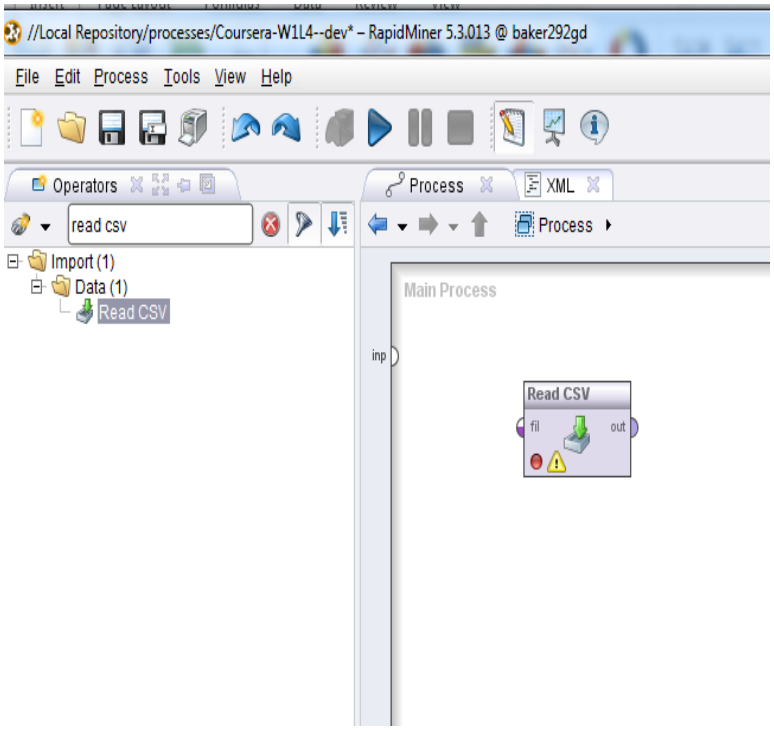
RapidMiner walkthrough

1. Install RapidMiner 5.3 from
   http://sourceforge.net/projects/rapidminer/files/1.%20RapidMiner/5.3/

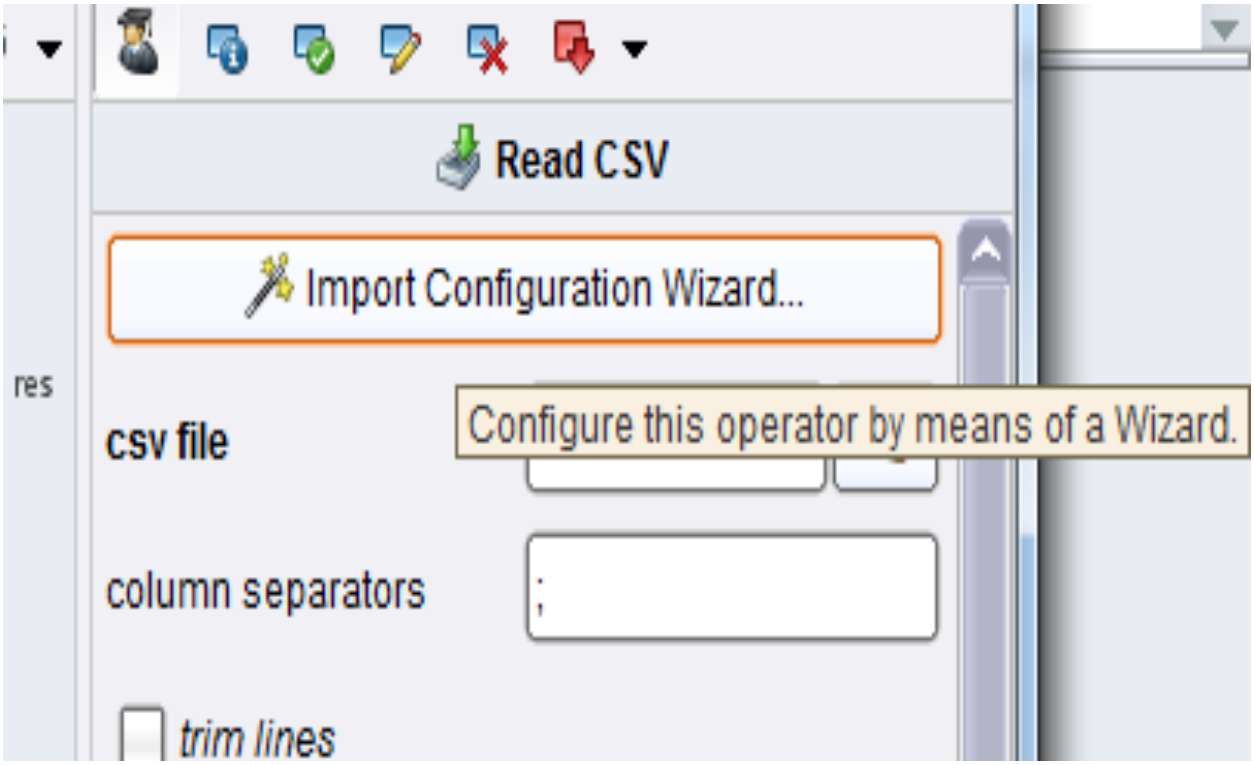
   I recommend using this version, since the assignments were written with this version, and there are small changes. (In practice, the differences between versions are small; I typically work with a version one before the current version, to get full open-source.)

