COMP7103 Data Mining

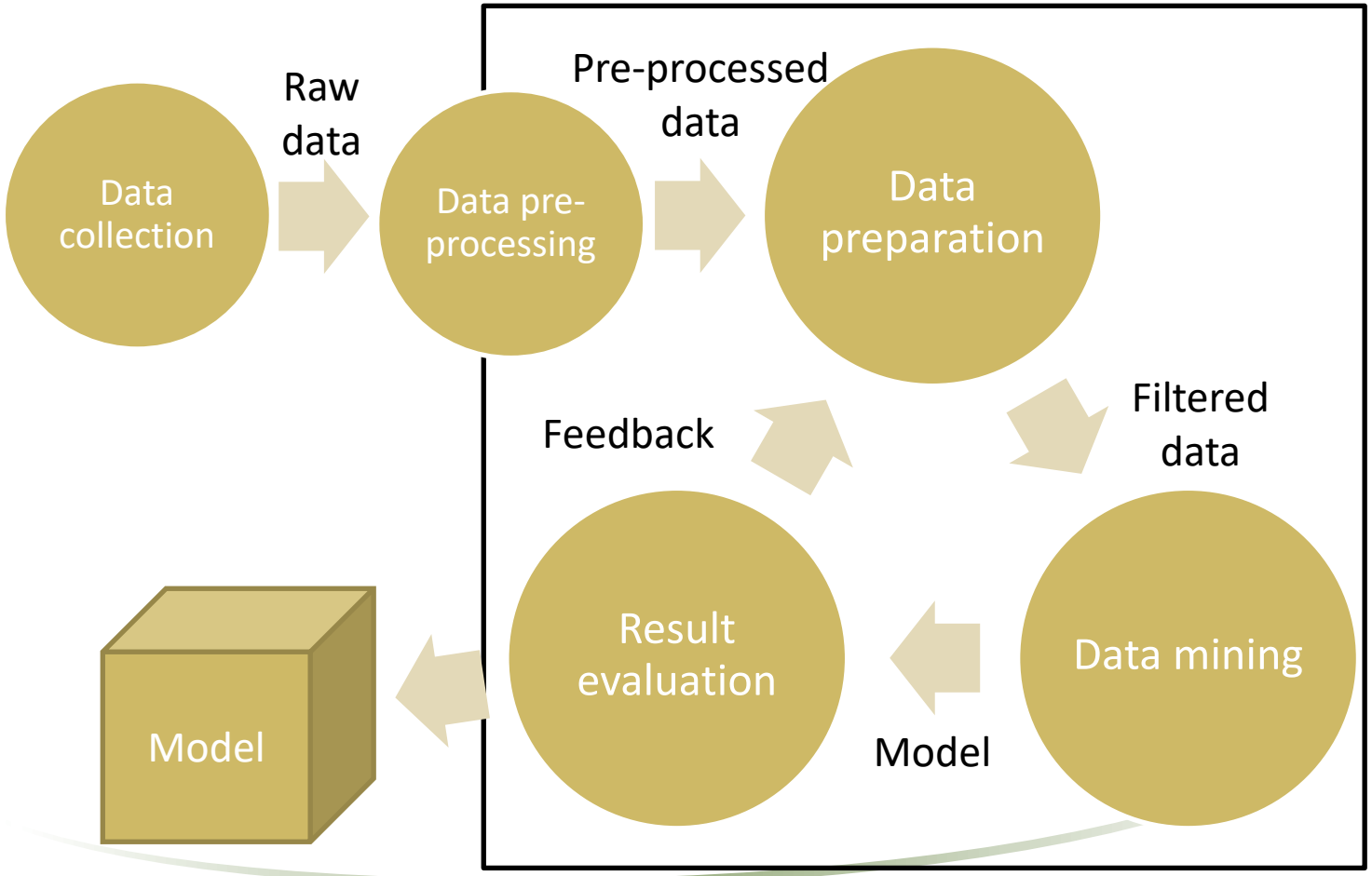
Tutorial 2

Weka classification / result evaluation

yklam2@cs2022

Objectives

- Evaluate classification result
- Understand how Weka works
- Parameter Selection in Weka
- Extra:
  - Get to know the side effect of filters and cross-validation

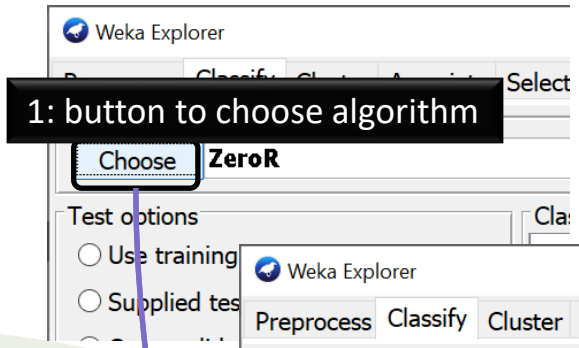


Data set

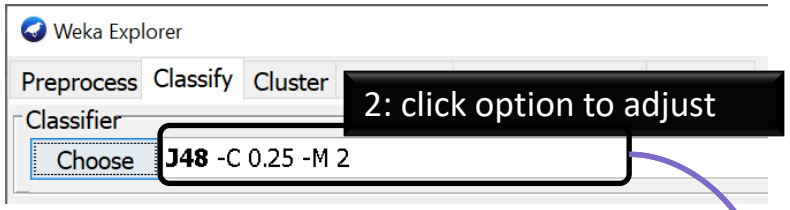
- We will be using the **Dry Bean Dataset data set** in UCI repository
  - https://archive.ics.uci.edu/ml/datasets/Dry+Bean+Dataset
  - 16 attributes + 1 class attribute

