ARDA ANDIRIN 17921934994

CMPE-442 Take-Home Final Assignment

Q1)
Similarity: Both PCA and feature selection tries to take the meaningful, valued data into the consideration, so in way both tires to reduce the complexity.

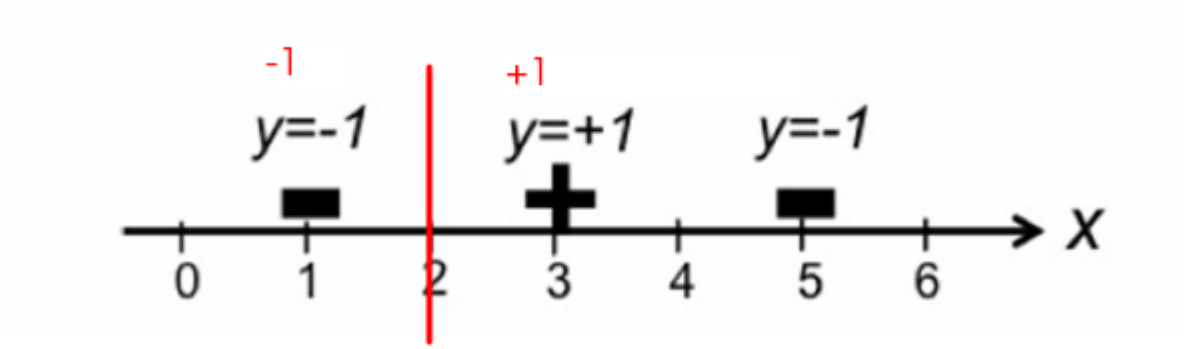
Difference: With feature selection we select important features. But with PCA we take the most important data from all the features.

Q2) k-means algorithm is an unsupervised learning problem. There are k cluster centers in the data and these centers move to a new point on the dataset with each iteration of distance calculation. The goal is to make these centers on the point so that there is a cluster center in the middle point of every cluster.

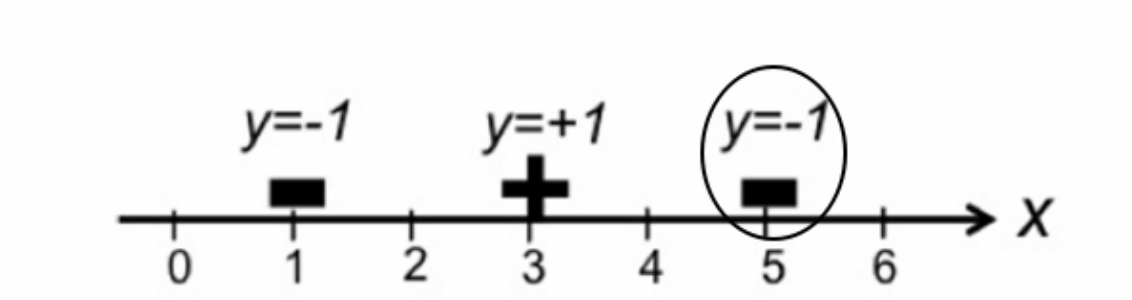
In the k-nn algorithm. A new data is presented to the dataset. Depending on the number of k, the closest k points' classes are sorted. And, the new data is assigned to the most common class among these neighbours.

So, k-means is an unsupervised learning algorithm while K-nn is a supervised learning algorithm. K means is used for clustering, k-nn is used for classification. K-means doesn't need labeled points but K-nn needs labeled points. K-means is usually used for things such as visualizing trends in social media or demographics of population while K-nn is used for classification of data where the target attribute is already known before hand.

