Using WEKA classifiers in your java program

Session 1

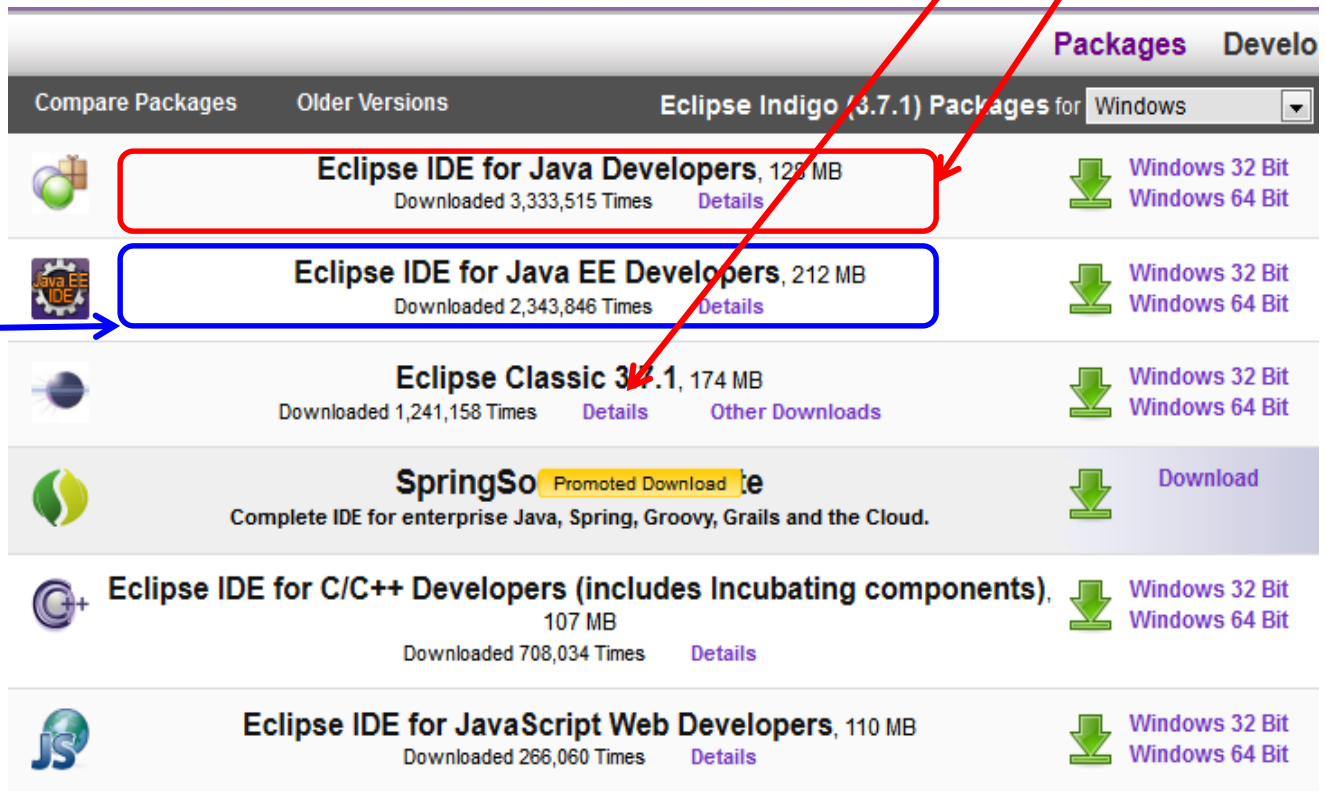
With Eclipse

