Data Mining III – Lesson 4

Tamara B. Sipes, Ph.D.

Last Time: Lesson 3

- Mining of the image segments dataset, beginning to end:
  - Problem definition
  - Data Preparation
  - **Data Mining**
  - **Evaluation**
  - Presentation

- The focus was on the method and evaluation

Lesson 3: In Summary

- Another start-to-end easy data mining project
- The importance of having data mining expertise and understanding of the methods and the evaluation
- Building our experience and intuition

Lesson 3: The Lesson Learned

- Data Mining is a highly iterative process
- Lots of paths and lots of feedback
- Many traversing options
- Your intuition will guide you, once you have gained a little bit more of the data mining experience
- Your knowledge about the methods, how they work, what they can accomplish, what kind of data they are best suited for, etc.

Lesson 4 Overview

- Continue the mining of the Image Segmentation dataset
- Transition to more complex data mining tasks
- Introduction to several of the common issues when mining real-world datasets
- Focus on model building for best novel data/unseen data evaluation results

Instructions

- Hands-on lesson #3
- Again, please have both the Lesson 4 window opened and the weka Explorer window
- Follow the step by step instructions, and perform the modeling of the dataset yourself as well, as the end result of this Lesson will serve as the start of your Assignment II

- Let's get started!

Dataset

- Open IMAGE_AssignmentII.arff that you saved at the end of Lesson 3
- TO DO:
  - Open weka 3.5.7 or similar
  - Choose Explorer from the Applications menu
  - “Open file” IMAGE_AssignmentII.arff

Problem Description

- Image Segmentation data
- Create a predictive model of IMAGE (classification)
- Each instance in the data is a 3x3 region
- Number of Instances: 1500
- Number of Attributes: 19 attributes
- 18 numeric attributes and 1 nominal (class) attribute
- 7 values of the class attribute IMAGE

Mining Summary

- We went through 10 data mining iterations
- We decided to keep a J48graft model

- Let's look at the rest of the iterations and the corresponding models

Representations Tree

- RepTree with minNum = 4 (the minimum number of instances in a leaf):
- weka.classifiers.trees.REPTree -M 4 -V 0.0010 -N 3 -S 1 -L -1
- Correctly Classified Instances 1417 94.4667 %
- Incorrectly Classified Instances 83 5.5333 %

Decision Tree

- weka.classifiers.trees.J48 -C 0.25 -M 2

Correctly Classified Instances	1436	95.7333 %
Incorrectly Classified Instances	64	4.2667 %

Grafting Decision Tree

- weka.classifiers.trees.J48graft -C 0.25 -M 2
- Resulting Tree:
  - Number of Leaves : 165
  - Size of the tree : 329

- Correctly Classified Instances: 96.2 %

Best-First Decision Tree

- weka.classifiers.trees.BFTree -S 1 -M 2 -N 5 -C 1.0 -P POSTPRUNED

  (Method for generating best-first decision trees)
- Resulting Tree:
  - Number of Leaves : 47
  - Size of the tree : 24

- Correctly Classified Instances 95.1333 %

Logistic Model Tree

- weka.classifiers.trees.LMT -I -1 -M 15 -W 0.0

  (Classifier for building 'logistic model trees', which are classification trees with logistic regression functions at the leaves)

- Number of Leaves : 9
- Size of the Tree : 17
- Correctly Classified Instances 96.8 %

ANN

- weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

  (method for the Multilayered Perceptron ANN Learning)

- Results: 96.7333 %
- Model: black box

Random Forest

- weka.classifiers.trees.RandomForest -I 10 -K 0 -S 1 (method for the Random Forest Learning)

- Results: 97.6 %
- Model: black box (“Random forest of 10 trees, each constructed while considering 5 random features”)

The Shelf-Life of a Model

- Ideally, a model should be fine-tuned every 6 months - 1 year
- In real life, the influx of new data is constant
- Depends on the problem
- To do:
  - Collect new data
  - Test your model on unseen/novel data
  - Rerun the model on the (old + new) Data