Available on Moodle:
`Dry_Bean_Dataset.arff`

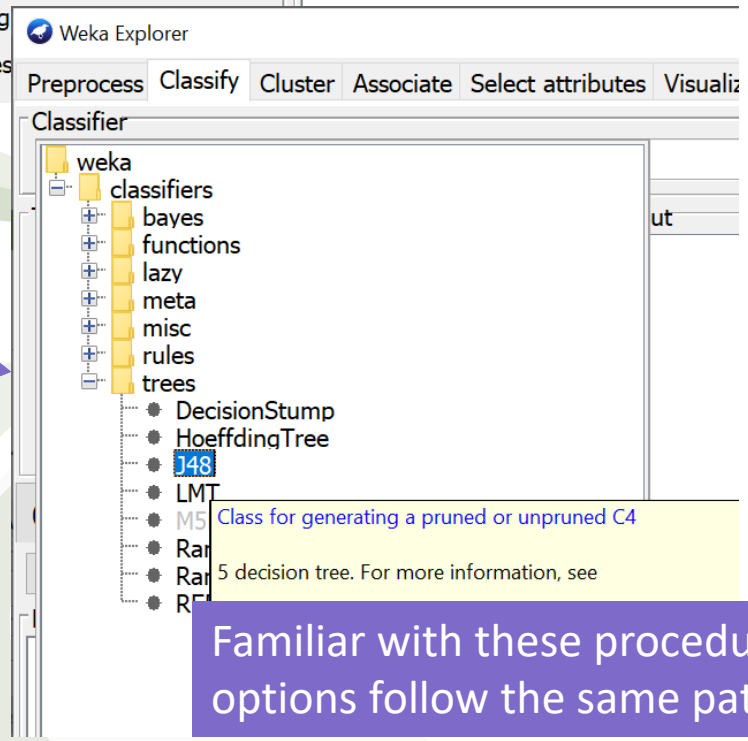
Configurating options in Weka



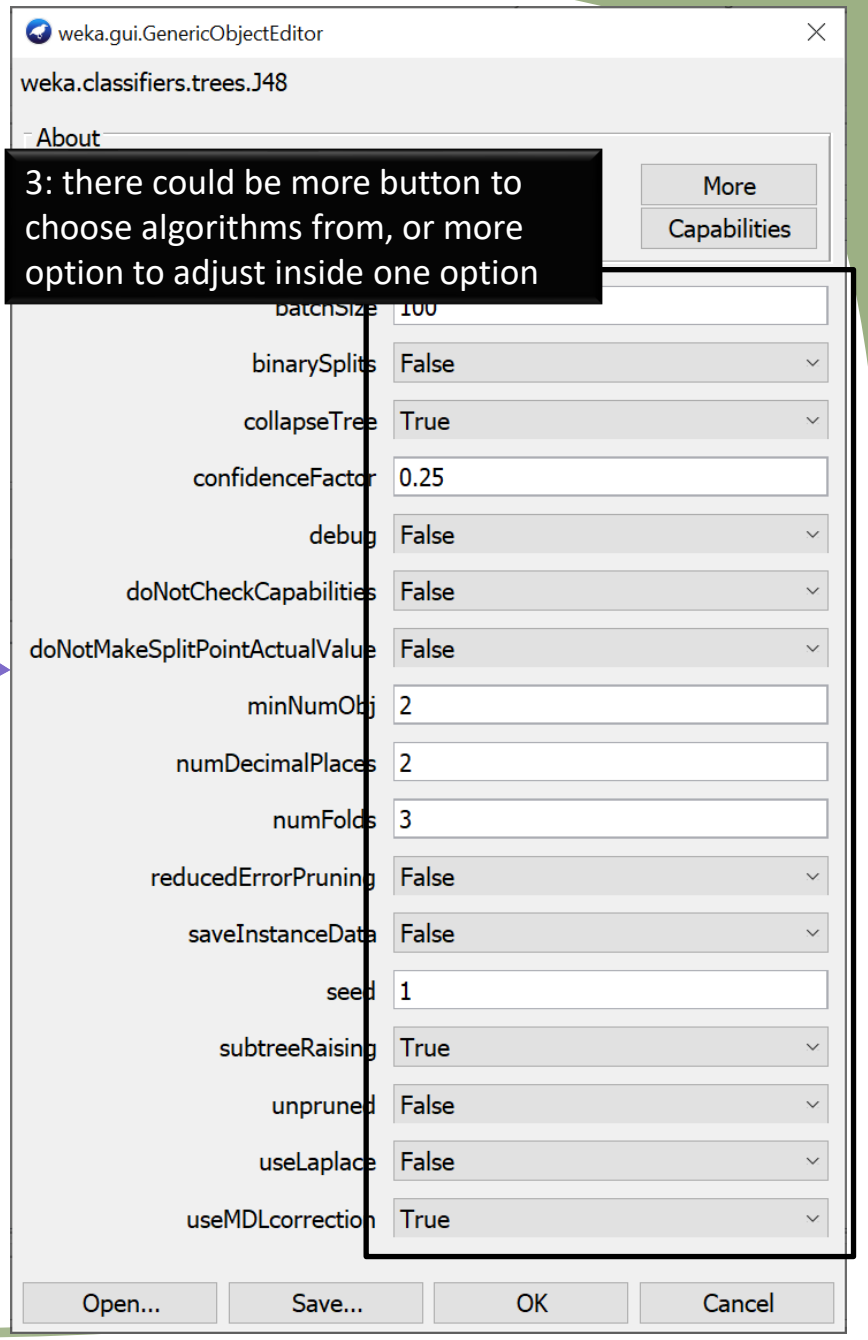
1: button to choose algorithm



2: click option to adjust

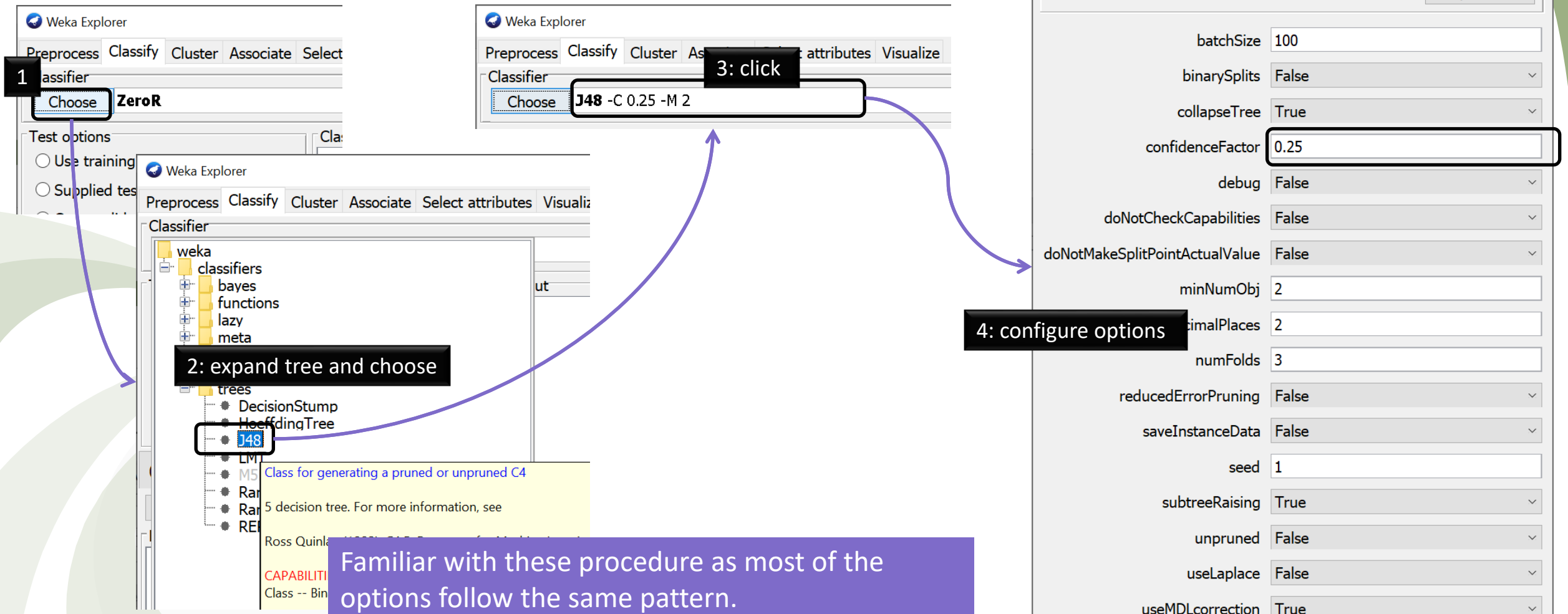


Familiar with these procedure as most of the options follow the same pattern.



3: there could be more button to choose algorithms from, or more option to adjust inside one option

Choosing classifier (the “Classify” tab)



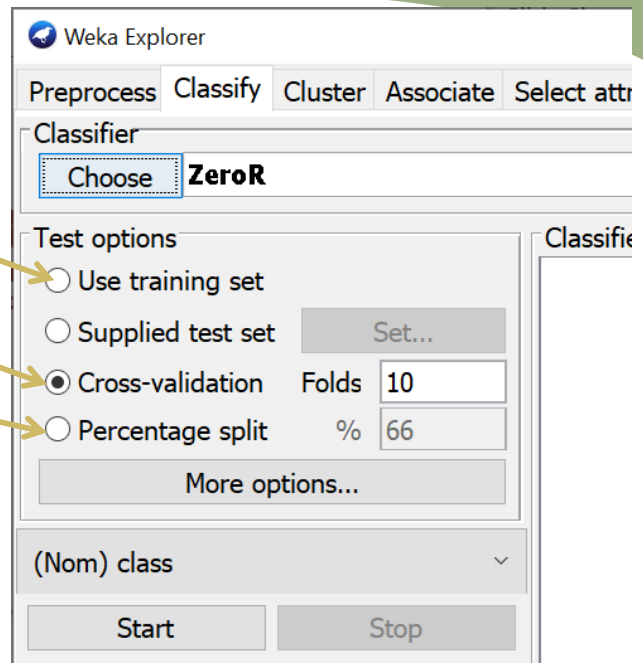
Familiar with these procedure as most of the options follow the same pattern.
This process maybe recursive, an option may allow you to pick an algorithm which you can adjust.

Result evaluation

NEVER use this!

CV

Holdout



- **Holdout**
  - Reserve some data for testing
- ***k*-fold Cross-validation**
  - Divide samples to k partitions, run classifier using $k-1$ partitions and test with the remaining one. Repeat the process for all combinations of $k-1$ partitions
- **Bootstrap**
  - Sample n instances with replacement as the training set, use those that is not sampled as the testing set
  - This could be done using the Bagging classifier (will not be covered in tutorial)

10-fold CV result (J48, C=0.25) - model

```
=== Classifier model (full training set) ===

J48 pruned tree
------------------

MajorAxisLength <= 328.843812
|   Compactness <= 0.860716
|   |   Perimeter <= 745.326
...
|   |   |   |   |   |   |   |   |   Extent <= 0.765244: CALI (13.0)
|   |   |   |   |   |   |   |   |   Extent > 0.765244
|   |   |   |   |   |   |   |   |   |   AspectRation <= 1.533909: BARBUNYA (4.0)
|   |   |   |   |   |   |   |   |   |   AspectRation > 1.533909: CALI (20.0/5.0)

Number of Leaves  :     259

Size of the tree :      517
```

This is the model built by the algorithm using full training set

Is it a good model?

