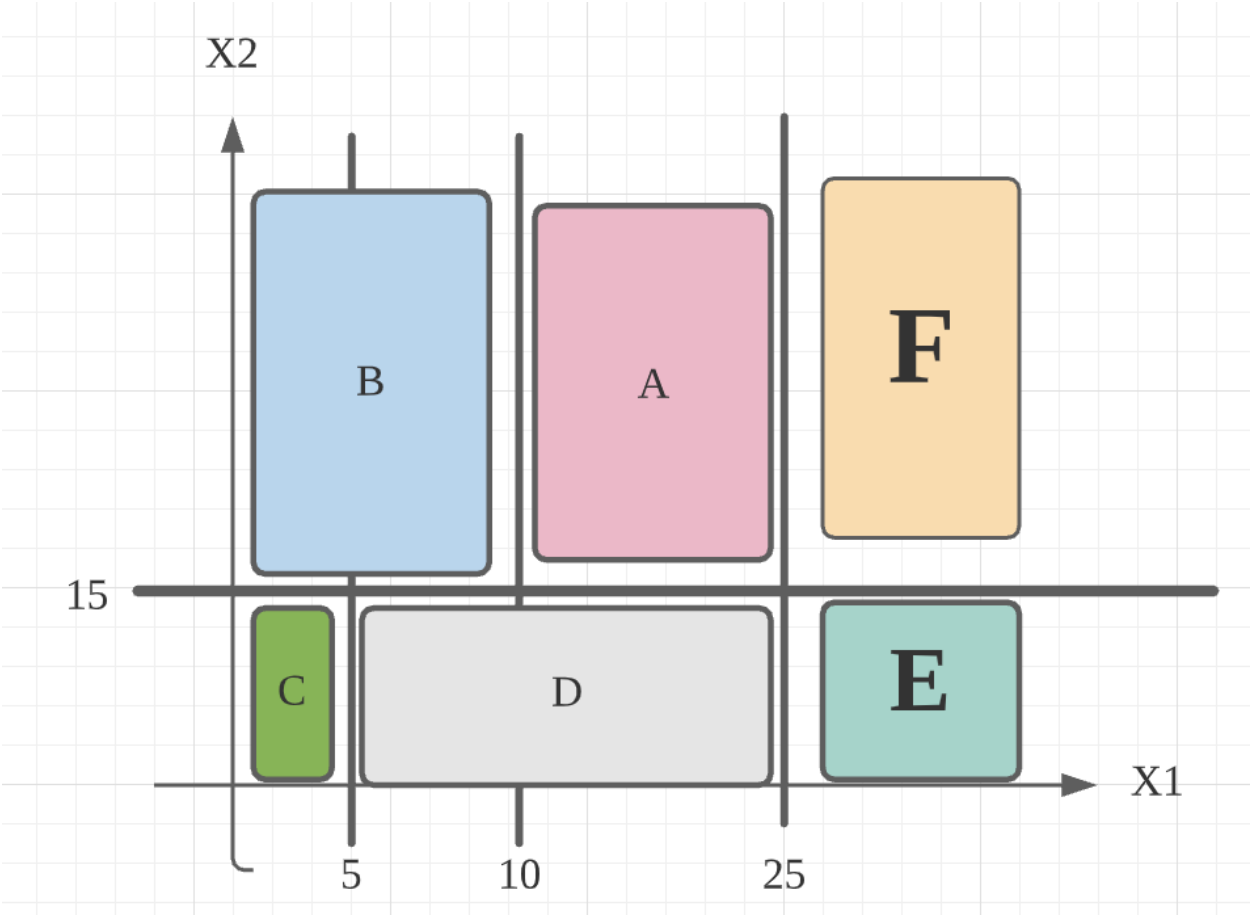
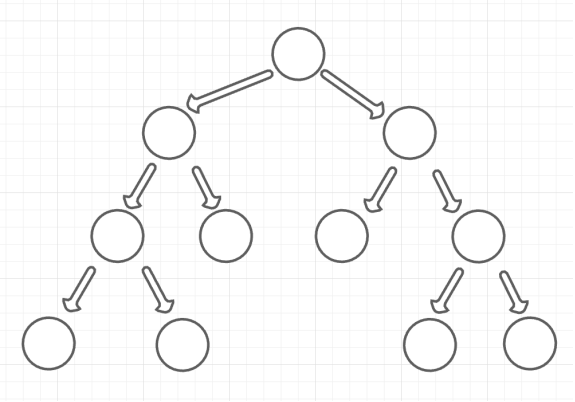
Q1.