EASY START

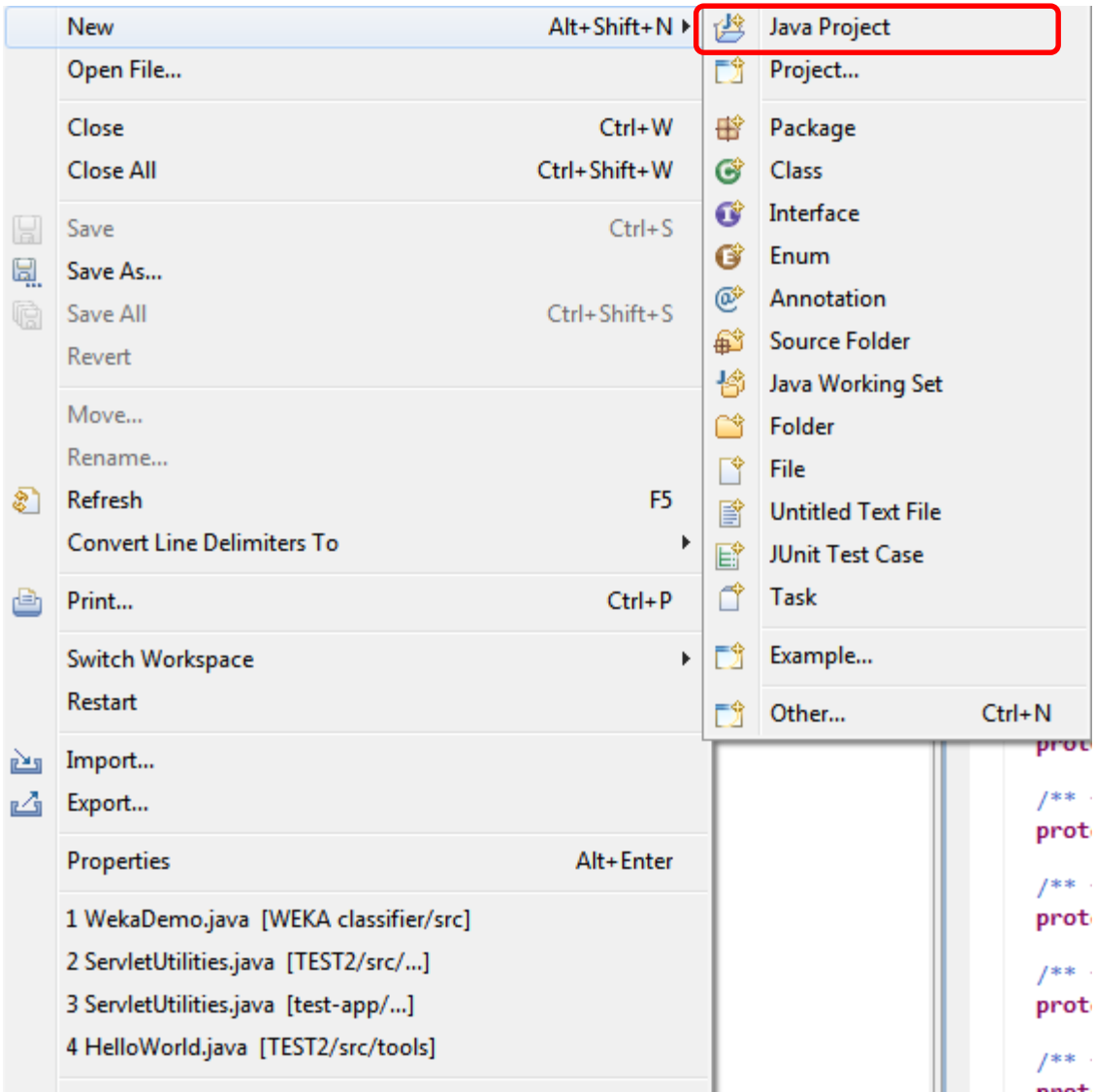
Install java development platform (eclipse)