Q3)
a) Since there are 3 samples, m=3. So, each data point has the weight 1/m. Hence 1/3 or 0.333.
b)



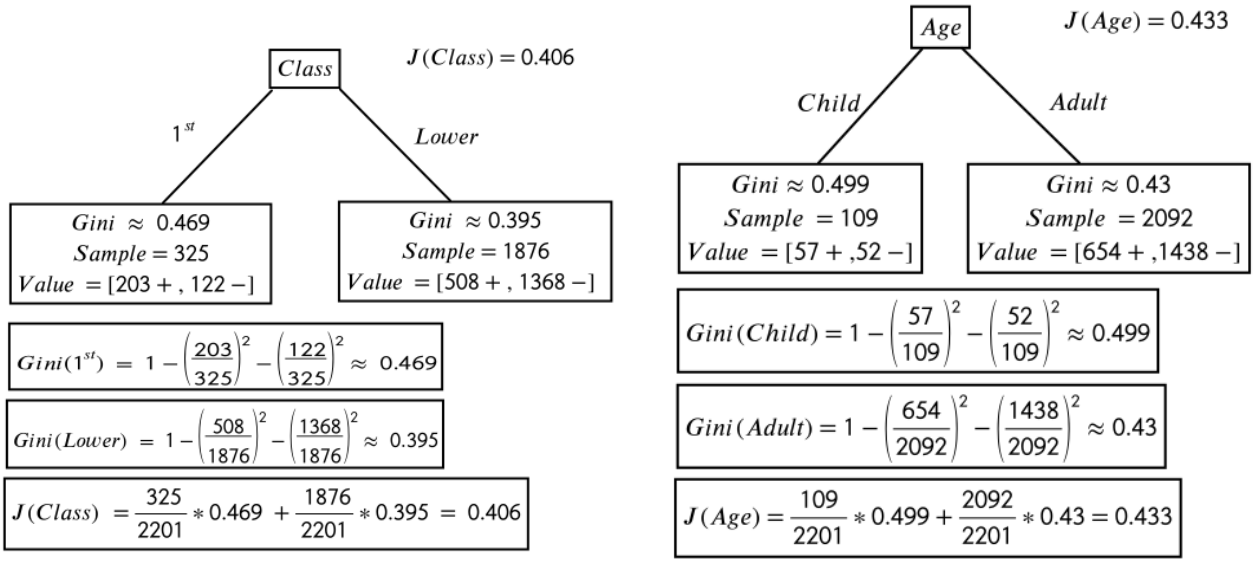
c)

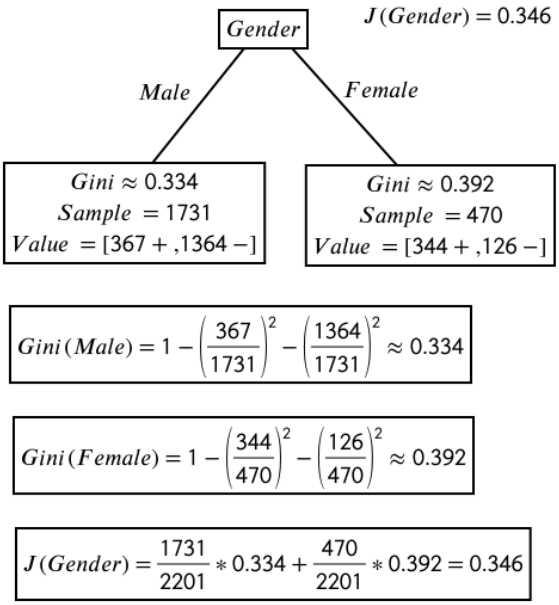


d) Total error = 1/3,
weight of stump = 1/2*log(1-total_error/total_error)
weight of stump = 1/2*log((1-1/3)/1/3)
weight of stump = 1/2*log(2)
weight of stump = 1/2*log(1-total_error/total_error)
weight of stump = 0.1505
For misclassified samples (x=5)
new sample weight = sample weight x $e^{\text{weight of stump}} = 1/3 * e^{0.1505}$
new sample weight = 1/3 * 1.162
new sample weight = 0.387

For correctly classfied samples (x=1 and x=3)
new sample weight = sample weight x $e^{-\text{weight of stump}} = 1/3 * e^{-0.1505}$
new sample weight = 1/3 * 0.8602
new sample weight = 0.287