a)

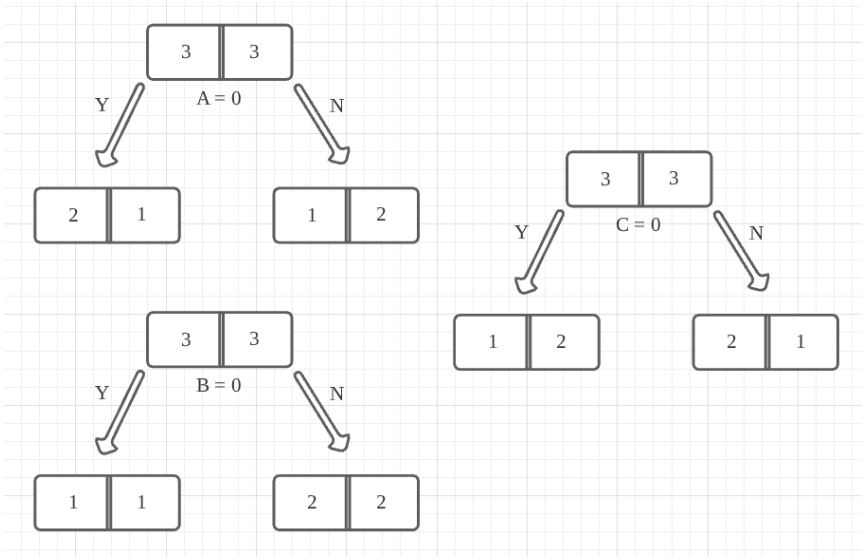




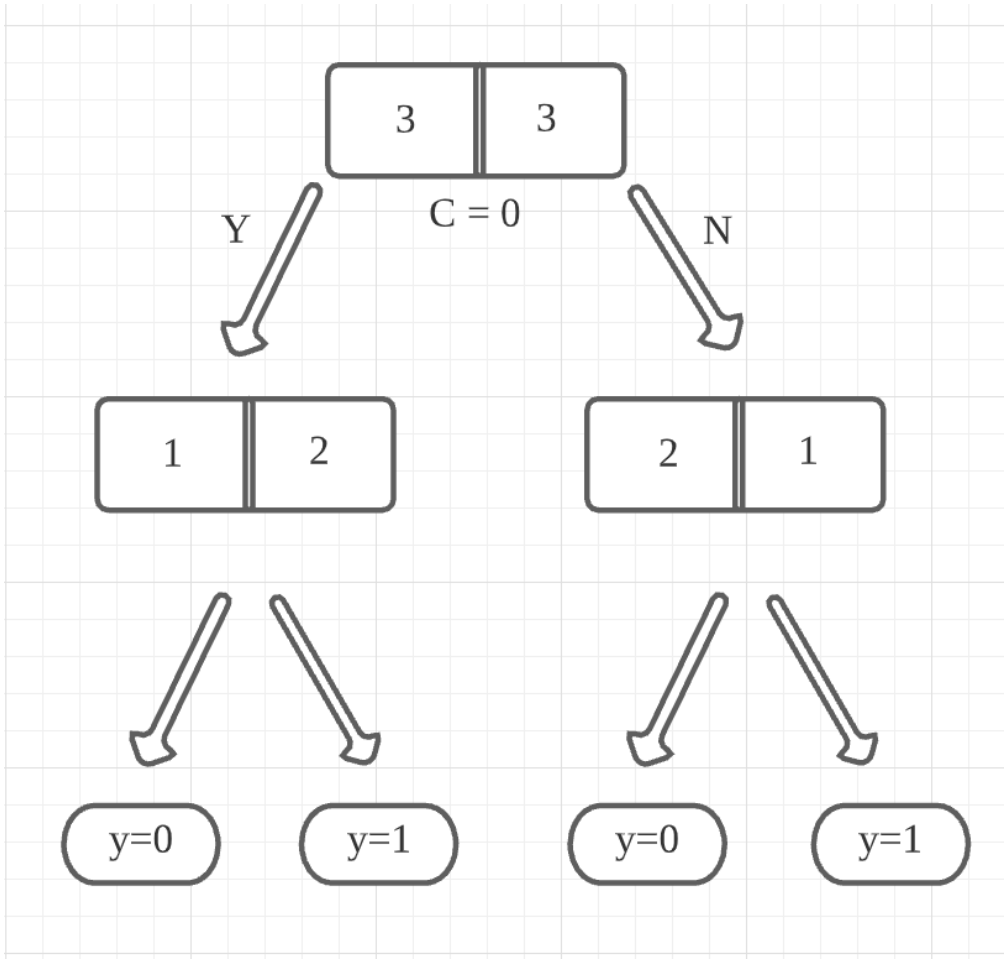
b) For depth-first search methods such as greedy algorithms the ideal search space is the space where all paths are connected to the solution. Redundancy increases the likelihood that arbitrary node expansion sequences lead to good trees.

c) These redundant trees are not a statistical problem, because it has no effect on the expressive power of decision trees.

