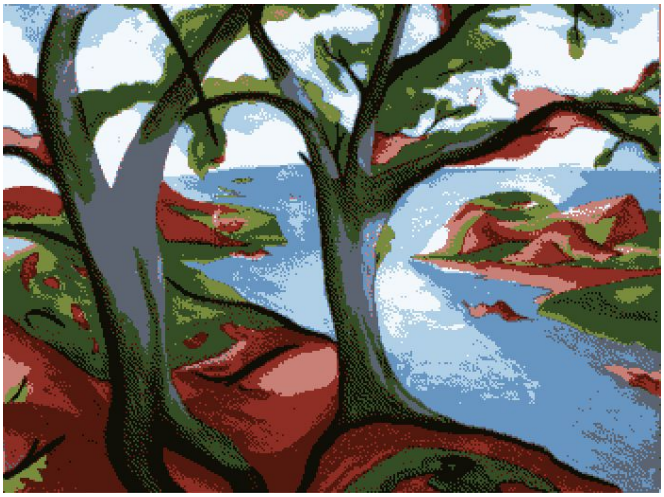

alpha 0.001
alpha 0.005
alpha 0.01
alpha 0.05
alpha 0.1
alpha 0.5
alpha 1
alpha 5
alpha 10
alpha 0.6
MSE loss
iterations
1e178
1e244


LinReg Height with Alpha 0.6
age
weight
height

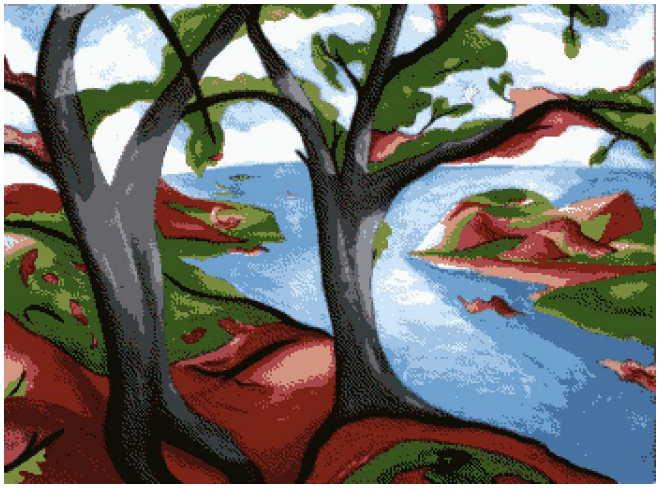
III. K-Means Clustering

k=5

k=10



k=15



The three images above are generated using k-means segmentation with 3 different k values as 5, 10 and 15. We can see from the observation of tree.png that there are roughly 5 colors in general and thus I choose the k value as 5 and then double and triple this k value expecting more accurate segmentation.

Using color(RGB tuple) as characteristic, I use sklearn.cluster.KMeans to find k clusters and generate the according segmentation image by plotting all pixels within a cluster with the cluster center RGB value. We can see from the results above that with the increase of k, the generated images have more segmentations and thus are more alike to the original tree.png.

The command to execute the program is and this program will generate 3 images with the k value as 5, 10 and 15 on the console:

```
python3 problem3.py
```