Potential items to look for during formal inspection

Cost Matrix.java

Coding Standards • understandable

• adhere code guidelines

• indentation : Good indentation

• no magic numbers: there are no magic numbers

// Line 180 to 182 and line 320 to 322

All of the strings can be assigned to a constant string values before using it

All index values can be declared as constant

• naming: Method name: applyCostMatrix()

// Line number 187

Instance inst = data.instance(i);

Object name as inst is not a proper name

// Line 310 and 339

A variable named as ‘classProbs’ can be well written as ‘classProb’ which estimates class probability

• units, bounds

• spacing: horizontal (btwn operators, keywords) and vertical (btwn methods, blocks) Comments

Comments:

Un-used code is not removed

• no needless comments

• no obsolete comments

• no redundant comments

• methods document parameters it modifies, functional dependencies

• comments consistent in format, length, level of detail

• no code commented out Logic

• array indexes within bounds

• conditions correct in ifs, loops • loops always terminate • division by zero • refactor statements in the loop to outside the loop Error Handling • error messages understandable and complete • edge cases (null, 0, negative) • parameters valid • files, other input data valid Code Decisions • code at right level of abstraction • methods have appropriate number, types of parameters • no unnecessary features • redundancy minimized • mutability minimized • static preferred over nonstatic • appropriate accessibility (public, private, etc.) • enums, not int constants • defensive copies when needed • no unnecessary new objects • variables in lowest scope • objects referred to by their interfaces, most generic supertype