NNSE-784-3510 - Applications of Machine Learning in Data Science

Project 1:

In this project, you will provide a demo of KNN algorithm by creating a basic implementation of this algorithm from scratch. Both MATLAB and Python implementations are accepted. **You should not use any MATLAB or Python functions for KNN, and must implement the KNN algorithm and visualization from scratch.**

Assume you have collected data about "two hundred" COVID-19 test subjects and your test has two features shown by X1 and X2 variables. The test results in two outcomes, positive or negative.

a) Generate your training class by assuming both X1 and X2 follow random Gaussian distribution and have the following relationship:

   X1 ~ G(0.5,2) where G stands for Gaussian distribution, 0.5 is the mean of the distribution and 2 is the standard deviation.

   X2 ~ G(1,3)+ 1.5*X1, where 1.5 is the linear correlation between X1 and X2.

   For any test subjects which have (X1 > 1) and (X2 > 3) the test outcome is positive. All other test subjects are negative.

b) Visualize your test subjects on a scatter plot of X1 and X2. Show the positive subjects with red and the negative subjects with blue. You must have 200 data points shown in blue and red scattered in your plot. This is a visualization of your training class. See Figure 1 for an example of the scatter plot, where X1 and X2 are horizontal and vertical axes respectively.

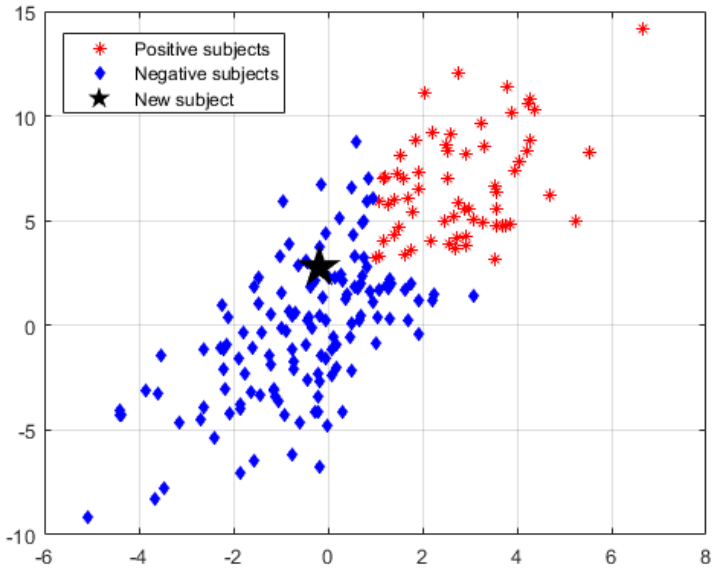

Figure 1 - COVID-19 Test Data Scatter Plot

c) Now that you have a training class, create a function called “KNN_Classifier” which takes a new subject’s X1 and X2 features as a vector (P), your training class generated in step a), and K as the number of neighbors to be counted when determining a new subject’s type in KNN algorithm. Your function must return the type of the new subject as well as its distance from every point in your training class. The function’s prototype in pseudo-code is provided below:

[Type, Distance] = KNN_Classifier(P, Training_Class, K)

d) Generate a new subject’s test results randomly by X1~N(0.5,2) and X2~N(1,2).

e) Using the visualization in step b), show the new subject’s datum on the scatter plot of your training class. This new subject is shown by a black star in Figure 1. Note that your scatter plot and new test subject will be slightly different from the example in Figure 1 as the data is generated randomly.

f) Determine the result of the new subject’s test by using the function in step c).

g) Using the training class generated in step a), generate the decision boundaries and positive and negative regions for K=1, K=4, K=8, and K=20. See examples below.