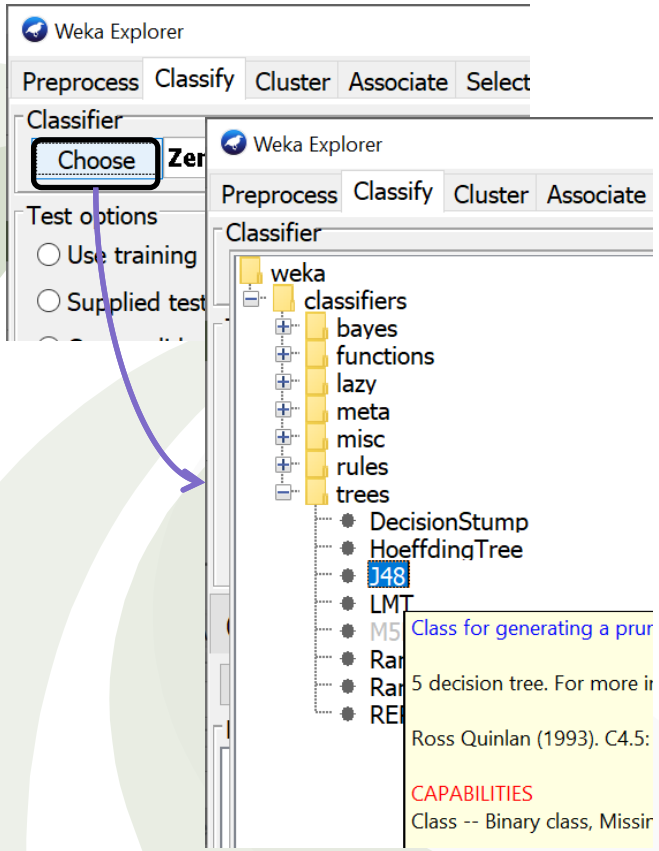
10-fold CV result (J48, C=0.25) – evaluation

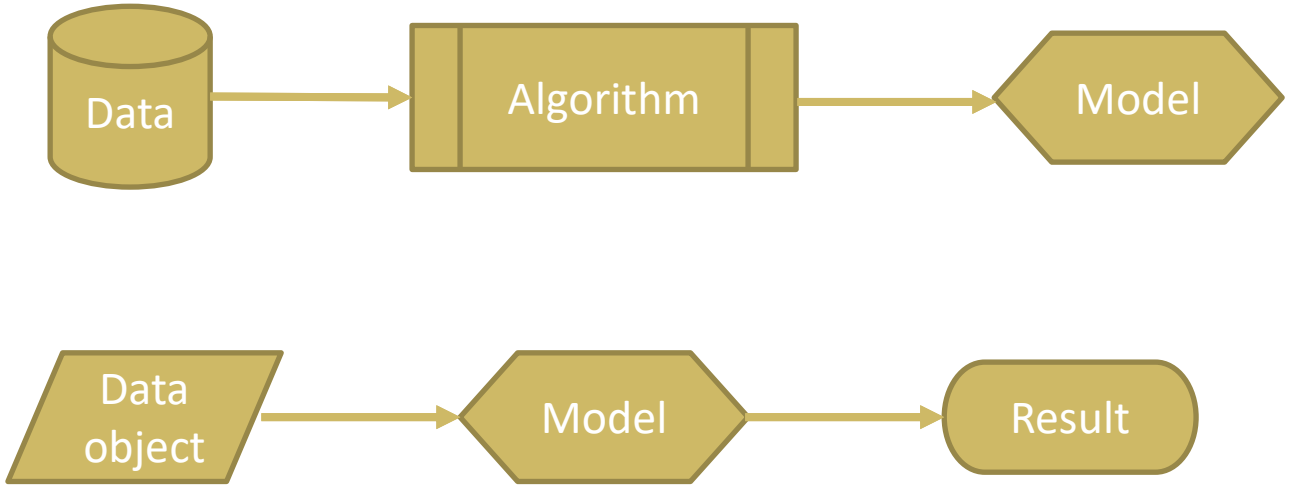
```
Correctly Classified Instances       12429               91.3158 %
Incorrectly Classified Instances      1182                8.6842 %
Kappa statistic                          0.8949
Mean absolute error                      0.0309
Root mean squared error                  0.1466
Relative absolute error                 13.0541 %
Root relative squared error             42.6569 %
Total Number of Instances            13611

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.941    0.011    0.938      0.941    0.939      0.929    0.977     0.937     SEKER
                 0.885    0.009    0.913      0.885    0.899      0.888    0.951     0.857     BARBUNYA
                 0.996    0.000    0.998      0.996    0.997      0.997    0.998     0.993     BOMBAY
                 0.923    0.011    0.918      0.923    0.920      0.910    0.970     0.879     CALI
                 0.941    0.009    0.943      0.941    0.942      0.933    0.973     0.900     HOROZ
                 0.857    0.032    0.864      0.857    0.860      0.827    0.946     0.834     SIRA
                 0.918    0.034    0.904      0.918    0.911      0.879    0.972     0.897     DERMASON
Weighted Avg.    0.913    0.020    0.913      0.913    0.913      0.893    0.967     0.889

=== Confusion Matrix ===

    a    b    c    d    e    f    g   <-- classified as
 1907   13    0    1    0   52   54 |    a = SEKER
   19 1170    1   89   16   27    0 |    b = BARBUNYA
    0    1  520    1    0    0    0 |    c = BOMBAY
    2   72    0 1504   39   13    0 |    d = CALI
    0   10    0   37 1815   45   21 |    e = HOROZ
   43   15    0    6   44 2258  270 |    f = SIRA
   62    0    0    0   10  219 3255 |    g = DERMASON
```

For each Classify run in Weka, a model and an evaluation result will be produced. **How does it work?**

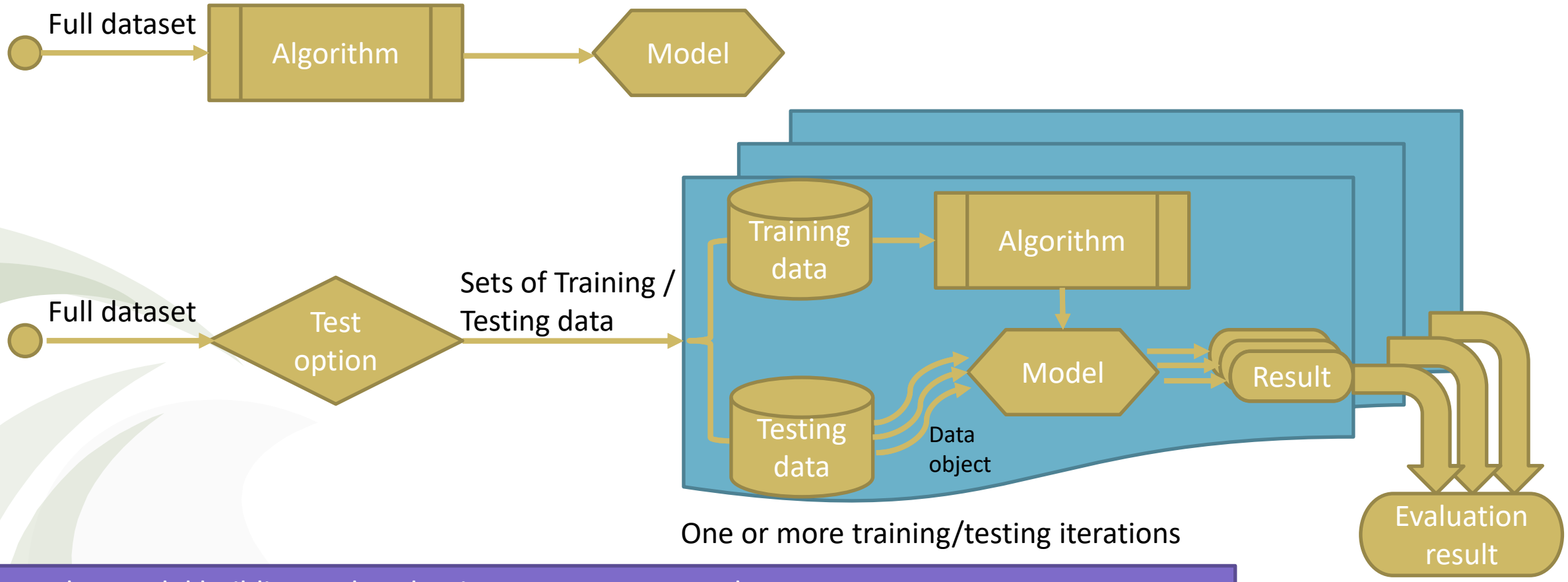
Algorithm and model in Weka

- Every algorithm in Weka takes data and produce a model.
- A classifier model takes one instance of data and produce a classification result



