Question4

If it is meaningless to weight the variables of the category class, it will disturb the closest distance, such as the distance between private and never-work, which is farther than private and state-gov, but in fact they are theoretically equal in distance.

Question5

At a percentage of 24.6% that the training data has an income > 50k.

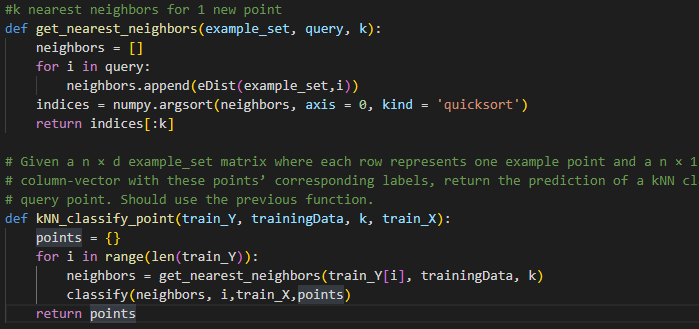
A model reached an accuracy 70% is a good model.



Using np.shape(trainingData), we know there are 85 dimensions.

Question6

Question7



Question8

def cross\_validation(train\_X, foldA, foldB, foldC, foldD, train\_Y, k):

    subset1 = combine: foldB, foldC, foldD

    subset2 = combine: foldA, foldC, foldD

    subset3 = combine: foldA, foldB, foldD

    subset4 = combine: foldA, foldB, foldC

    pointSet1 = fold A

    pointSet2 = fold B

    pointSet3 = fold C

pointSet4 = fold D

    accuracyset1 = get the accuracy of subset1

    accuracyset2 = get the accuracy of subset2

    accuracyset3 = get the accuracy of subset3

    accuracyset4 = get the accuracy of subset4

    var = numpy.var([accuracyset1, accuracyset2, accuracyset3, accuracyset4])

    Average = ((accuracyset1 + accuracyset2 + accuracyset3 + accuracyset4)/4)

Question9

K = 1 0.987

K = 3 0.897

K = 5 0.879

K = 7 0.863

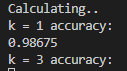
K = 9 0.833

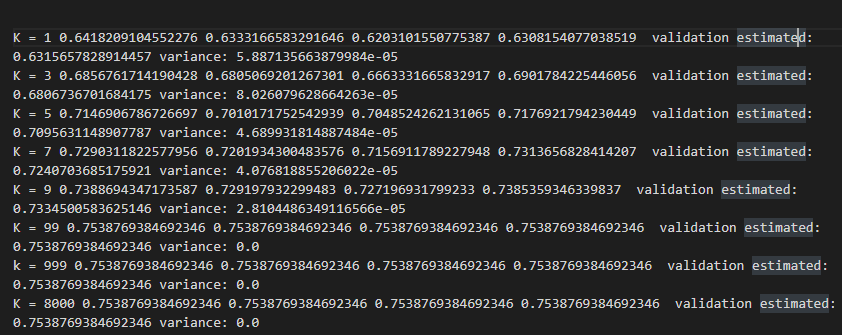
K = 99 0.833

K = 999 0.824

K = 8000 0.755

The best number of neighbors (k) I observed is when k is smallest. When k = 1, the training accuracy is 0.987, where the error isn’t exactly zero.





Cross-validation accuracy rate will increase with increasing k. Both a small and a large number of k might result in overfitting or underfitting, respectively. Overfitting implies that a model performs well on training data but poorly when fresh data is introduced.