IMAGE Unseen Data

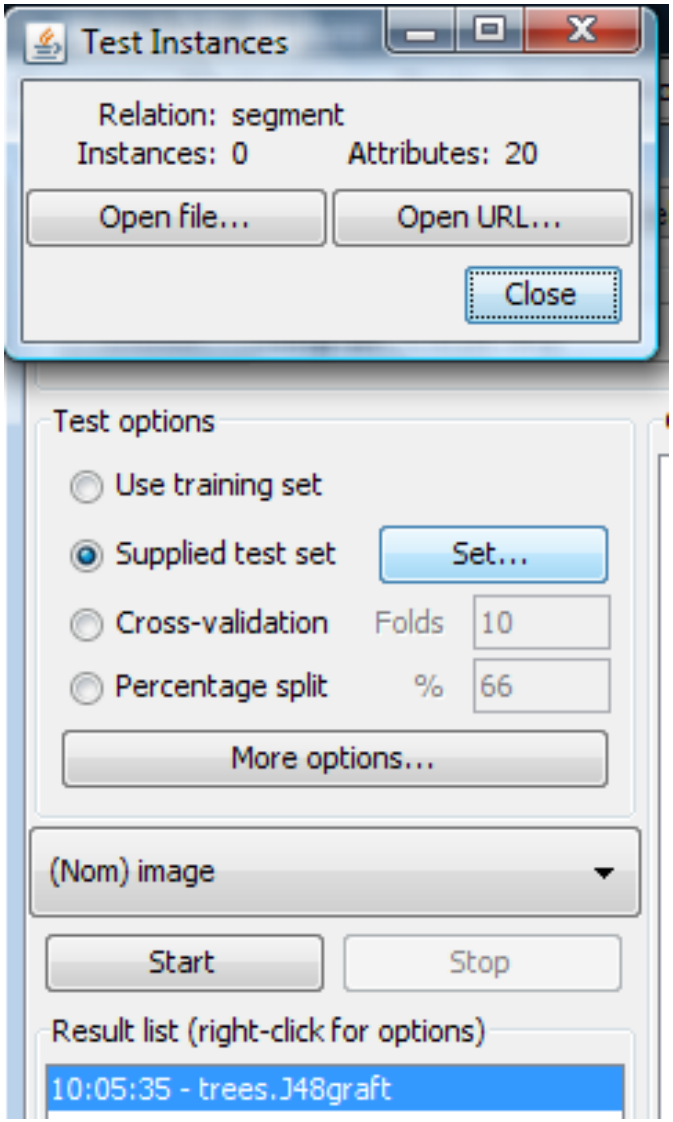
- The researchers form the U of Massachusetts Vision Research Center inform us that the novel IMAGE data that we have not used is now available!
- Class Resources:

  ImageSegmentationData_TestSet.arff
- Number of instances: 810
- Let's test our existing models, and check to see how the estimate we used, the 10-fold cross validation scores, performed

Running a Model on Unseen Data

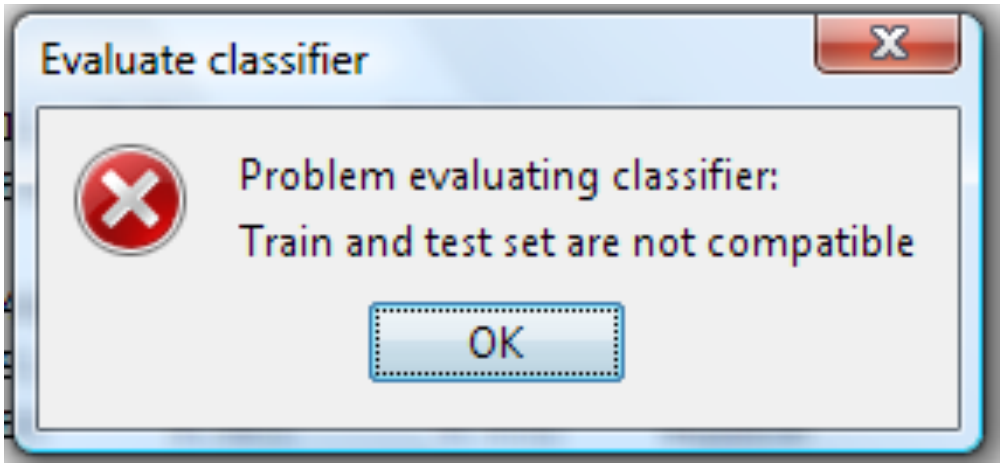
- In weka, to run a model on unseen data:
  - Go to Classify Tab (for classification problems)
  - Select “Supplied test set”
  - Click on “Set”
  - Choose the ImageSegmentationData_TestSet.arff file from your local directory
  - Click “OK”

STS Evaluation



Oops!