Q2. Learn a Decision tree from the training set given char shows that



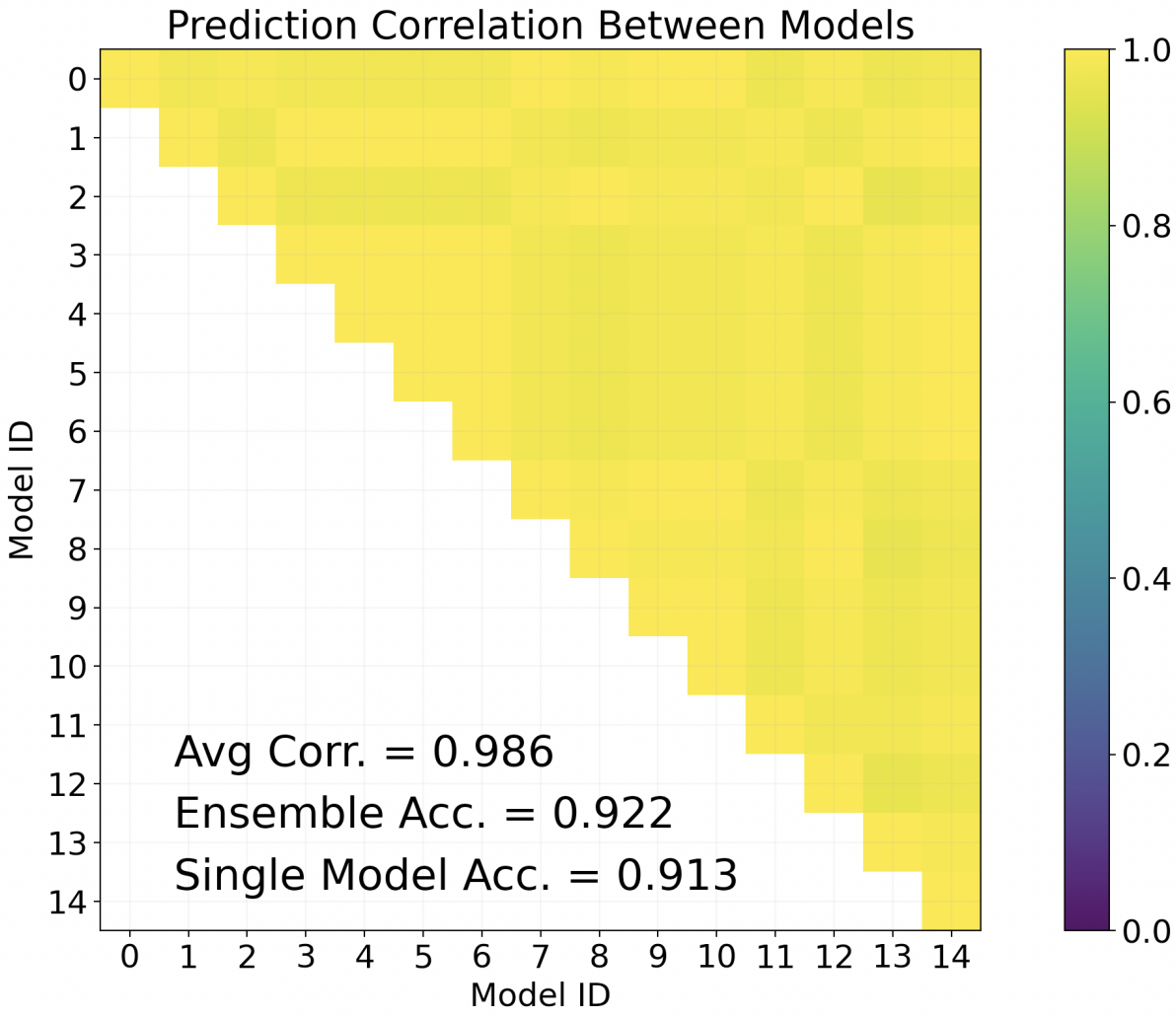
Here is final decision tree



Q3.

a) Apply bagging by uniformly sampling train data points with replacement to train each ensemble member.