2. Open RapidMiner 5.3 and open a new process

3. Type Read CSV into the operator box to create a new “Read CSV” Operator



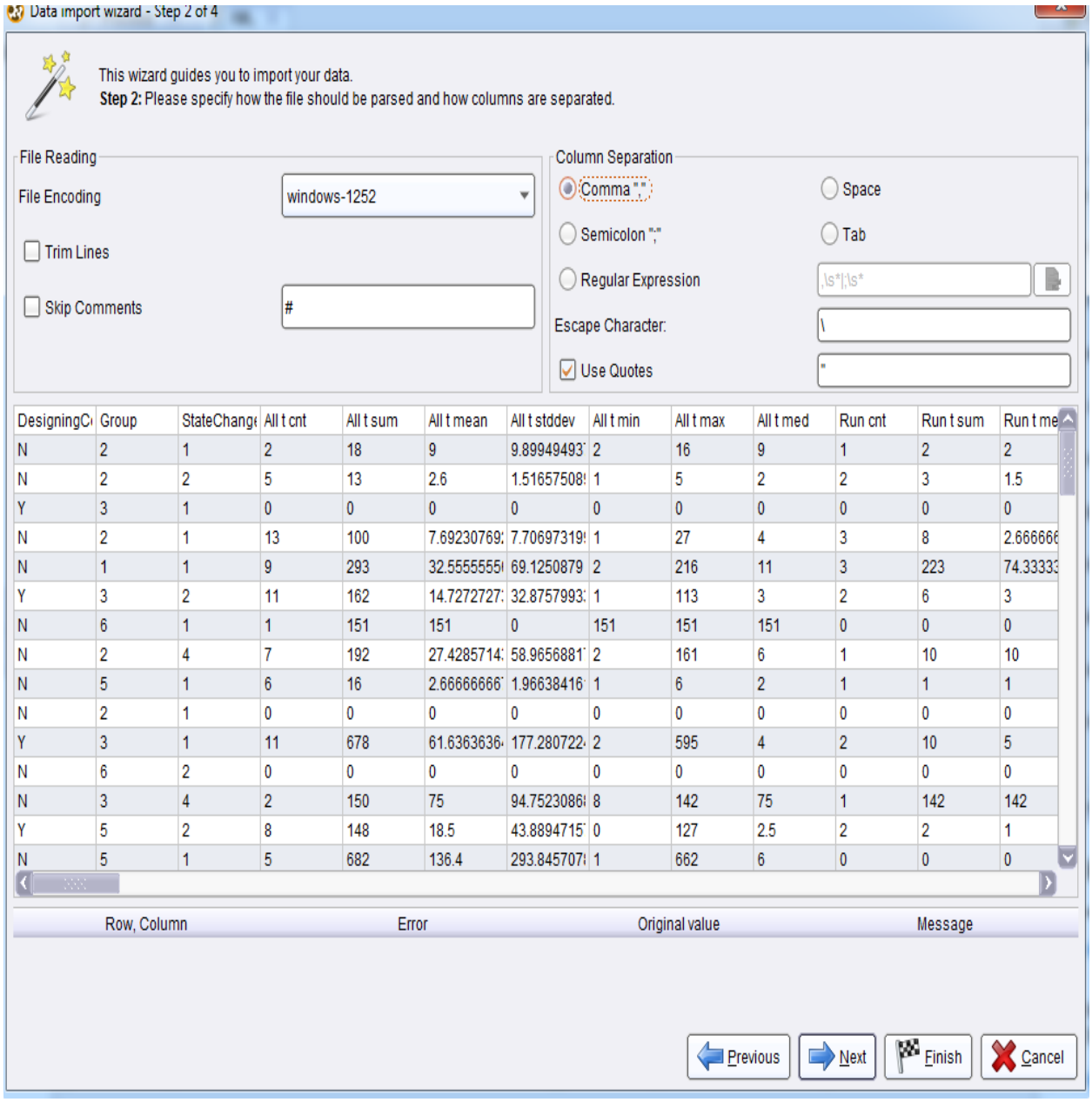
4. Click on the Import Configuration Wizard on the right side of the interface