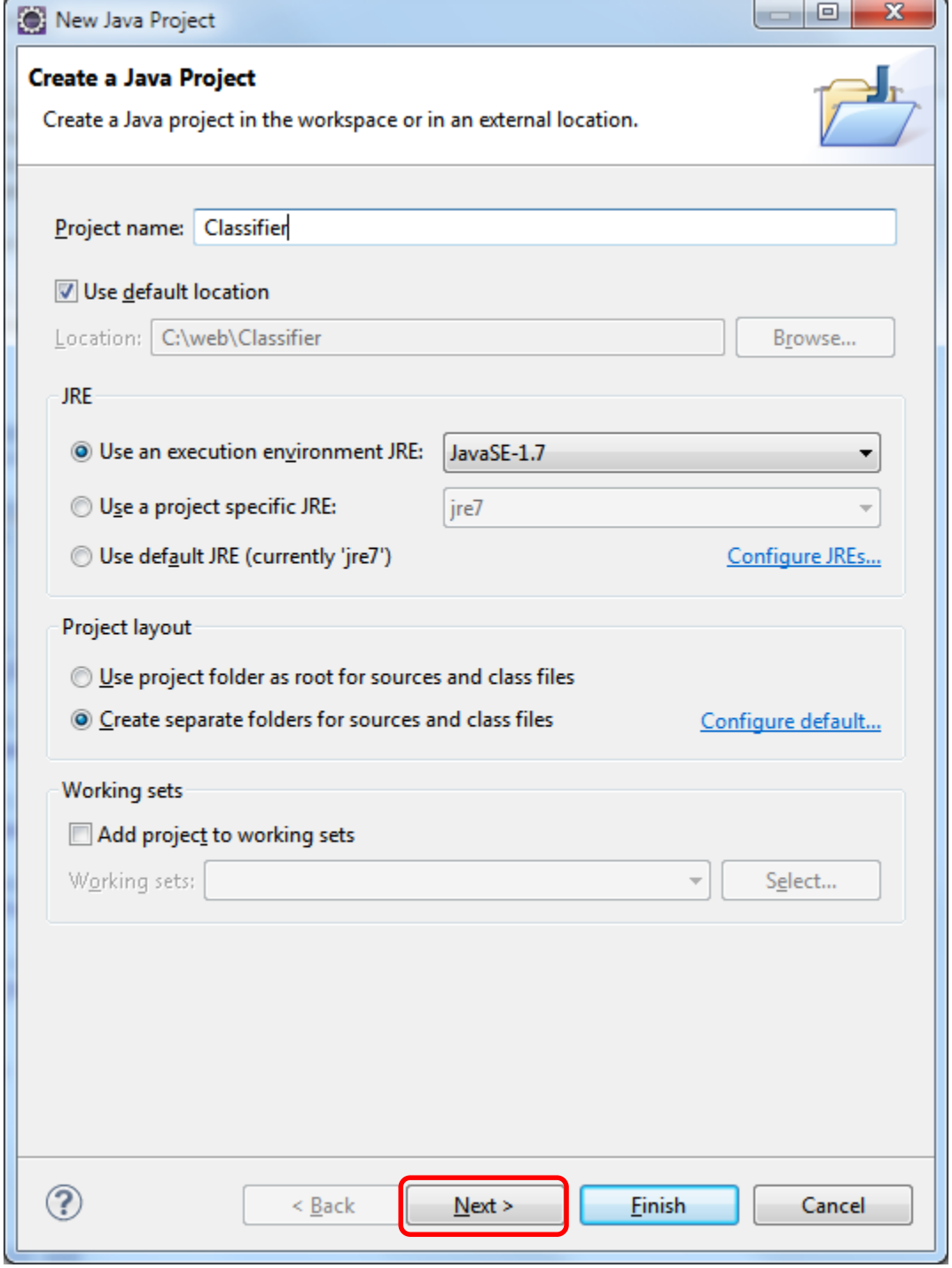
If you develop a desktop application

If you develop a WEB application (JSP)