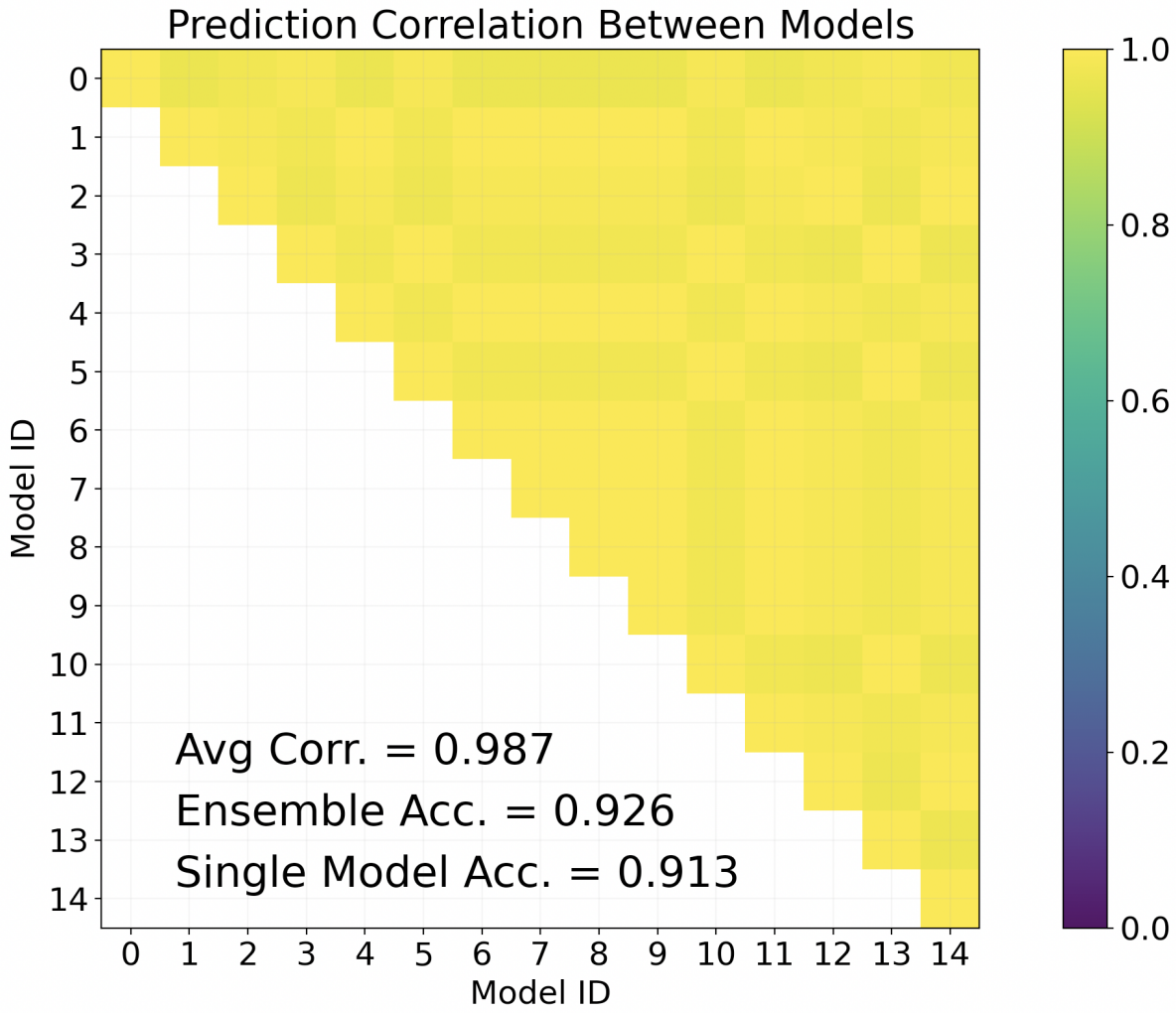
```
data_index = [data_index for data_index in range(X_train.shape[0])]
random_data_index = np.random.choice(data_index, X_train.shape[0], replace=True)
```



I tried to set the index number on each model vector set and choose randomly to read the vector set in difference.

b) When set the hyperparameter as “max_depth = 5”

```
clf = tree.DecisionTreeClassifier("entropy",max_depth = 5) # b
clf = clf.fit(X_data, y_data)
preds[:,m]= clf.predict(X_test)
```



Q4.