- 19 vs. 20 attributes!
- Need to be compatible!



Making The Sets Compatible

- Open ImageSegmentation_TestSet.arff from within weka
- Delete the region-pixel-count variable
- Save as: ImageSegmentationData-TestSet_19attr.arff

Next

- Now, we are ready to use the unseen data set
- Let's evaluate the existing models using the STS evaluation:
  - Representations Tree
  - Decision Tree
  - DT graft
  - BF DT
  - LM tree
  - ANN
  - Random Forest

Representations Tree

- weka.classifiers.trees.REPTree -M 4 -V 0.0010 -N 3 -S 1 -L -1
- 10-fold evaluation:
- Correctly Classified Instances 1417 94.4667 %
- Incorrectly Classified Instances 83 5.5333 %
- STS evaluation:
  - Correctly Classified Instances 768 94.8148 %
  - Incorrectly Classified Instances 42 5.1852 %

Decision Tree

- weka.classifiers.trees.J48 -C 0.25 -M 2

- 10-fold evaluation:
  - Correctly Classified Instances 1436 95.7333 %
  - Incorrectly Classified Instances 64 4.2667 %

- STS evaluation:
  - Correctly Classified Instances 779 96.1728 %
  - Incorrectly Classified Instances 31 3.8272 %

Grafting Decision Tree

- weka.classifiers.trees.J48graft -C 0.25 -M 2
- 10-fold evaluation:
  - Correctly Classified Instances 1443 96.2 %
  - Incorrectly Classified Instances 57 3.8 %
- STS evaluation:
  - Correctly Classified Instances 777 95.9259 %
  - Incorrectly Classified Instances 33 4.0741 %

Best-First Decision Tree

- weka.classifiers.trees.BFTree -S 1 -M 2 -N 5 -C 1.0 -P POSTPRUNED
- 10-fold evaluation:
  - Correctly Classified Instances 1436 95.1333 %
  - Incorrectly Classified Instances 64 4.8667 %
- STS evaluation:
  - Correctly Classified Instances 769 94.9383 %
  - Incorrectly Classified Instances 41 5.0617 %

Logistic Model Tree

- weka.classifiers.trees.LMT -I -1 -M 15 -W 0.0
- 10-fold evaluation:
  - Correctly Classified Instances 96.8 %
  - Incorrectly Classified Instances 3.2 %

- STS evaluation:
  - Correctly Classified Instances 776 95.8025 %
  - Incorrectly Classified Instances 34 4.1975 %

ANN

- weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
- 10-fold evaluation:
  - Correctly Classified Instances 96.7333 %
  - Incorrectly Classified Instances 3.2667 %

- STS evaluation:
  - Correctly Classified Instances 768 94.8148 %
  - Incorrectly Classified Instances 42 5.1852 %

Random Forest

- weka.classifiers.trees.RandomForest -I 10 -K 0 -S 1 (method for the Random Forest Learning)

- 10-fold evaluation:
  - Correctly Classified Instances 1436 97.6 %
  - Incorrectly Classified Instances 64 2.4 %

- STS evaluation:
  - Correctly Classified Instances 785 96.9136 %
  - Incorrectly Classified Instances 25 3.0864 %

Analysis

- The best: Random Forest – 96.9%
- The worst: ANN and RepTree – 94.8%
- Our chosen model: J48graft – 95.9% [~ - 0.2 %]
- The worst score drop: ~ - 1.9 % (ANN)
- The highest increase: ~ + 0.4 % (DT)

Do cross validation scores hold?

- In practice, the 10-fold evaluation scores most often hold really well!
- The most common score difference is in the range:

  [-2 %, +2 %]
- Did you ever wonder why it is the "10" in the 10-fold cross validation? Why not 3-fold? Or, 20-fold?

Which model did the best?

- Random Forest performed the best
- Not very readable model
- Another candidate: J48
- The score pretty good, readability excellent!

Why the Differences?