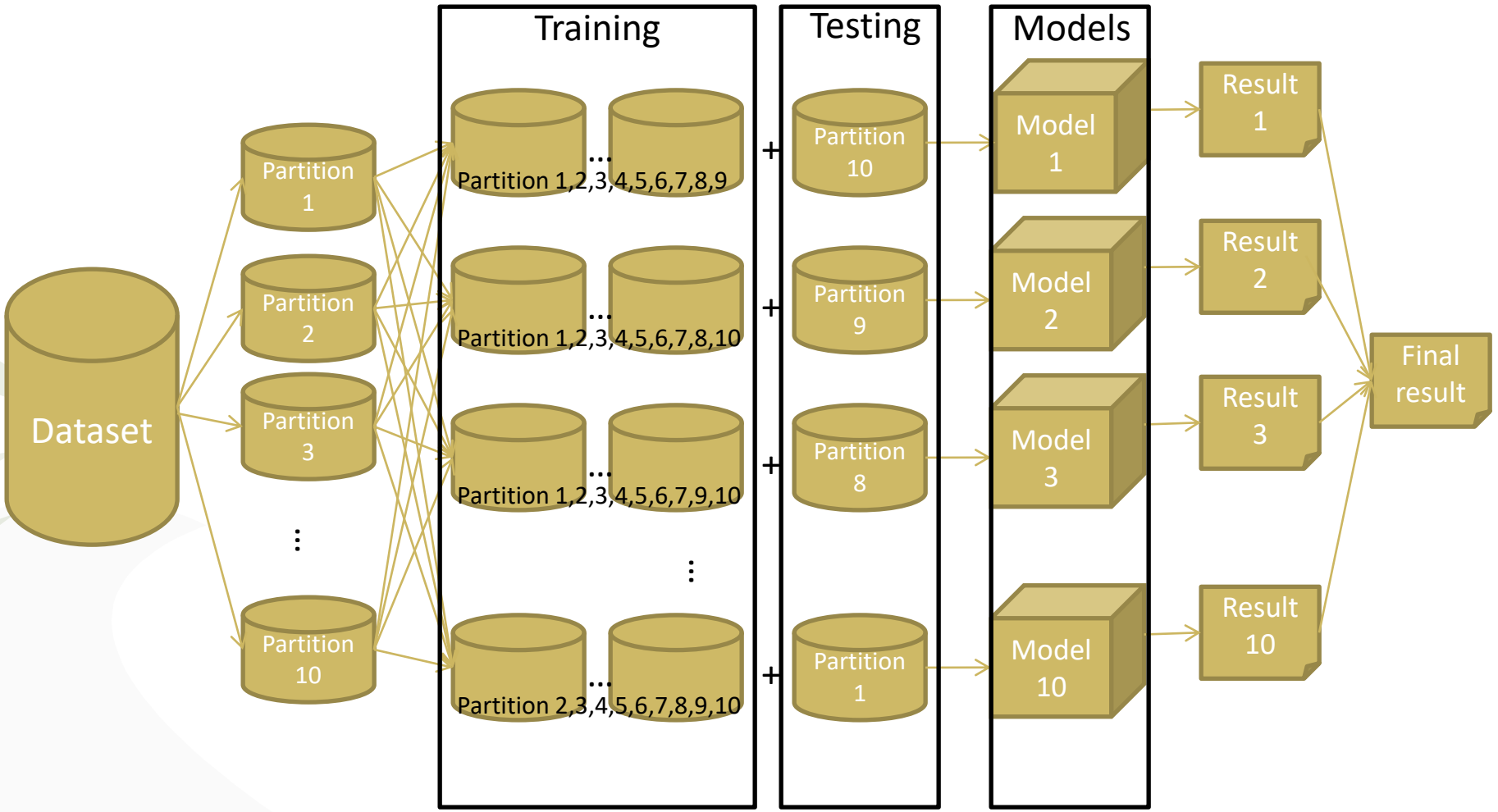


Procedure in Weka



In Weka, model building and evaluation are two separated process.
Model building: Always use full set of data
Evaluation: Split data into training/testing sets for one or more iterations of training/testing, then combine the result.

Cross-validation illustrated



Quick question

- Consider the previous 10-fold CV result (J48, C=0.25)
  - How many models are built in the process of evaluation in Weka?
  - How many models are built throughout the whole process in Weka?

Result good?

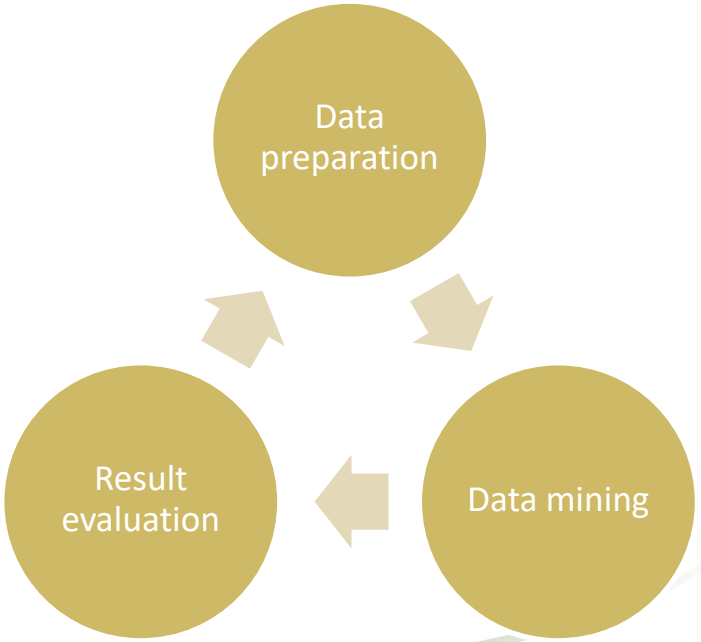
- We cannot determine whether a model is good enough based on one single evaluation.
- **Comparison** is always needed.
  - Comparison allows validation of the process
    - Very good result may be caused by overfitting!
  - Comparison requires **different algorithms** to be evaluated.
  - Changing test option does not count -- why?

If we get different results under different test options on the same algorithms, we better go pessimistic.

Parameter selection

- When testing different algorithms, we may also tune the parameters to produce better models. This is a **parameter selection** process.
  - How many attributes should be tuned?
  - Which attribute should be used?
  - What are the values of the parameters should be used?

- It is a **trial-and-error** process
  - Repeat data mining process to fine tune parameters.



Example: Picking the "C" of J48

- The J48 classifier takes one parameter, '**C**', to control the resulted tree size.
  - The smaller the value, the smaller the tree.
  - A larger tree may be more accurate, but it may be overfitting
  - A smaller tree may be less accurate, but it may be more predictive
- Here are the results of running J48 on the dataset, using 10-fold cross-validation, with C=0.1, 0.2, …, 0.5