a)

```
def initalizeCentroids(dataset, k):
  idx = np.random.permutation(dataset.shape[0])
  centroids = dataset[idx[:k]]
  # print("This is centroids: ", centroids)
  # raise Exception('Student error: You haven\'t implemented initializeCentroids yet.')
  return centroids
```

b)

```python
def computeAssignments(dataset, centroids):

  # How do I get num_clusters from outside
  clusters = centroids.shape[0] #k = j
  distance = np.zeros((dataset.shape[0], clusters))
  for k in range(clusters):
    dist = np.linalg.norm(dataset - centroids[k], axis=1)
    distance[:,k] = np.square(dist)
  # raise Exception('Student error: You haven\'t implemented computeAssignments yet.')
  # print(np.shape(assignments))
  # print("this is distance: ", distance)

  assignments = np.argmin(distance, axis=1)
  # print("This is shape of assignments: ", np.shape(assignments)) # n * 1

  return assignments
```

c)

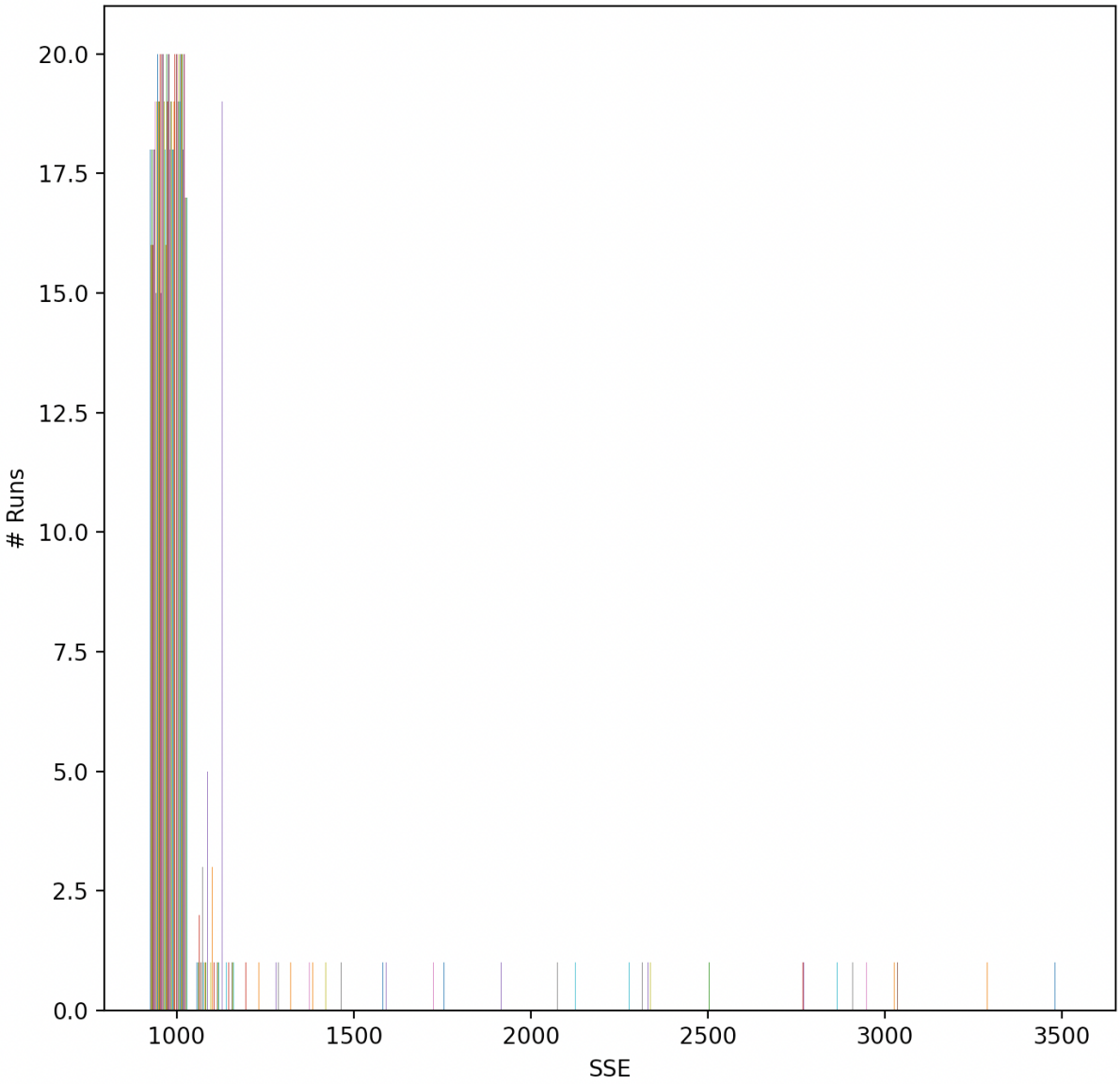
```python
j = 0
for j in range(k):
  # print("This is j in for statement: ", j)
  # print(assignments == j)
  centroids[j] = np.mean(dataset[assignments == j], axis=0)
  counts.append(np.count_nonzero(dataset[assignments == j]))
```