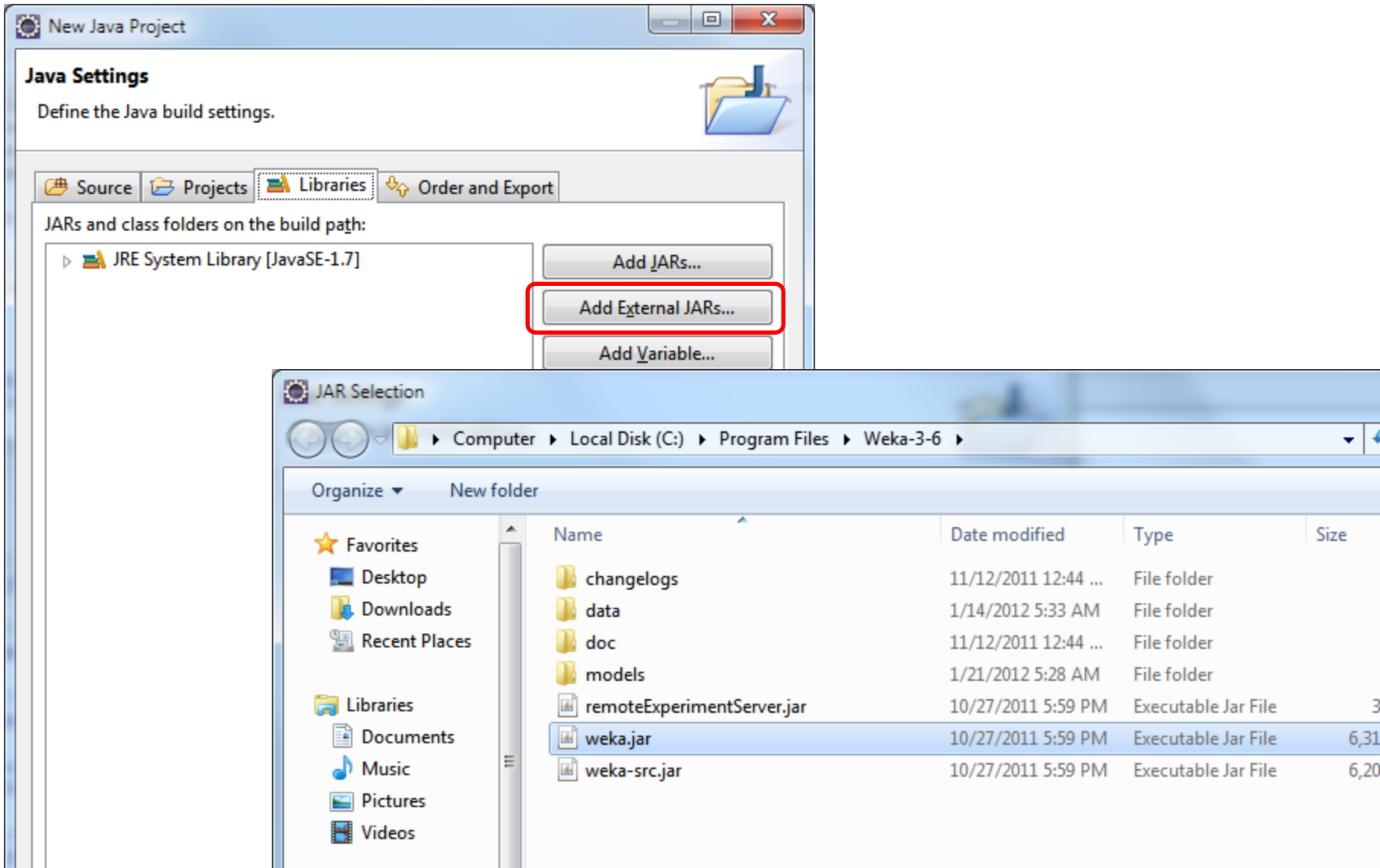
New java project





Project name

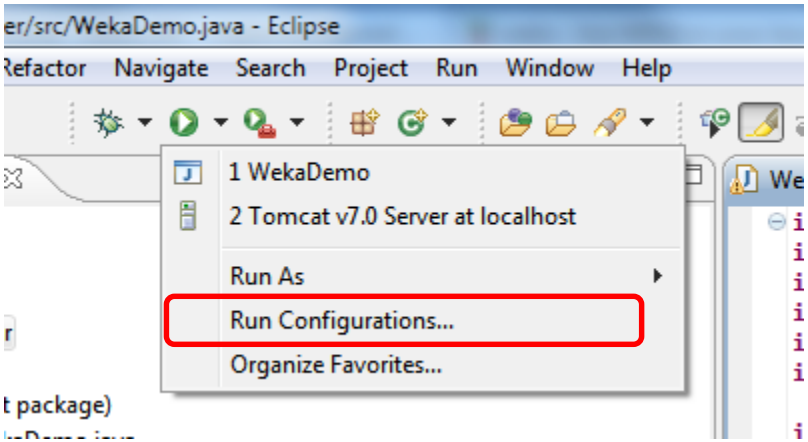
Add WEKA library

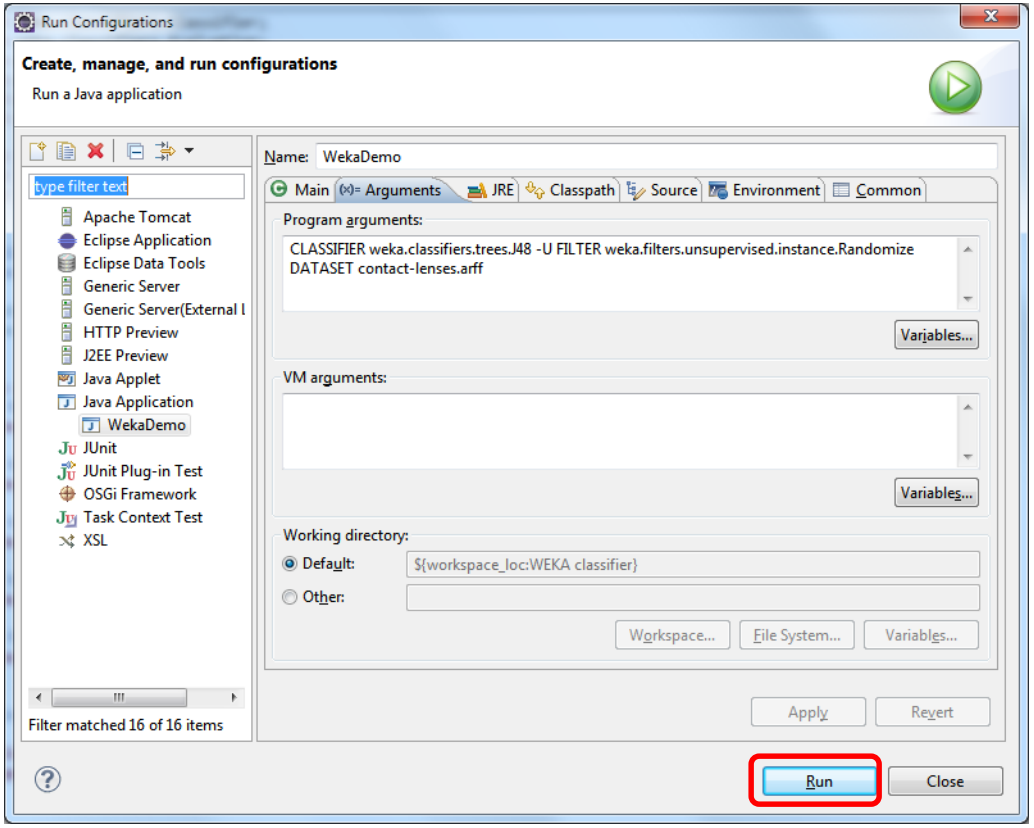


Add Weka Demo

- Add new class “WekaDemo”
- Copy and paste attached code in WekaDemo.java
- Save
- Copy your input file into project folder

Set program parameters





Unpruned tree

CLASSIFIER weka.classifiers.trees.J48 -U
FILTER
weka.filters.unsupervised.instance.Randomize
DATASET contact-lenses.arff

Program output

```
Weka - Demo
===========

Classifier...: weka.classifiers.trees.J48 -U -M 2
Filter.......: weka.filters.unsupervised.instance.Randomize -S 42
Training file: contact-lenses.arff

J48 unpruned tree
------------------

tear-prod-rate = reduced: none (12.0)
tear-prod-rate = normal
|   astigmatism = no: soft (6.0/1.0)
|   astigmatism = yes
|   |   spectacle-prescrip = myope: hard (3.0)
|   |   spectacle-prescrip = hypermetrope: none (3.0/1.0)

Number of Leaves  : 4

Size of the tree : 7
```

```
Correctly Classified Instances          17               70.8333 %
Incorrectly Classified Instances         7               29.1667 %
Kappa statistic                          0.4381
Mean absolute error                      0.2167
Root mean squared error                  0.446
Relative absolute error                 57.3529 %
Root relative squared error            102.1123 %
Total Number of Instances               24

=== Confusion Matrix ===
 a  b  c   <-- classified as
 4  0  1 |  a = soft
 0  1  3 |  b = hard
 1  2 12 |  c = none

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.8       0.053      0.8       0.8       0.8       0.853    soft
                 0.25      0.1        0.333     0.25      0.286     0.575    hard
                 0.8       0.444      0.75      0.8       0.774     0.633    none
Weighted Avg.    0.708     0.305      0.691     0.708     0.698     0.669
```

Writing your own classification program

1. Convert you training set into arff format
2. Convert the data you want to classify into arff format
3. Sample code is in file WekaClassify.java
4. The program takes two arguments: the training file name and the name of the file with new instances to be classified

Sample program arguments: contact-lenses.arff newinstances.arff

COMMAND LINE USAGE

Compile jour java code

javac -classpath /path/to/weka.jar WekaDemo.java

javac -classpath /path/to/weka.jar WekaClassify.java

Run your code

java -classpath .; /path/to/weka.jar WekaClassify contact-lenses.arff newinstances.arff

To increase java heap size:

java -Xmx1G -cp /path/to/weka.jar WekaClassify contact-lenses.arff newinstances.arff