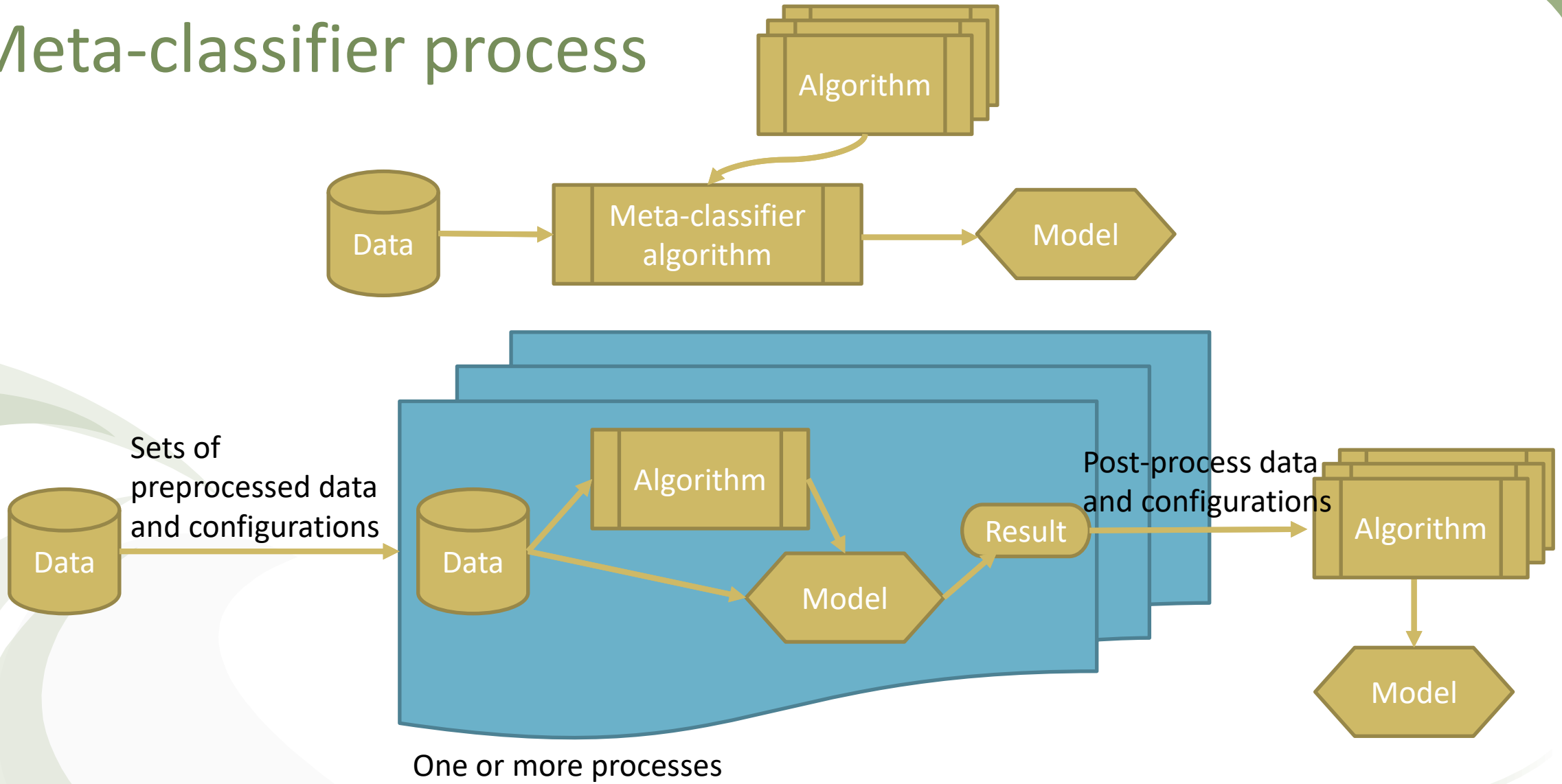
C	Accuracy	F-Measure(avg)
0.1	91.4261%	0.914
0.2	91.2424%	0.912
0.3	91.3158%	0.913
0.4	91.1763%	0.912
0.5	91.066%	0.911

Which one will you choose?
How about 0.25?

meta-classifier

- Weka provides a set of **meta-classifiers** that combine tools with existing classifiers.
- For example, the **CVParameterSelection** (cross-validation parameter selection) meta classifier allow you to run a parameter selection for any classifier.
  - It evaluate different parameter values on models build by a classification algorithm using cross-validation.
  - Parameter that generate a model with **best accuracy** will be chosen.
- Other useful meta-classifiers includes:
  - Bagging (Bootstrapping)
  - Vote (Ensemble)

Meta-classifier process



Meta classifiers works with one or more algorithms and produce a model.
As it also takes data and produce a model, a meta-classifier can be used as an algorithm in another meta-classifier.

CVParamterSelection in Weka

Meta-classifier: Algorithm that works with other classifier(s)

Parameter to test: “C”
From: “0.1”
To: “0.5”
Number of tests: “5”
(You can pick more if you like)

Click and change to Classifier: **J48**

Final option

Result

- C=0.1 is picked.

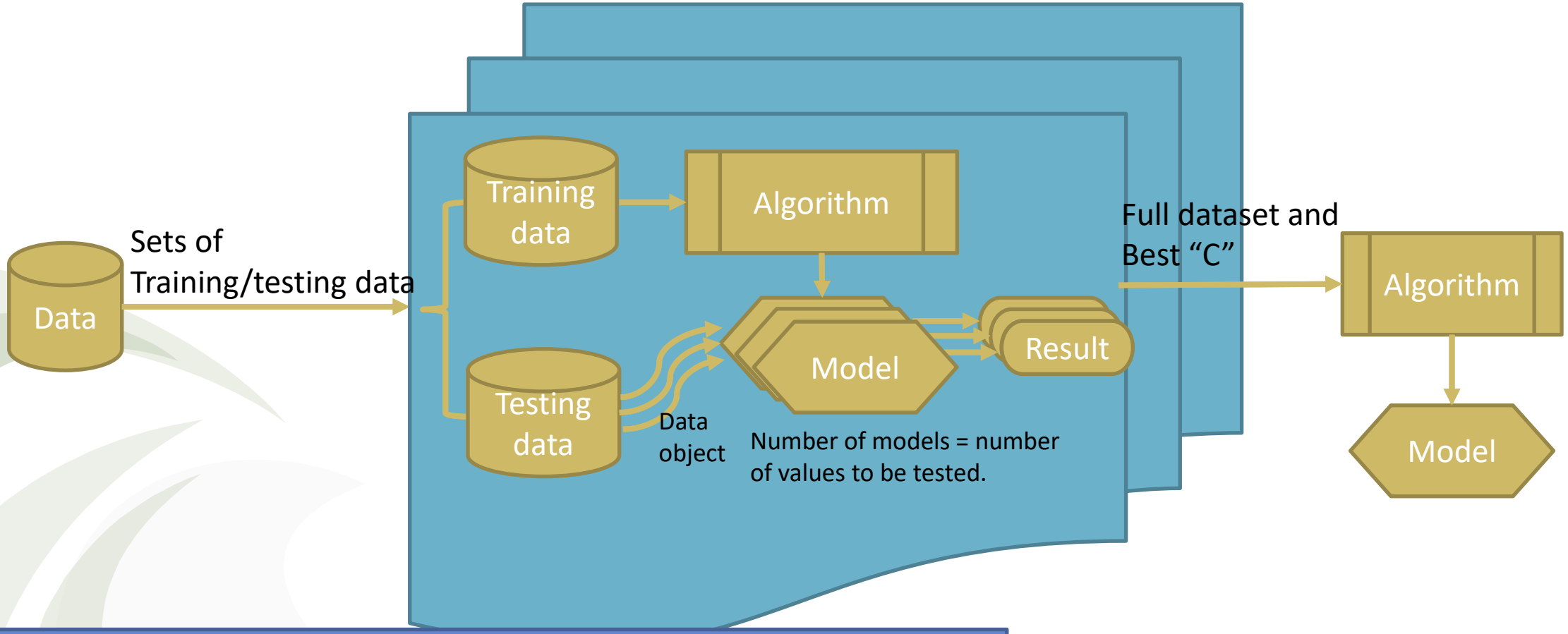
```
=== Classifier model (full training set) ===

Cross-validated Parameter selection.
Classifier: weka.classifiers.trees.J48
Cross-validation Parameter: '-C' ranged from 0.1 to 0.5 with 5.0 steps
Classifier Options: -C 0.1 -M 2
```

- With 10-fold CV as test option, accuracy = 91.3967%
- Remember the result we have got previously?

C	Accuracy	F-Measure(avg)
0.1	91.4261%	0.914

- Why is it different? What is really evaluated?
  - Think about how Weka works again.

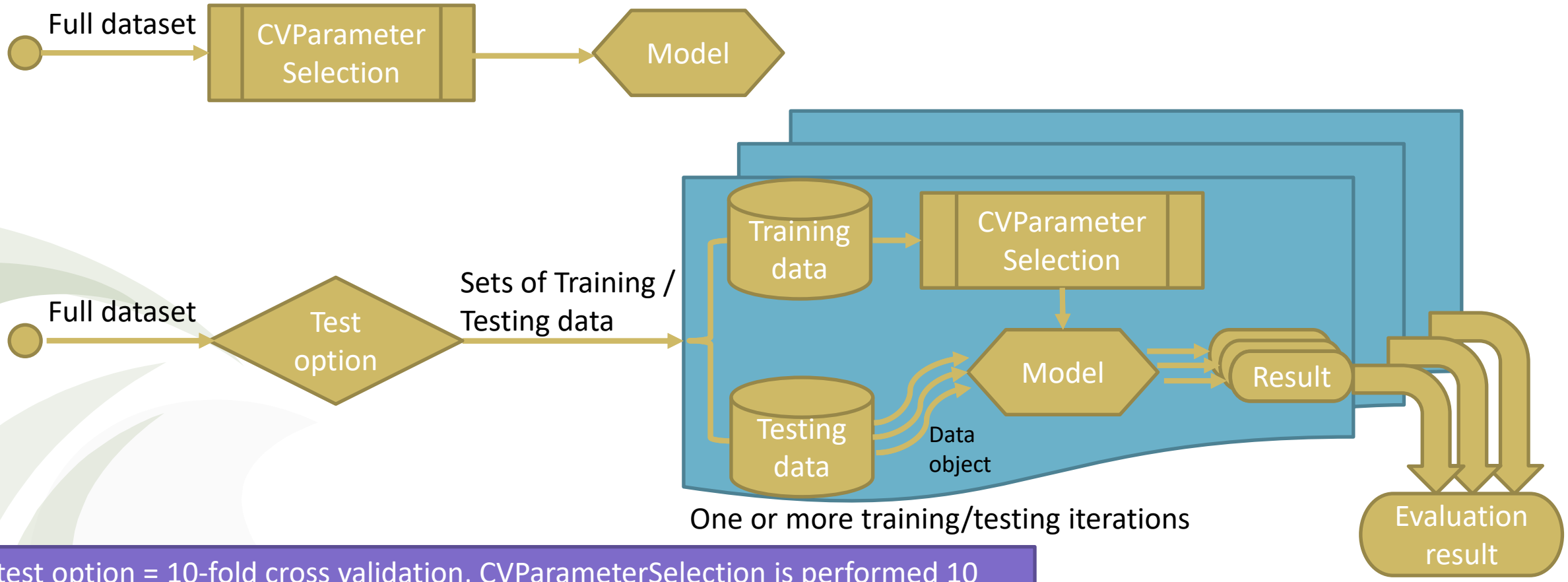
CVParameterSelection process (C = 0.1, 0.2, … 0.5)