d)

```python
def calculateSSE(dataset, centroids, assignments):

  # print("This is SSE centroids: ", centroids)
  # print("This is SSE assignments: ", assignments)
  clusters = centroids.shape[0]
  # sse = np.zeros((dataset.shape[0], clusters))
  sse = []
  for k in range(clusters):
    sse_get = np.linalg.norm(dataset[assignments == k] - centroids[k], axis=0)
    sse_sum = np.sum(sse_get)
    sse_square = np.square(sse_sum)
    sse.append(sse_square)

  sse = np.sum(sse)
  # print("This is sse: ", sse)
  return sse
```

Q5)

```
for a in range(50):
  centroids, assignments, sse_t = kMeansClustering(X, k=k, max_iters=max_iters, visualize=False)
  SSE_rand.append(sse_t)

plotClustering(centroids, assignments, X, title="Second Clustering")
```



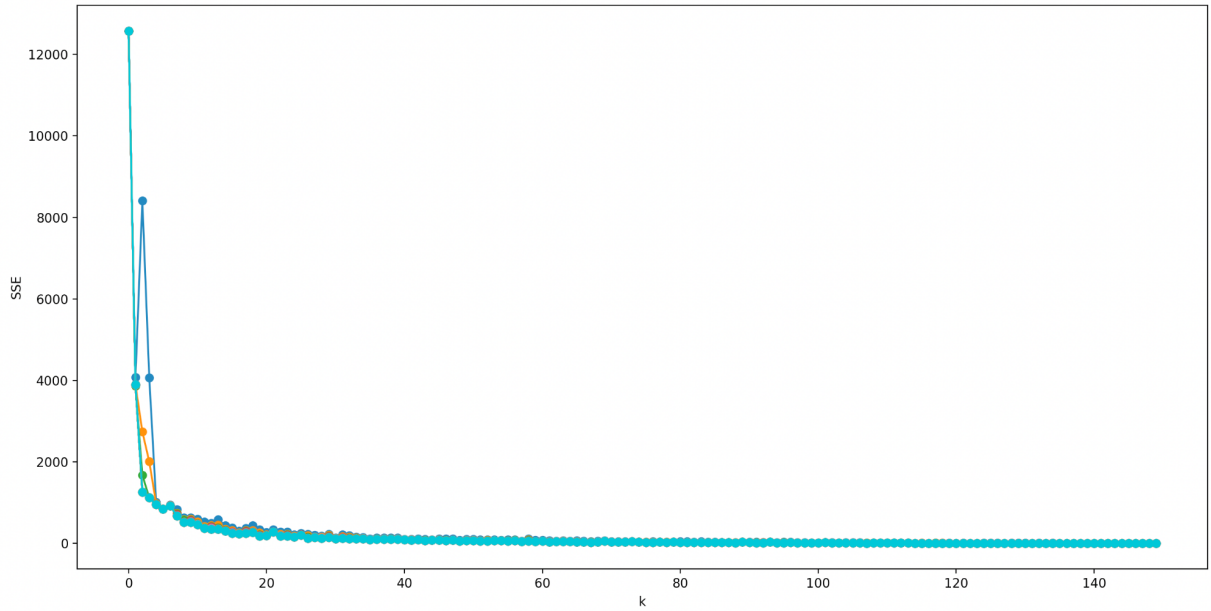
Provide the resulting plot and discuss how this might affect how you apply k-means to a real dataset.

- When iteration goes increasing then k-means's SSE will decrease, which means that running many running iterations gives generally better results than a small iteration.

Q6)

```
k = 1
for a in range(150):
  k = a + 1
  centroids, assignments, sse_t = kMeansClustering(X, k=k, max_iters=max_iters, visualize=False)
  SSE_vs_k.append(sse_t)
```

Provide the resulting plot and discuss why choosing k based on the sum-of-squared error doesn't make sense.