Q4)
a)





According to the calculation of costs, Gender(G) should be picked as the root of decision tree with the lowest cost among the features..

b)

$$Accuracy = \frac{\#of\ correctly\ classified}{total\ samples}$$

$$Accuracy = \frac{1364 + 344}{2201} = 0.776$$

$$Accuracy = 77.6\ \%$$

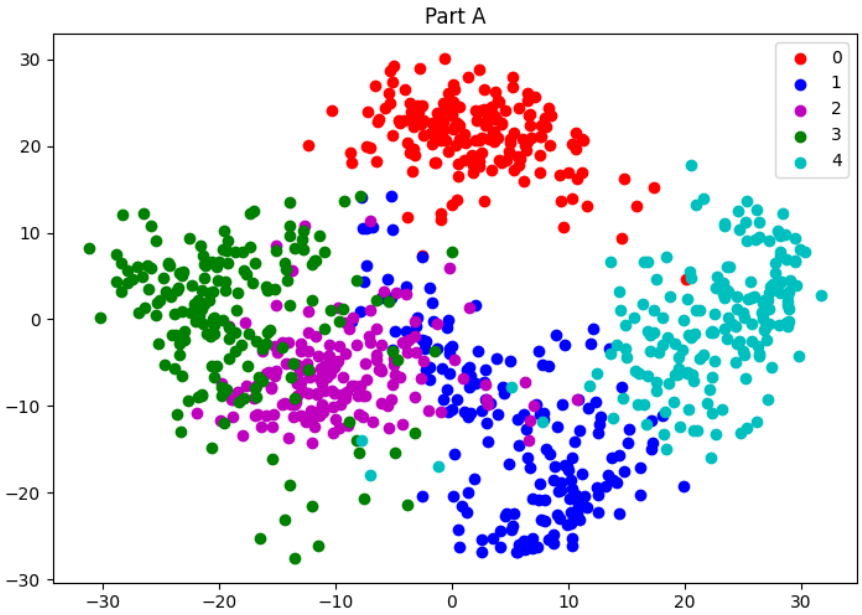
Q5)

a) It is a regression problem because output takes continuous values.
b) I used scikit's Support vector regression(SVR) class. Here are the learnt parameters:

```
@_deprecate_positional_args
def __init__(self, *, kernel='rbf', degree=3, gamma='scale',
             coef0=0.0, tol=1e-3, C=1.0, epsilon=0.1, shrinking=True,
             cache_size=200, verbose=False, max_iter=-1):
```

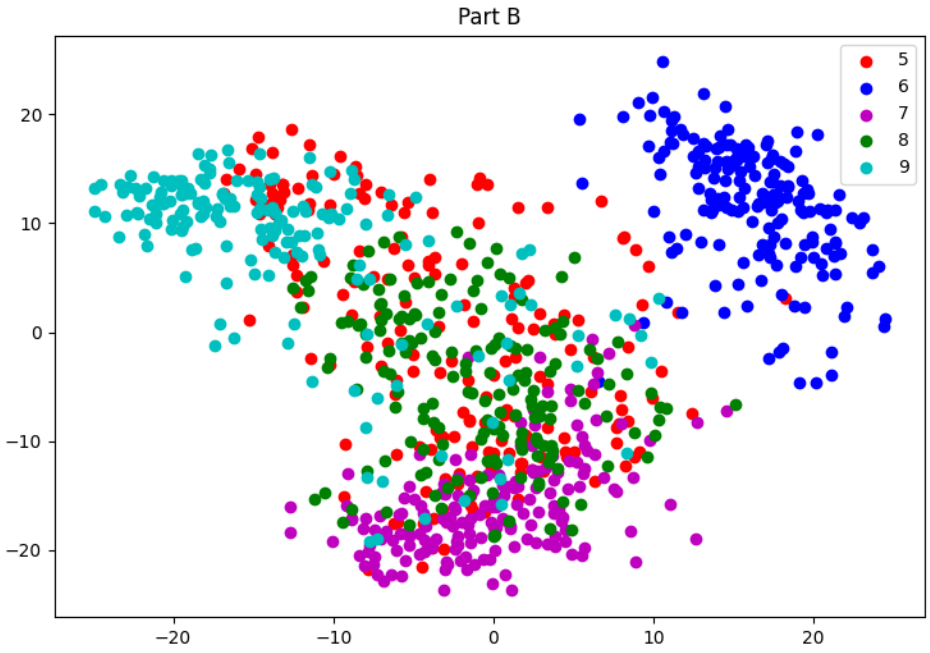
c) Here are the preedictions = [3.43261808 2.05860259 4.97811039]
It is in order so,
M2 = 3.43261808
M7 = 2.05860259
M11 = 4.97811039