- 10-fold CV is just an estimate, even though an excellent one
- There is another reason, in this case, though!
- We never checked whether our dataset was balanced

Balanced Data

- Having a balanced dataset means that there is approximately the same number of each types of examples represented to the learning method
- In other words, the distribution of the class variable values should be even
- In the image dataset, do we have a balanced sample?
- Another word for balanced in data mining is: stratified

“Balance Check”

- Open IMAGE_AssignmentII.arff
- Click on the IMAGE variable

Selected attribute

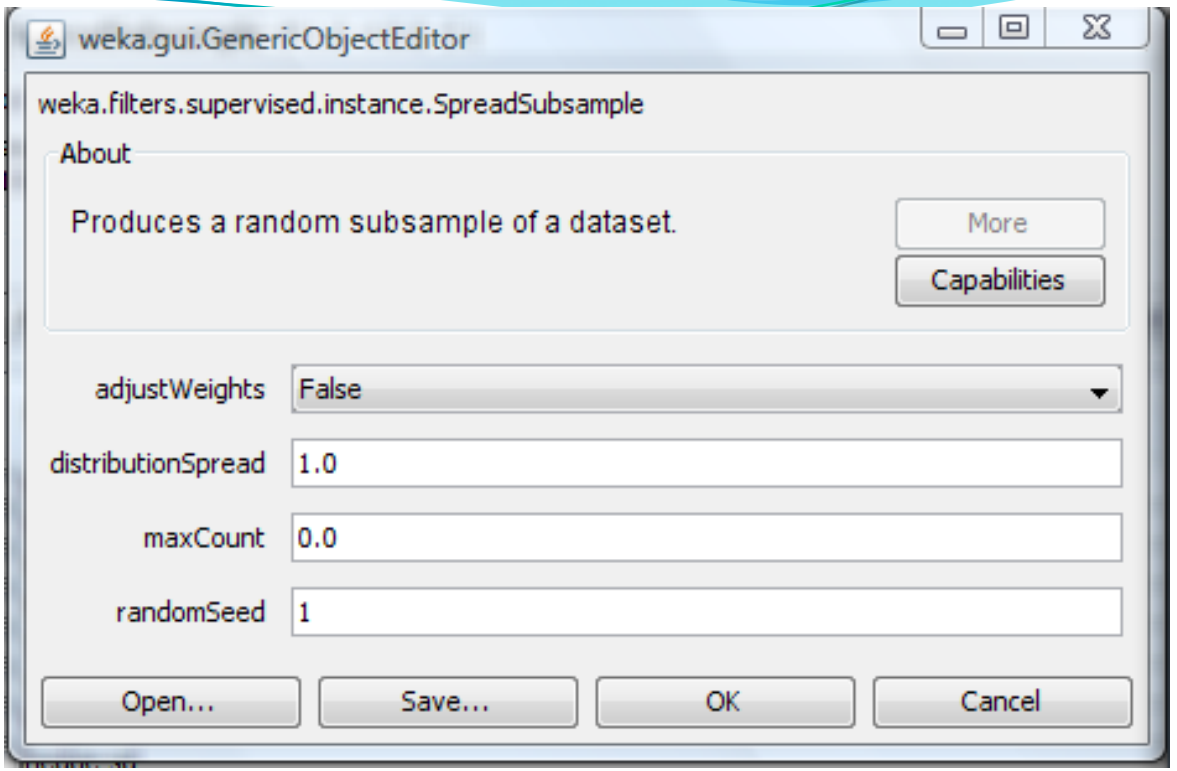
Name: image
Missing: 0 (0%)
Distinct: 7
Type: Nominal
Unique: 0 (0%)

Label	Count
brickface	205
sky	220
foliage	208
cement	220
window	204
path	236
grass	207

How to Balance a Dataset?