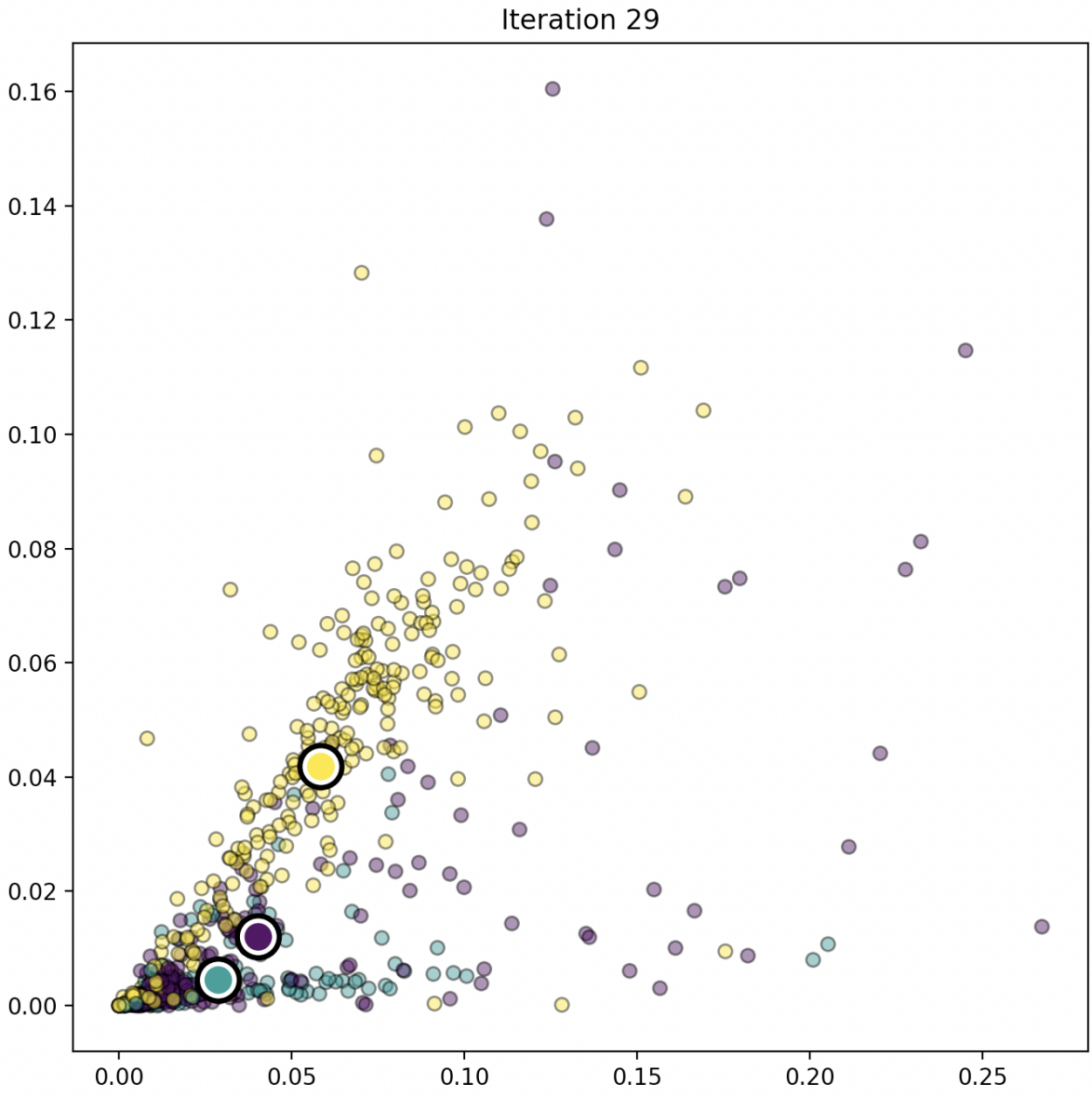


The k-means algorithm requires a set of values for k to be clustered. In general, the variance within a cluster is small, and the larger the variance between clusters, the better. This indicates that the smallest sum of squared error (sse), the center points of each cluster, are well characterized by multiple runs of k-means on the same data. As a result, which means are not implied as K based on the SSE does not make sense.

Q7)

a) There are three big kinds of groups on the dataset that consist of trees, buildings, and roads. I think setting k is equal to 10 is not good enough to distinguish between images which means too high values.
This is because the characteristics of each image seemed to not be well distinguished and hard to distinguish when clustering was distinguished by 10 features.

b) I think k = 3 is good balance clustering on this problem



c) SSE is not an indicator of the clustering qualities. Even with a clustering result has a high "SSE" value, the utilization of the cluster can be used by human judgment.