CVParameterSelection performs cross-validation and collect evaluation result of every "C". It then pick a "C" and build a model using all data.

How many models are built in this process?

Full procedure in Weka (CVParameter Selection)



With test option = 10-fold cross validation, CVParameterSelection is performed 10 times in the evaluation process (plus one time in the model building process)

Why so complicated? This ensure that models are always evaluated using unseen testing data. This includes models in the parameter selection process.

How many models are built in the full process?

Final words on “test option”

- The purpose of evaluation is to **estimate** the performance of models built by an algorithm.
- “Test option” in Weka only affects the evaluation result.
  - You get a different estimation with a different test option. You are still evaluating the same algorithm.
- Thus, changing test option does not count as “trying different algorithms” in a data mining process.

Get to know the side effect of filters and cross-validation

Extra demo (self-read)

Case study

- This is important to understand what you are using in the data mining process.
- When you get an exceptionally good result, double check that your process is correct.
- The coming demo illustrate a process using the **resample** filter.
  - This filter usually gives very good evaluation result, **but the result is misleading**.

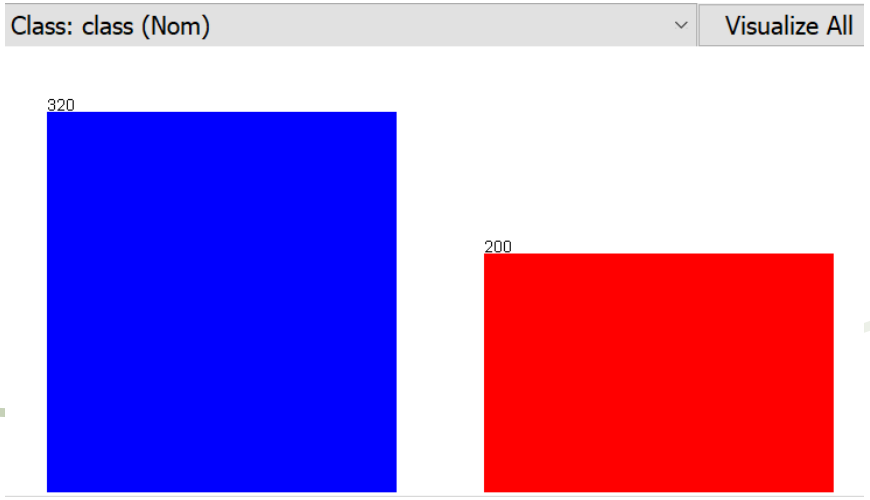
- **Warning: this demo shows a process that is incorrect. DO NOT attempt this in assignment.**

We will be using one of the data set from the **Early stage diabetes risk prediction dataset** in UCI repository
https://archive.ics.uci.edu/ml/datasets/Early+stage+diabetes+risk+prediction+dataset

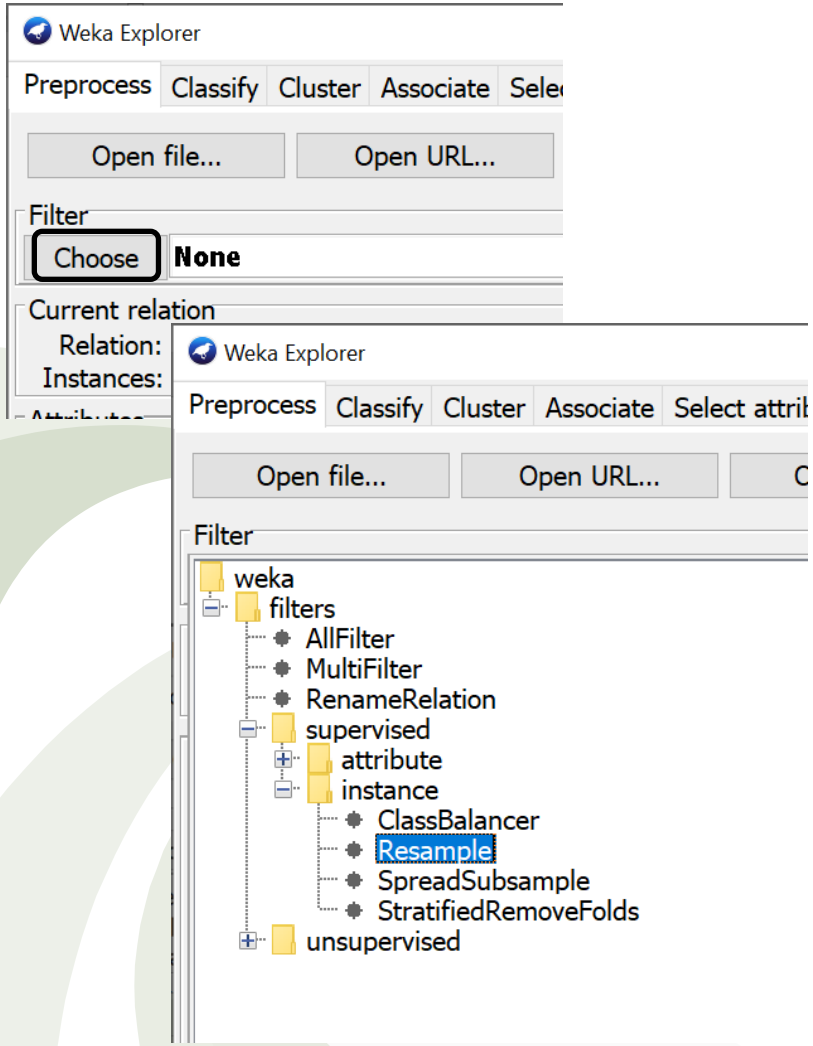
Available on Moodle:
`diabetes_data_upload.csv`