Q6)
a)



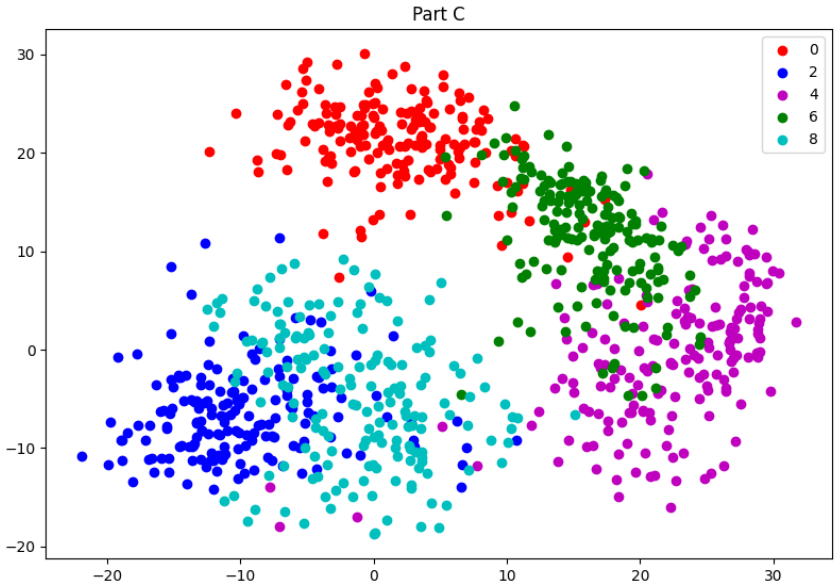
We can observe that 2(Magenta) and 3(Green) can be confused they are close to each other. 4(Cyan), 0(Red), 1(Blue) have clear seperation for others.

b)



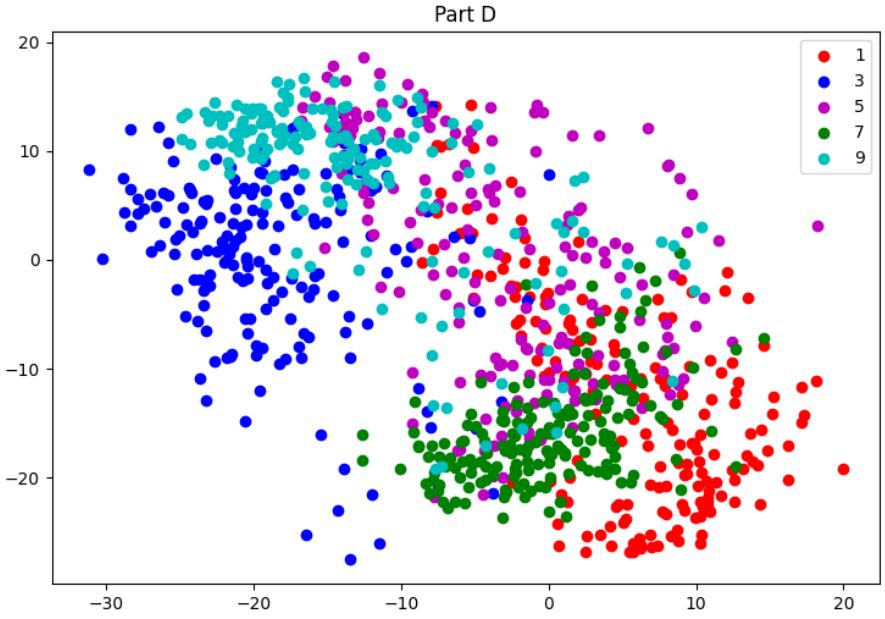
6(Blue) and 9(Cyan) are distinct from other three integers. Especially 5(Red) and 8(Green) can be confused. 7(Magenta) is also distinguishable.