5. Select file
   “SaoPedroetal(2013)_UMUAI_DesigningControlledExperiments_cummandlocalfeatures.csv”

   You will have to download it from the course webpage

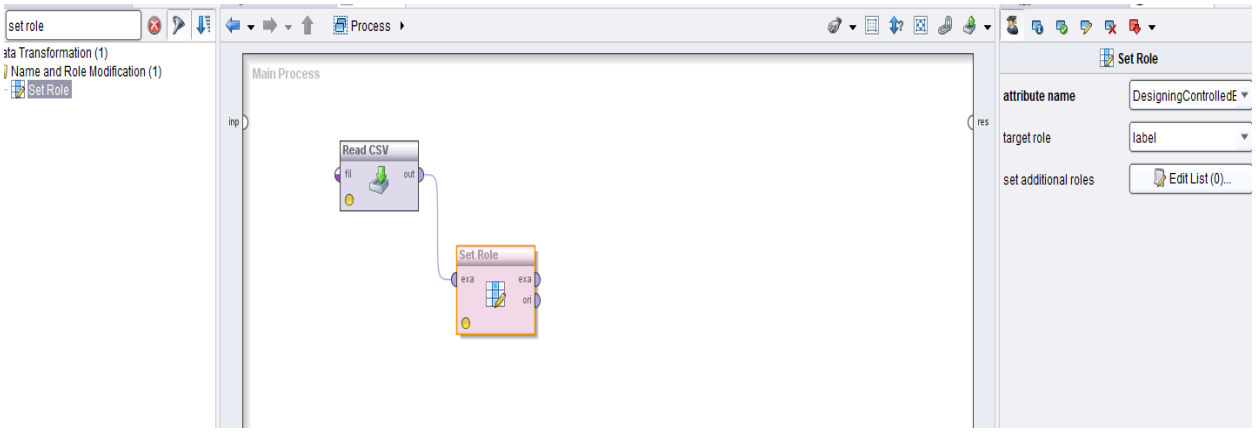
6. This is a “csv” file, so select “Comma Delimited”



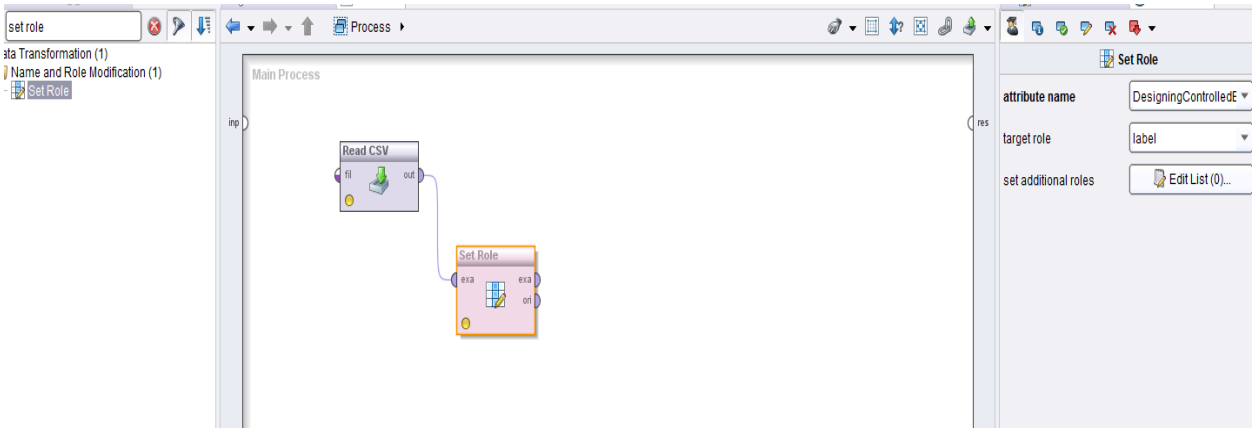
7. Click Next until the system does not let you click Next anymore. Then click Finish.