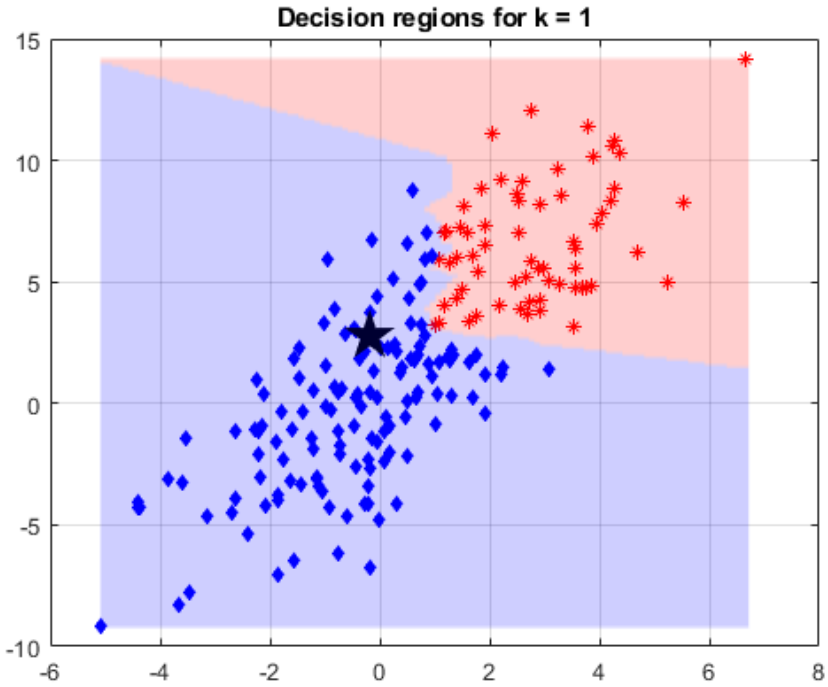

Figure 2 - Decision Regions for K=1

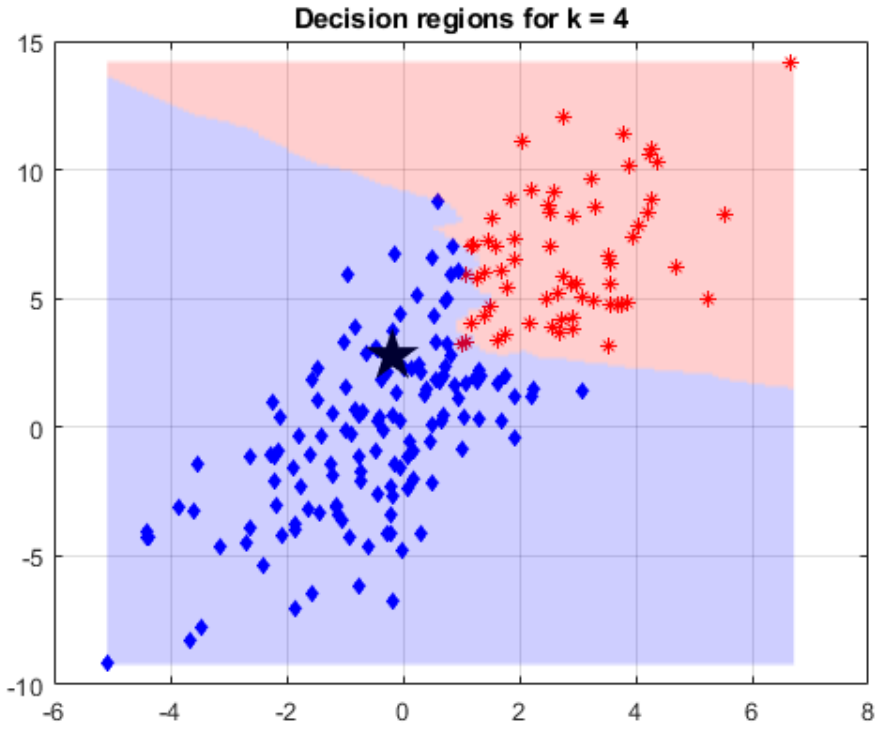

Figure 3 - Decision Regions for K=4

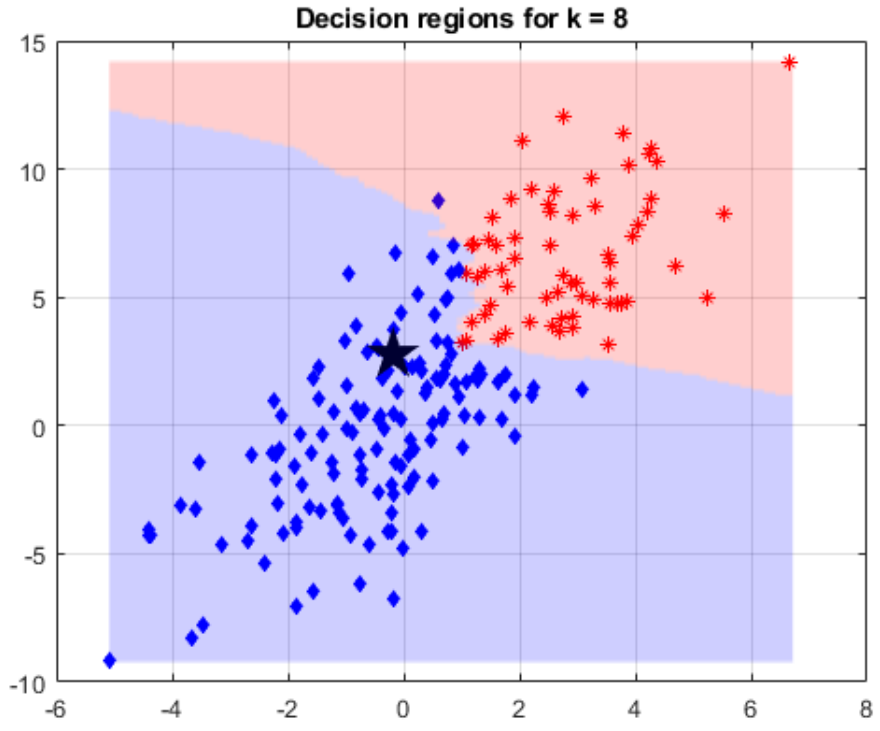

Figure 4 - Decision Regions for K=8