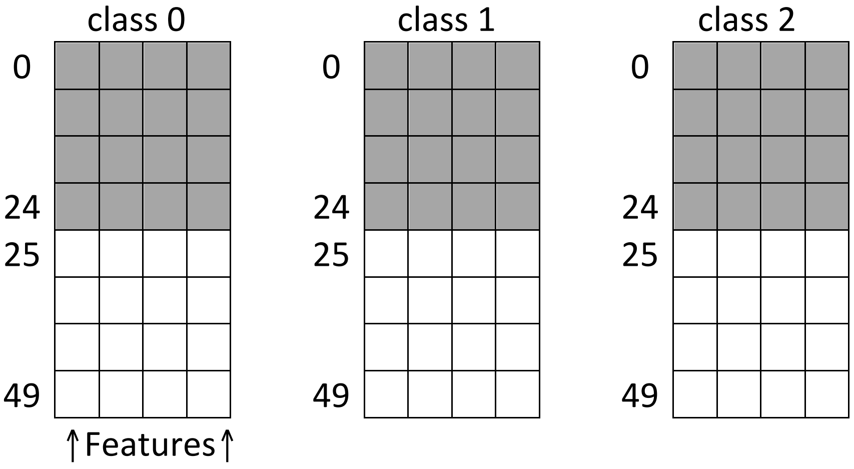
k-PPV

1. Make sure you have the following files onto your virtual machine: iris.data and kPPV.java. You will be able to compile and test your program using Eclipse IDE (specify a local folder for workspaces, such as Desktop):
    a. create a new java project named kPPV
    b. add the source file kPP.java in the src folder of the project within Eclipse (specify copy the file)
    c. add the iris.data file to kPPV project (root; specify copy the file)
    d. when the code is ready (see below) you can run it directly or execute it step by step.

2. In the kPPV.java file, complete the *computeDistances* method taking as input one example (vector X) and returning the distances table (also given as input but empty) filled with all Euclidian distances computed between X and all learning examples inside *data* structure. *data* [][][] structure is accessible everywhere in your file and has the following structure :

data [NbClasses][NbEx][NbFeatures] : ▨ learning set, □ test set



3. Write a function that returns the class of an example (X) using the distances table previously computed. For the 1-PPV, this class is the one corresponding to the class of the nearest neighbor (corresponding to minimum value in distances table).

4. In *Main* function, perform the evaluation procedure that will go through all test examples in data, find its class according to 1-PPV and compute the recognition rate and confusion matrix.

5. (Optional) Add a parameter *KNeighbors* in order to turn your 1-PPV into a real K-PPV.

6. (Optional) Modify your program to do cross-validation.