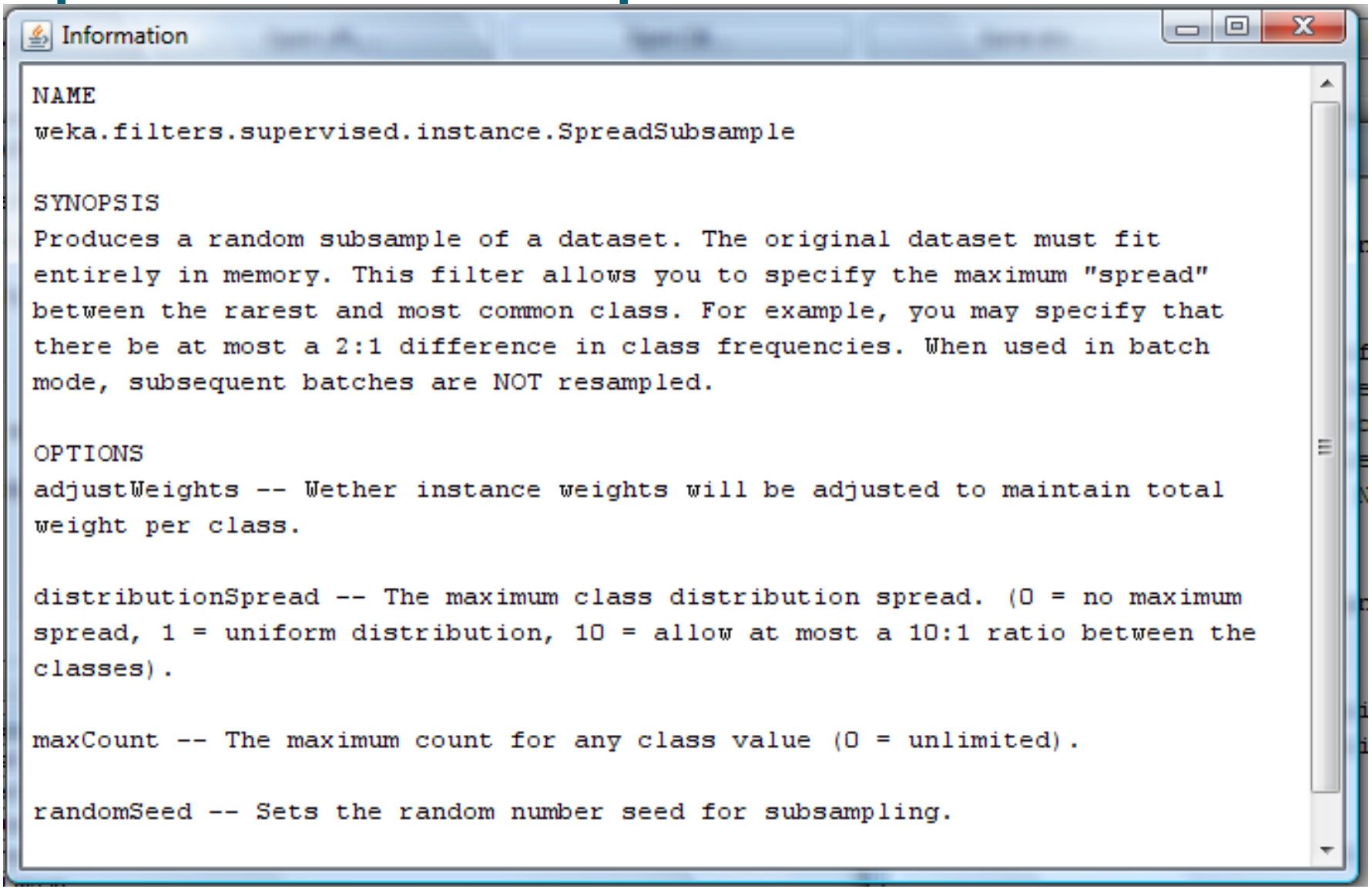
- Counts: min = 204, max = 236
- Two options:
  - Subsample (keep only 204 of each)
  - Expand (sample with replacement, keep 236 of each)