8. Create a “Set Role” operator in the operator box at the top-left.

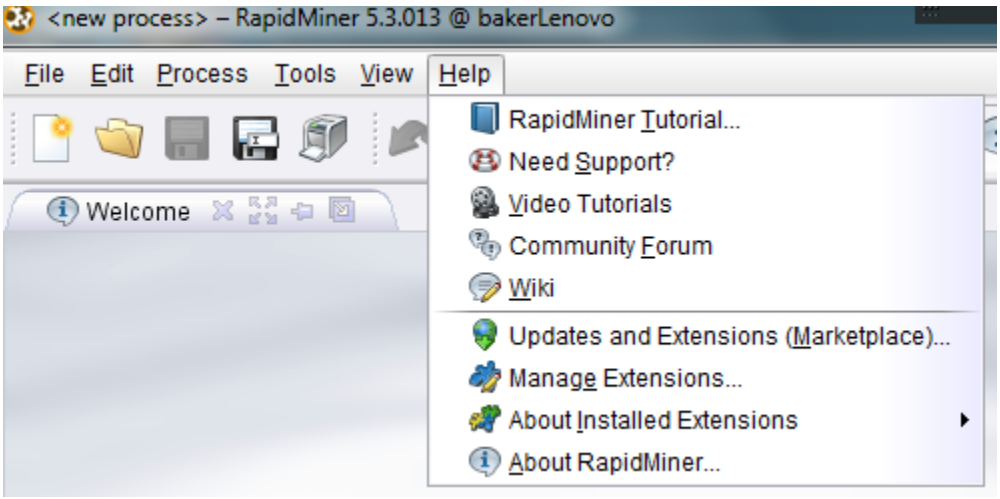
Then connect the output bubble on the right side of “Read CSV” to the input bubble on the left side of “Set Role” by clicking on the output bubble and then clicking on the input bubble. Your screen should look like this.

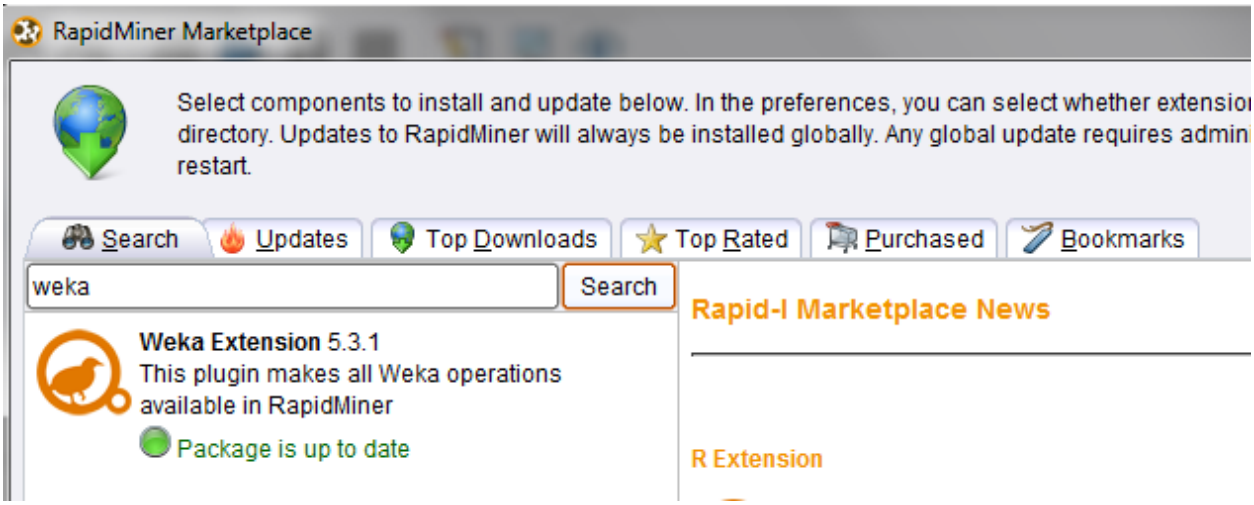


9. Now go over to the right side and select DesigningControlledExperiments as the variable you want to change, and set it to be a “label” in the target role box.