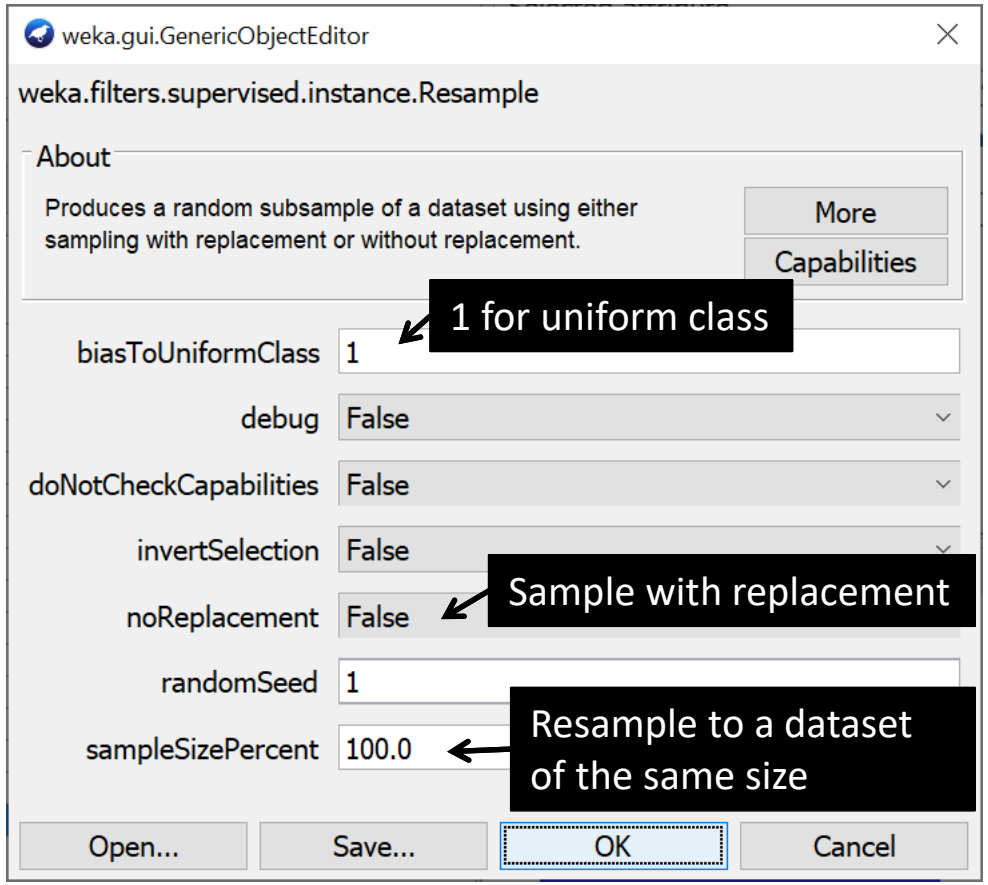
The motivation

- Weka provides many useful filter that you can use in the “Preprocess” section.
- Its common to use filters to process our data.
  - As part of data preprocessing
  - To reduce dimension
- The resample filter is a very commonly **misused** example.
  - Why is it used? It’s an attempt to balance instances that support different classes in the dataset.
  - If the dataset has an imbalanced class distribution, there will be a chance that the resulting model favors the majority class.

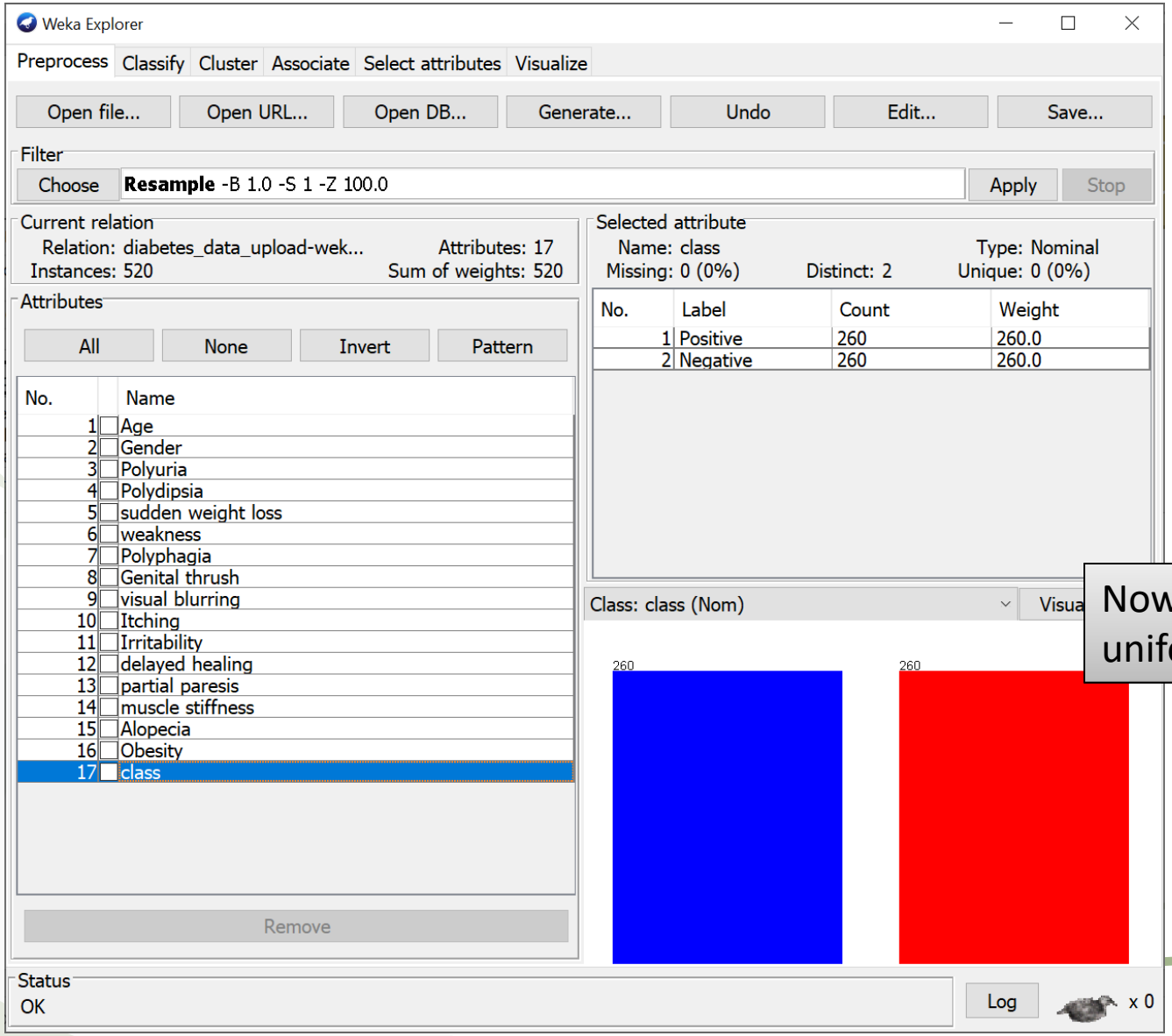


Resample filter (supervised)





Filtering result



Now, we have a uniform class

Result comparison

- Let's compare the result of J48 (cross-validation) before and after resampling (C=0.25)

Great improvement! But…

Before resample

```
Correctly Classified Instances          499               95.9615 %
Incorrectly Classified Instances         21                4.0385 %
…
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.950    0.025    0.984      0.950    0.967      0.916    0.966     0.975     Positive
                 0.975    0.050    0.924      0.975    0.949      0.916    0.966     0.910     Negative
Weighted Avg.    0.960    0.035    0.961      0.960    0.960      0.916    0.966     0.950
…
```

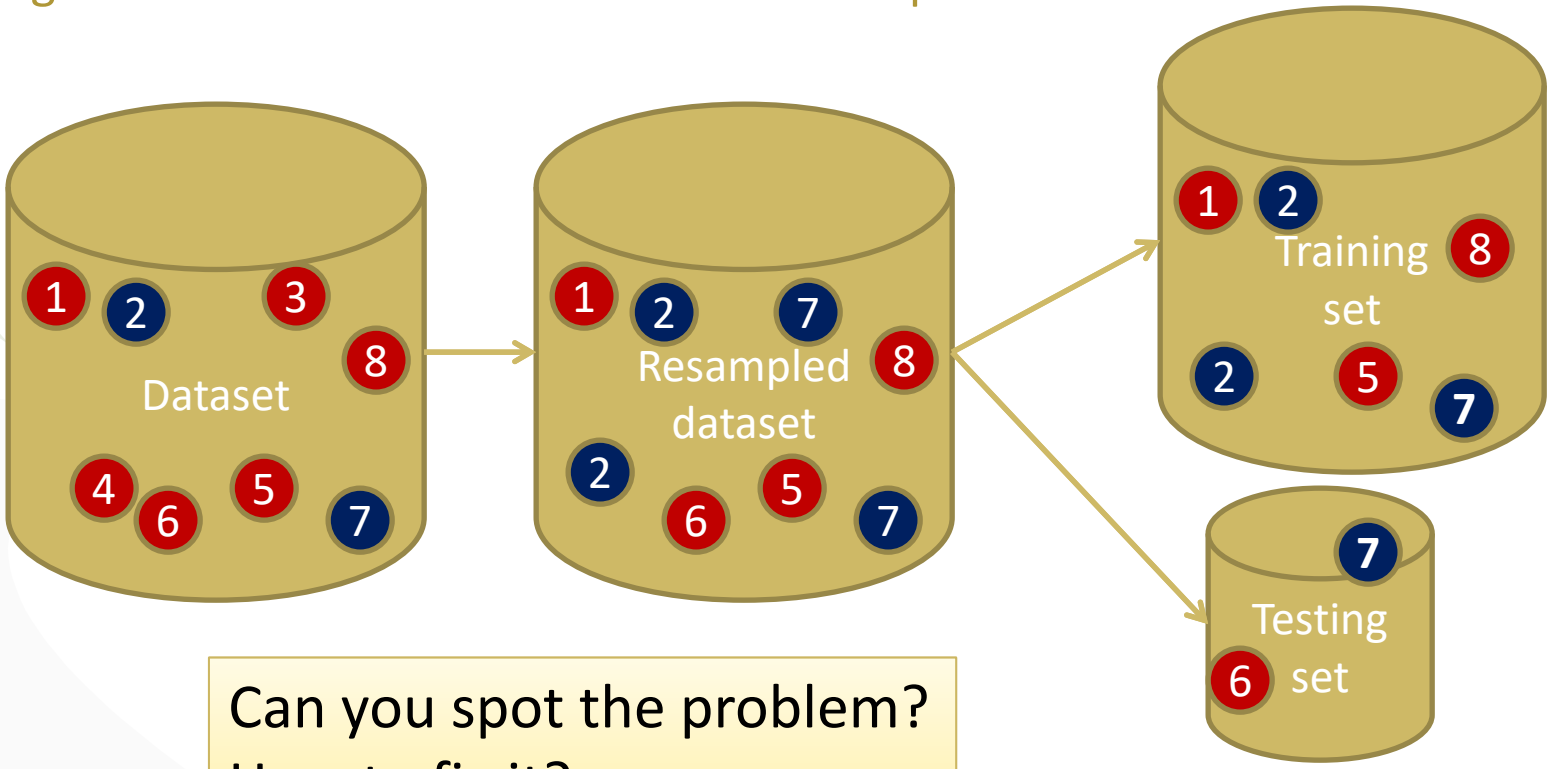
After resample

```
Correctly Classified Instances          507               97.5    %
Incorrectly Classified Instances         13                2.5    %
…
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.954    0.004    0.996      0.954    0.974      0.951    0.979     0.985     Positive
                 0.996    0.046    0.956      0.996    0.976      0.951    0.979     0.958     Negative
Weighted Avg.    0.975    0.025    0.976      0.975    0.975      0.951    0.979     0.971
…
```