Sampling



- Filter: SpreadSubsample

SpreadSubsample: Parameters

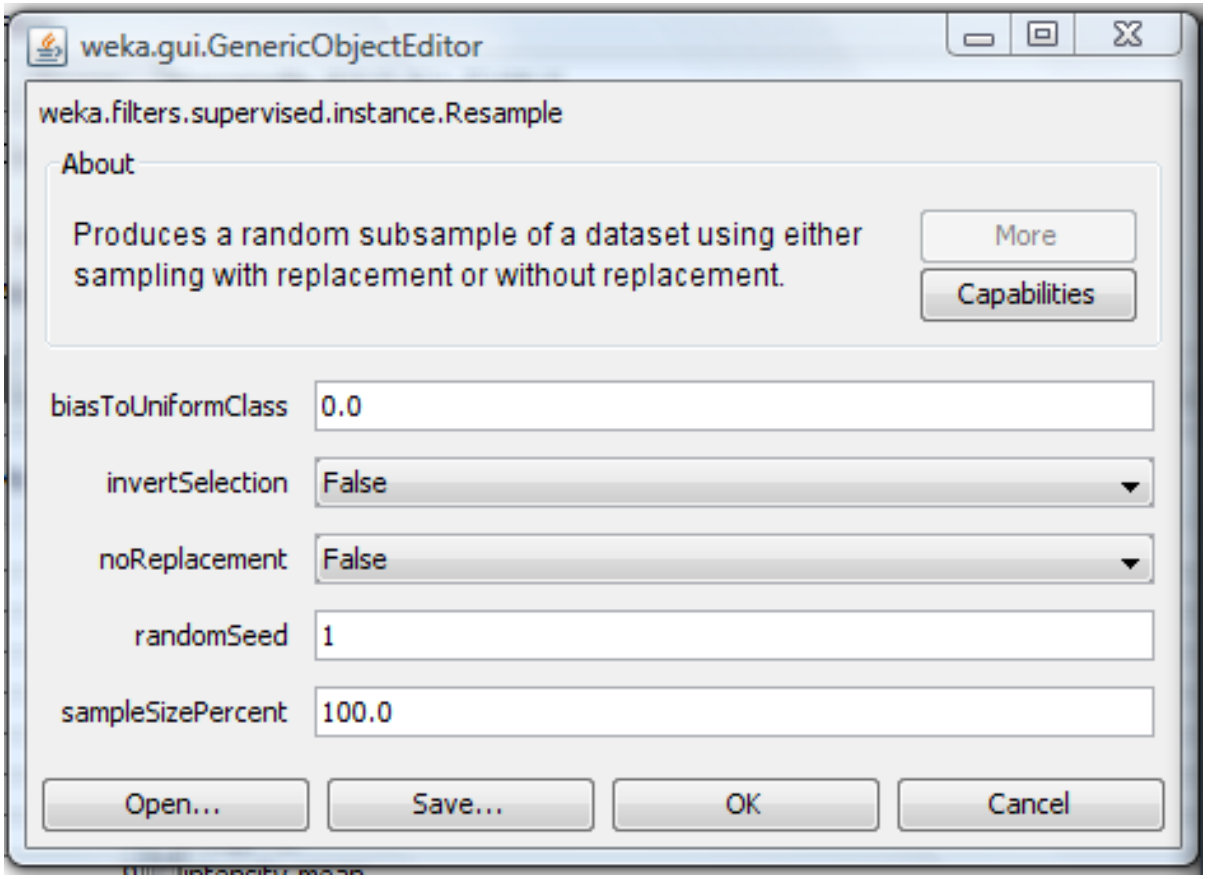


SpreadSubsample: The Call

- weka.filters.supervised.instance.SpreadSubsample -M 1.0 -X 0.0 -S 1

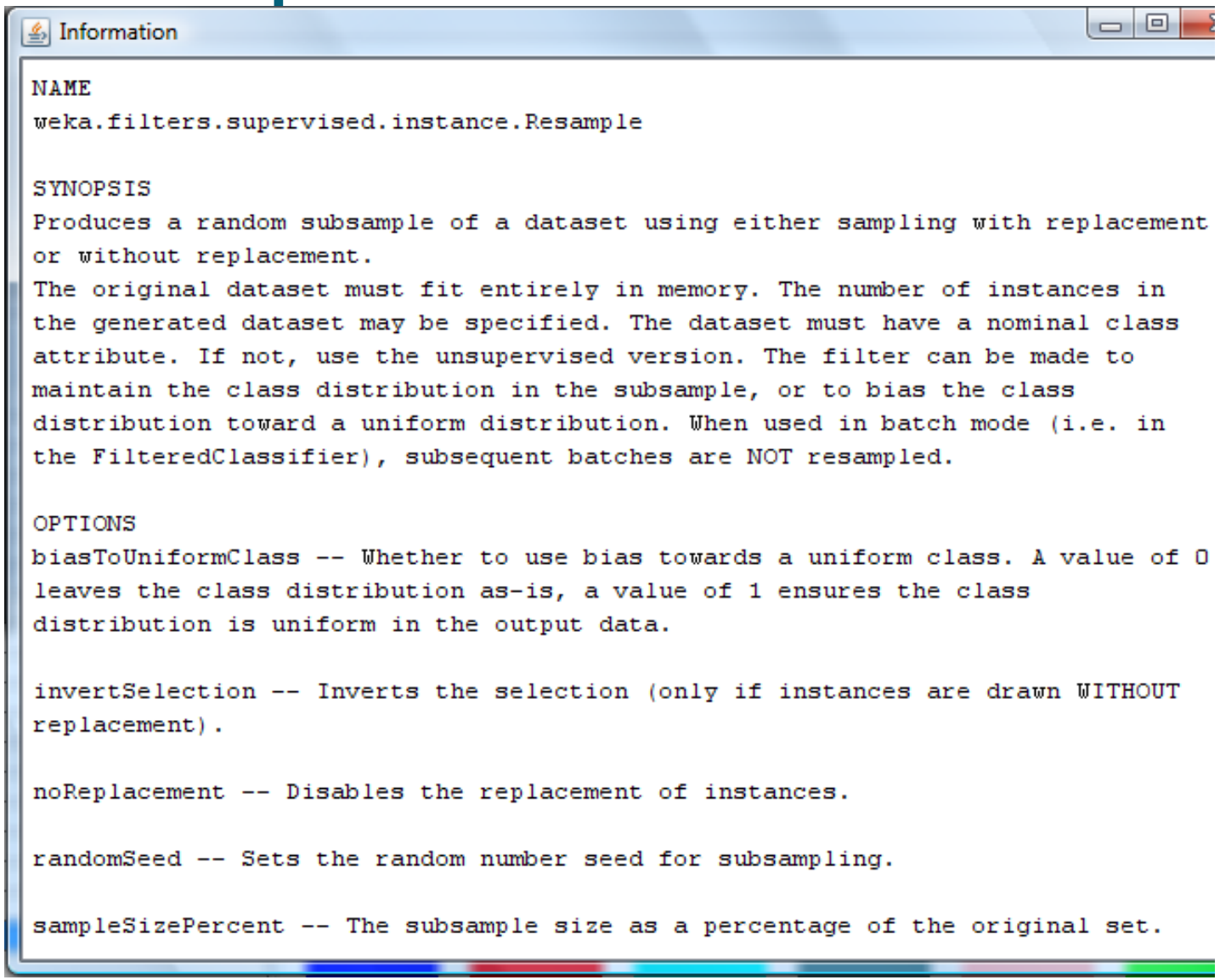
  (distributionSpread = 1.0 for uniform class distribution)
- Click “Apply”
- Number of instances: 1428

Expanding



- Filter: Resample

Resample: Parameters



Resample: The Call

- weka.filters.supervised.instance.Resample -B 1.0 -S 1 -Z 110.00

(distributionSpread = 1.0 for uniform class distribution)

(sampleSizePercent = 110.0, as it is ~ largest class distribution value * 7)

- Click “Apply”
- Number of instances: 1650

What's Next?

- Since we have more data, and know that 10-fold evaluation works really well, let's COMBINE our datasets, and retrain
- We would get 1500 + 810 = 2310 instances!

Merging Two Datasets

- No option to merge two datasets within weka
- Open both IMAGE_AssignmentII.arff and ImageSegmentationData_TestSet_199attr.arff in two WordPad windows
- Highlight and copy the smaller (test set) data without the header
- Paste it at the end of the first file
- Save as IMAGE_AssignmentII_bothSets.arff
- Open IMAGE_AssignmentII_bothSets.arff in weka

Examine the Data

- Examine all 19 attributes!
- Examine the IMAGE variable especially

Selected attribute

Name: image Type: Nominal
Missing: 0 (0%) Distinct: 7 Unique: 0 (0%)

Label	Count
brickface	330
sky	330
foliage	330
cement	330
window	330
path	330
grass	330

Rebuild the Models

- Now, we are ready to rebuild the models using ALL the data
- Let's evaluate the existing models using the 10-fold evaluation:
  - Representations Tree
  - Decision Tree
  - DT graft
  - BF DT
  - LM tree
  - ANN
  - Random Forest