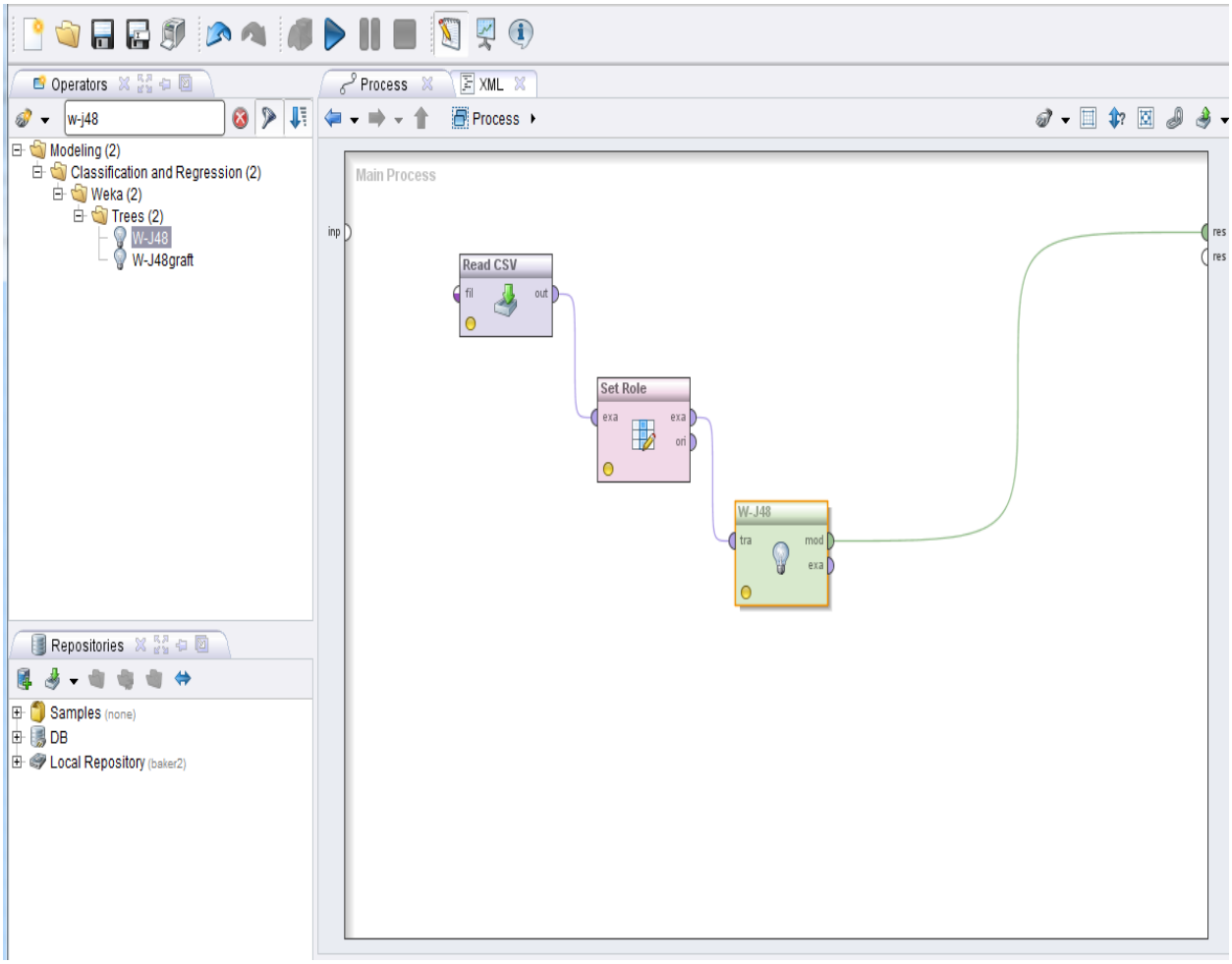


10. Install the WEKA Expansion Pack. To do this go to the Help menu, and select Updates and Extensions (Marketplace). Search for Weka, and install the Weka Expansion Pack.