Wait, is there anything wrong here?

Something wrong…

- Let's review what we have done
  - Considering one of the folds in the cross-validation process…

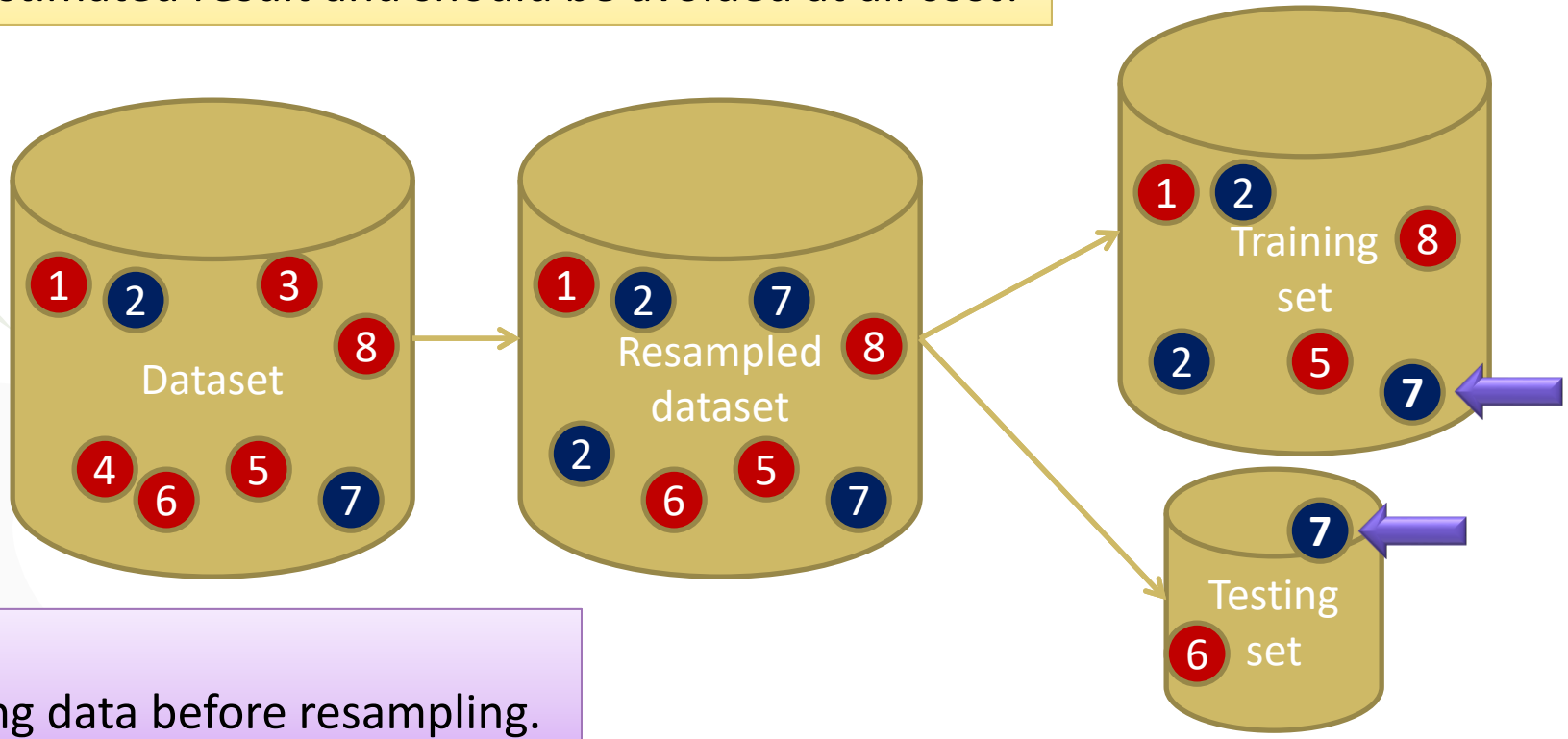


Can you spot the problem?
How to fix it?

Something wrong…

Make sure you understand the side effects of the algorithm you use.
Be careful not to overfit your model accidentally.
If you suddenly see a great improvement, double check what you have done.

Some instance of data is used in both training and testing.
This gives overestimated result and should be avoided at all cost!



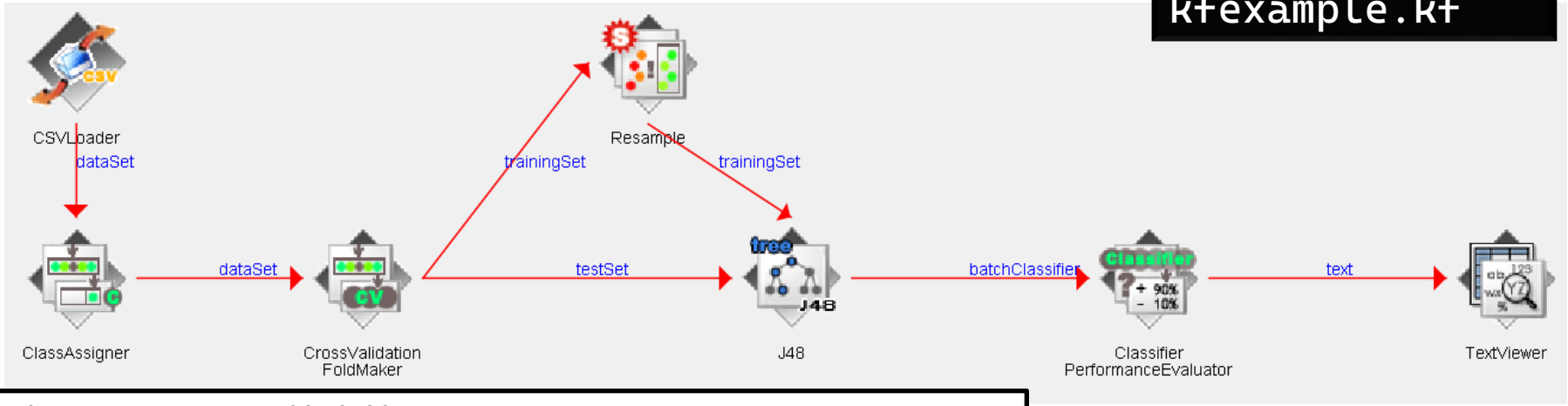
Solution?
Split training/testing data before resampling.

Unfortunately, this cannot be done in Weka explorer.
You may write your own program to do this, or consider using the **Weka Knowledge Flow**

Weka Knowledge Flow

- Weka Explorer provide simple UI to perform linear workflow.
- For more sophisticated workflow, you may use the knowledge flow.
  - This will not be covered in the tutorial due to time limitation.
- Here is an example:

Available on Moodle:
`kfexample.kf`




```
Correctly Classified Instances          487               93.6538 %
Incorrectly Classified Instances         33                6.3462 %
…
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.922    0.040    0.974      0.922    0.947      0.870    0.937     0.952     Positive
                 0.960    0.078    0.885      0.960    0.921      0.870    0.937     0.848     Negative
Weighted Avg.    0.937    0.055    0.939      0.937    0.937      0.870    0.937     0.912
…
```