K = 10

```
This is k = {{k}}'s sse 10 [7425139.174069157, 7022292.992992168, 6817410.136609184, 6773853.383776075, 6733547.5010894695, 6699922.958163977,
6657299.9800862195, 6639203.351538355, 6634523.285041547, 6621475.926290478, 6607737.961866474, 6598388.138576403, 6593549.441883461, 6585470.2
13255816, 6583672.963728916, 6579224.289340409, 6574145.449957849, 6571299.0566853415, 6571150.196199233, 6570950.800701439, 6570744.888088513,
 6571078.667195292, 6565023.898285216, 6563170.49552673, 6561525.721364669, 6561067.366951551, 6560304.952802133, 6560304.952802133, 6560304.95
2802133, 6560304.952802133]
Cluster 0 [36]
Cluster 1 [15]
Cluster 2 [39]
Cluster 3 [54]
Cluster 4 [167]
Cluster 5 [84]
Cluster 6 [69]
Cluster 7 [76]
Cluster 8 [66]
Cluster 9 [39]
```

K = 3

```
This is k = {{k}}'s sse 3 [9807020.548321165, 8196862.3620492015, 7957043.911945285, 7912477.132646165, 7902009.140855674, 7889651.337208735, 7
892242.240397999, 7891576.738645298, 7891576.738645298, 7891576.738645298, 7891576.738645298, 7891576.738645298, 7891576.738645298, 7891576.738
645298, 7891576.738645298, 7891576.738645298, 7891576.738645298, 7891576.738645298, 7891576.738645298, 7891576.738645298, 7891576.738645298, 78
91576.738645298, 7891576.738645298, 7891576.738645298, 7891576.738645298, 7891576.738645298, 7891576.738645298, 7891576.738645298, 7891576.7386
45298, 7891576.738645298]
Cluster 0 [163]
Cluster 1 [263]
Cluster 2 [219]
```

Additionally when K=100, in this case, dividing clustering was too detailed to distinguish the characteristics by pictures.

```
This is k = {{k}}'s sse 100 [4527274.571256993, 4282688.151193015, 4182166.4999030028, 4143623.8480637004, 4134445.807997218, 4132385.650867929
, 4131510.5083690165, 4130497.8181362525, 4129872.8416588847, 4128400.9001312633, 4128400.9001312633, 4128400.9001312633, 4128400.9001312633, 4
128400.9001312633, 4128400.9001312633, 4128400.9001312633, 4128400.9001312633, 4128400.9001312633, 4128400.9001312633, 4128400.9001312633, 4128
400.9001312633, 4128400.9001312633, 4128400.9001312633, 4128400.9001312633, 4128400.9001312633, 4128400.9001312633, 4128400.9001312633, 4128400
.9001312633, 4128400.9001312633, 4128400.9001312633]
Cluster 0 [3]
Cluster 1 [6]
Cluster 2 [11]
Cluster 3 [8]
Cluster 4 [1]
Cluster 5 [6]
Cluster 6 [3]
Cluster 7 [1]
Cluster 8 [2]
Cluster 9 [3]
Cluster 10 [3]
Cluster 11 [2]
Cluster 12 [1]
Cluster 13 [1]
Cluster 14 [2]
Cluster 15 [1]
Cluster 16 [6]
Cluster 17 [21]
Cluster 18 [3]
Cluster 19 [3]
Cluster 20 [1]
Cluster 21 [2]
Cluster 22 [4]
Cluster 23 [11]
Cluster 24 [2]
Cluster 25 [3]
```

Q8.
Suppose after you finish clustering, you want to assign a name to each cluster corresponding to what is in them.

1. Provide a “label” for each of your clusters.

Labeled as Including tree, Including building, Including Road.

< Seems Natural Forest >

< Seems building >

< Seems Road >

2. Estimate the purity of your clusters by counting the number of examples in the plots that seem to violate your labelling, divided by the total number of images shown.

i) Approximately 90% of purity of Label 1 (approximately 10% violate)
ii) Approximately 98% of purity of Label 2 (approximately 2% violate)
iii) Approximately 88% of purity of Label 3 (approximately 12% violate)

3.

1. Approximately how many hours did you spend on this assignment? **7 days**

2. Would you rate it as easy, moderate, or difficult? **moderate**

3. Did you work on it mostly alone or did you discuss the problems with others? **Alone**

4. How deeply do you feel you understand the material it covers (0%–100%)? **90%**

5. Any other comments? **N/A**