Representations Tree

- weka.classifiers.trees.REPTree -M 4 -V 0.0010 -N 3 -S 1 -L -1
- 10-fold evaluation on training only data:
  - Correctly Classified Instances 1417 94.4667 %
  - Incorrectly Classified Instances 83 5.5333 %
- STS evaluation:
  - Correctly Classified Instances 768 94.8148 %
  - Incorrectly Classified Instances 42 5.1852 %
- 10-fold evaluation on all the data:
  - Correctly Classified Instances 2211 95.7143 %
  - Incorrectly Classified Instances 99 4.2857 %

Decision Tree

- weka.classifiers.trees.J48 -C 0.25 -M 2
- 10-fold evaluation on training only data:
  - Correctly Classified Instances 1436 95.7333 %
  - Incorrectly Classified Instances 64 4.2667 %
- STS evaluation:
  - Correctly Classified Instances 779 96.1728 %
  - Incorrectly Classified Instances 31 3.8272 %
- 10-fold evaluation on all the data:
  - Correctly Classified Instances 96.7965 %
  - Incorrectly Classified Instances 3.2035 %

Grafting Decision Tree

- weka.classifiers.trees.J48graft -C 0.25 -M 2
- 10-fold evaluation on training only data:
  - Correctly Classified Instances 1443 96.2 %
  - Incorrectly Classified Instances 57 3.8 %
- STS evaluation:
  - Correctly Classified Instances 777 95.9259 %
  - Incorrectly Classified Instances 33 4.0741 %
- 10-fold evaluation on all the data:
  - Correctly Classified Instances 96.8398 %
  - Incorrectly Classified Instances 3.1602 %

Random Forest

- weka.classifiers.trees.RandomForest -I 10 -K 0 -S 1 (method for the Random Forest Learning)
- 10-fold evaluation on training only data:
  - Correctly Classified Instances 1436 97.6 %
  - Incorrectly Classified Instances 64 2.4 %
- STS evaluation:
  - Correctly Classified Instances 785 96.9136 %
  - Incorrectly Classified Instances 25 3.0864 %
- 10-fold evaluation on all the data:
  - Correctly Classified Instances 97.619 %
  - Incorrectly Classified Instances 2.381 %

To Do:

- Perform the same analysis for:
  - BF DT
  - LM tree
  - ANN

Analysis

- The evaluation scores increased for ALL the models!
- The best: Random Forest – 97.6%, even though it increased accuracy by the smallest amount
- Grafting DT and DT – 96.8%
- What about the rest of the models on your To Do list?
- Our chosen model: J48graft – 96.8% [+0.6398 %]
- The smallest increase: ~ +0.019 % (RF)
- The highest increase: ~ + 1.24 % (DT)

Stop?

- Yes
- Let's pick a model and fine tune the parameters
- LMT? Really slow to build.
- Another option?
- How about wJ48graft?
  - 96.8 % - really nice
  - Faster to build
  - Easier readability
  - J48graft vs. J48?

Grafting DT or DT?

- DT:
  - Number of Leaves : 39
  - Size of the tree : 77
- Grafting DT:
  - Number of Leaves : 141
  - Size of the tree : 281
- Choose: Decision Tree

The Model

```
region-centroid-row <= 155
|   rawred-mean <= 27.2222
|   |   hue-mean <= -1.89048
|   |   |   hue-mean <= -2.24632: foliage (160.0/1.0)
|   |   |   hue-mean > -2.24632
|   |   |   |   saturation-mean <= 0.772831
|   |   |   |   |   region-centroid-col <= 110
|   |   |   |   |   |   rawred-mean <= 0.666667
|   |   |   |   |   |   |   region-centroid-row <= 150: foliage (14.0/1.0)
|   |   |   |   |   |   |   region-centroid-row > 150: window (2.0)
|   |   |   |   |   |   rawred-mean > 0.666667
...
```

Next step

- Fine-tuning of the parameters, to see whether the results could improve

- Present the Final Fine-Tuned Tree to the Research Group

Conclusion

- Knowing that the end model needs to be user-friendly, we could have skipped modeling using ANN and Random Forest
- This would be different for some other tasks, where the risk of having a misclassified instance is too costly (e.g. patient diagnosis, part malfunction, etc.)

Let's Save Our Work

- First, save the current data file as IMAGE_AssignmentII_ready.arff, we will use it in Assignment II

Next

- Assignment II
- Next is Lesson 5: Another example of complex datasets and tasks in data mining