c)



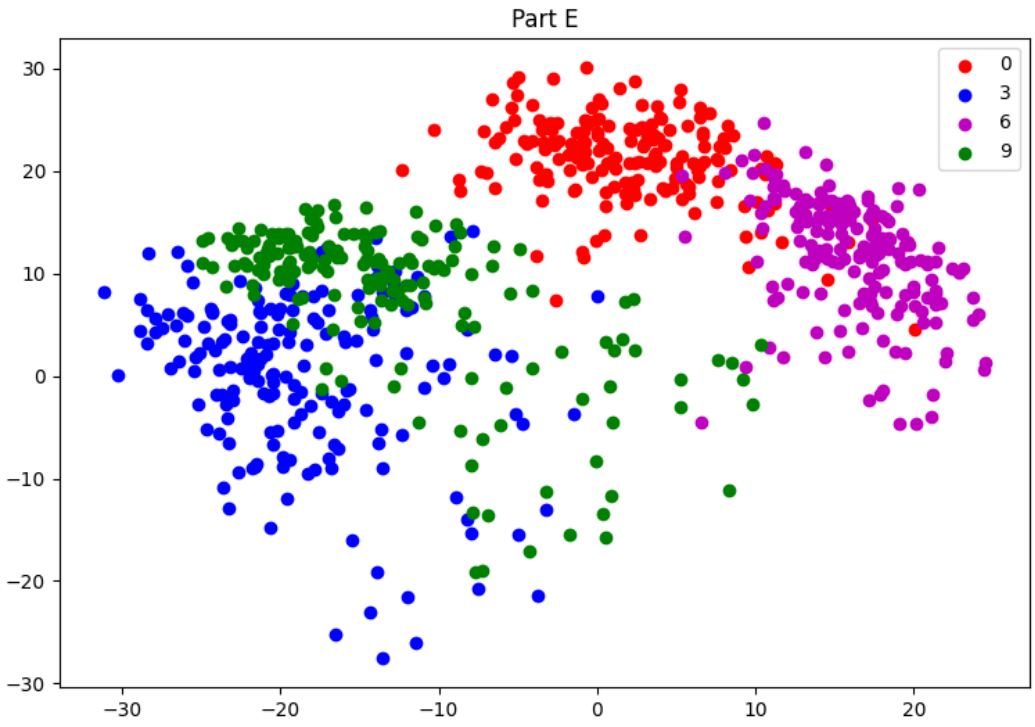
0(Red) has clear seperation from other clusters. 2(Blue) and 8(Cyan) can be somehow confused. Altough there is some mixing between 4(Magenta) and 6(Green) we can say that they are seperable.

d)



Here 5(Magenta) is all over the place. 7(Green) and 1(Red) can be confused. 3(Cyan) and 9(Blue) can are also close to each other and can be confused.

e)



Again 3(Blue) and 9(Green) are close to each other and can be confused. Altough there is some closeness and outliers, we can say that 0(Red) and 6(Magenta) are seperable.

Q7)
Hyperparameters:
Learning Rate = 0.3,
Units in the hidden layer = 4,
Number of iteartions = 1000

Accuracy = 100 %. Note that on some runs I got 92 and 96 percent accuracy rates too.