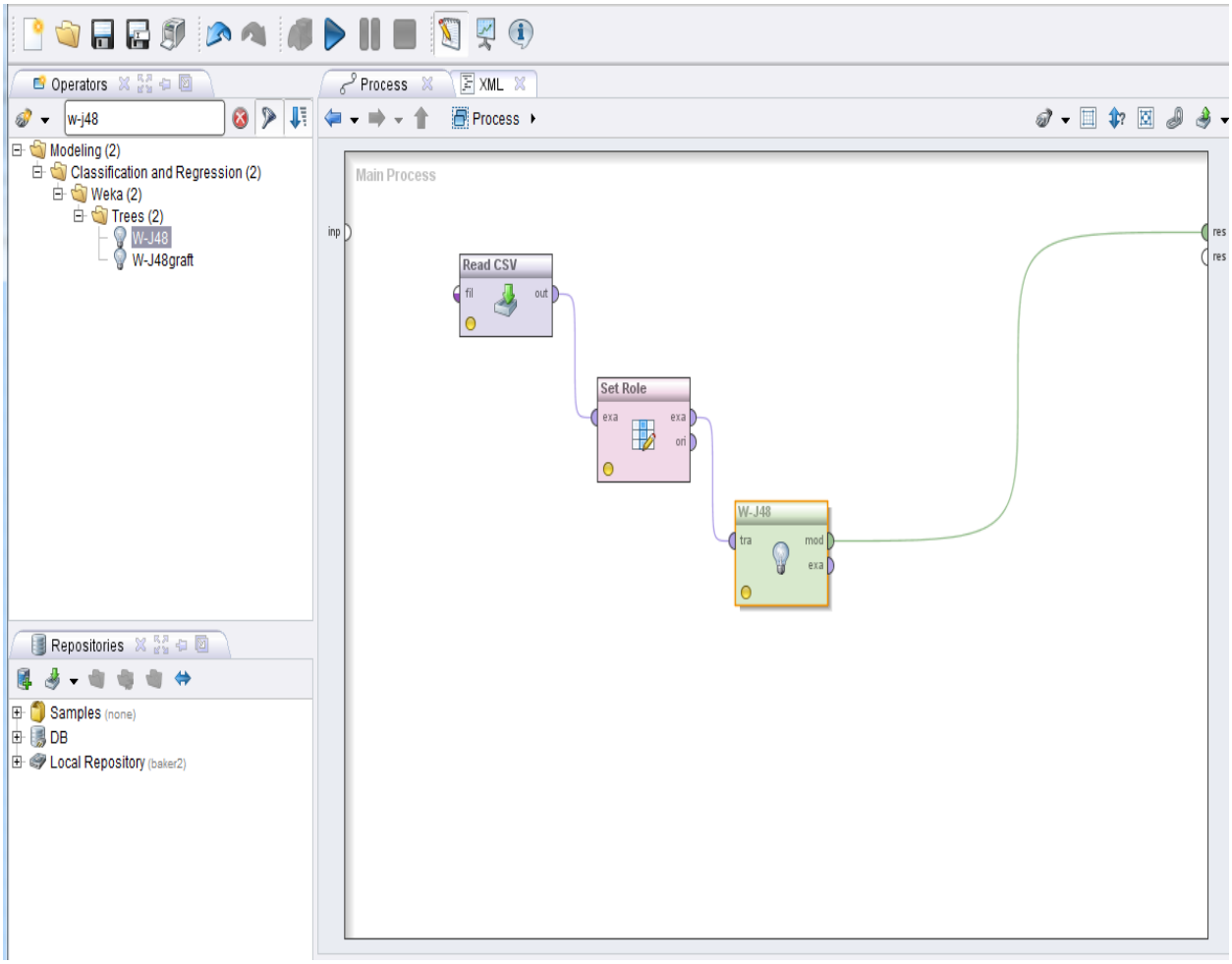




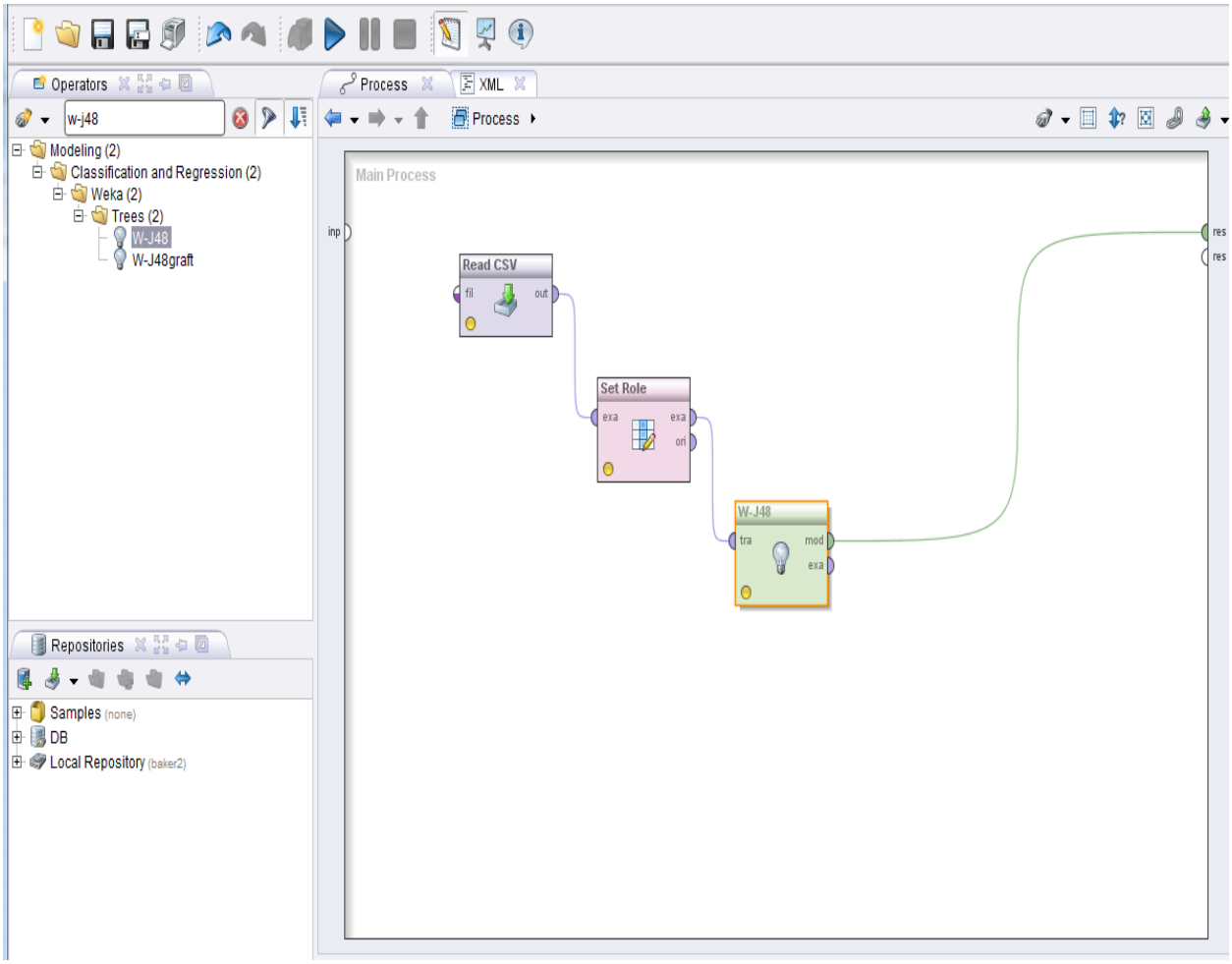
11. Type w-j48 into the operators window, and create the w-j48 operator



`12. Now connect the output bubble from Set Role (exa for example set) to the input bubble from J48 (tra for training set)



13. Then connect the output bubble from W-J48 (model) to the res (result) bubble on the far right



14. Then press play at the top of the screen. After a minute or so (possibly longer for slower computers), you should see your model

15. This representation shows how the model makes decisions. You can read it as follows:

If the variable CM cvs cnt is less than or equal to zero, then the model predicts No.

In the original data set, there were 271 cases where this prediction was correct, and 2 cases where it was wrong. So the confidence of this prediction is (271)/(271+2) = 271/273 = 99.27%.

If the variable CM cvs cnt is greater than zero, then the model goes to the next variable.

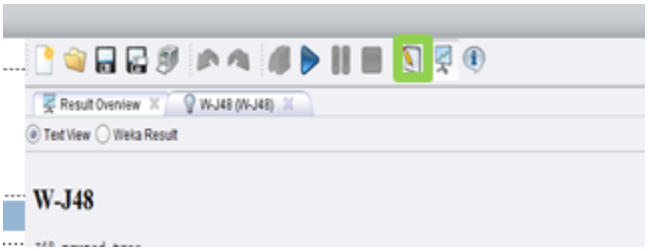
If the variable CVS ct is less than or equal to zero, then
If the variable Run T Sum is less than or equal to 11, then
about 11 other things,

to finally get to a prediction of No with 10/11 = 90.9% confidence

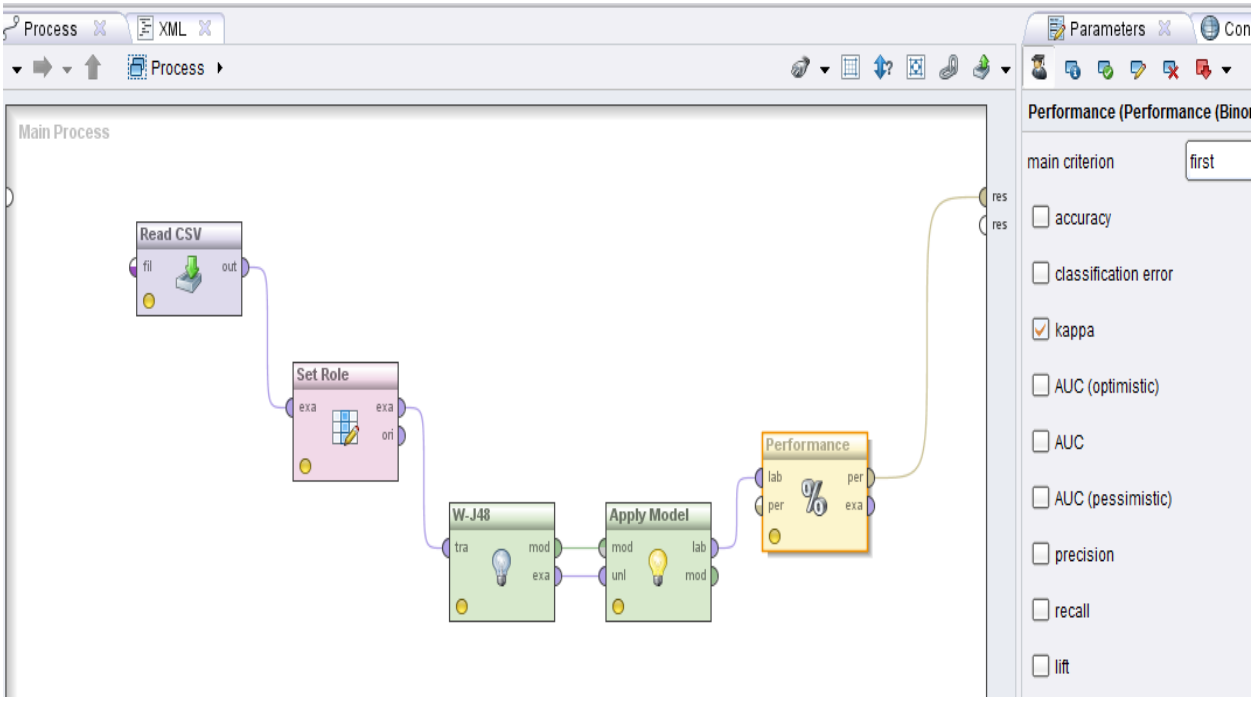
(Note that you have to scroll down to see the case where CVS ct is greater than zero).

16. Note that J48 decision trees are extremely complicated to think through all at once. And they are one of the simpler algorithms to interpret!

17. Click on the little writing icon at the top to go back to the main screen



18. Now add two more operators to the right of W-J48. First, an Apply Model, and second, a Performance (Binomial Classification). Make sure that you link the operators as shown here. You can delete a link by right-clicking on it and selecting delete, or you can click on it and press the delete button. Choose kappa in the window to the right. Then press run.



19. You should see this screen. This shows you the model's Kappa and confusion matrix. The kappa is excellent, in fact too good. Keep in mind we did not use cross-validation, so this model is being trained and tested on the same data set.

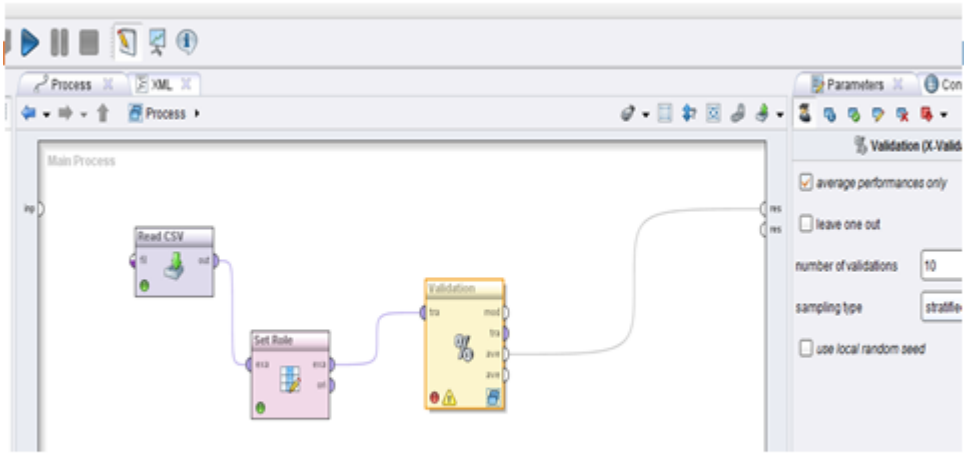
    Here's how to read the confusion matrix. There are 165 cases where the model says "Y" and the data says "Y". There are 383 cases where the model says "N" and the data says "N". There are 11 cases where the model says "N" and the data says "Y". There are 5 cases where the model says "Y" and the data says "N".

kappa: 0.933

	true N	true Y	class precision
pred. N	383	11	97.21%
pred. Y	5	165	97.06%
class recall	98.71%	93.75%	

20. Now go back to the main screen, and create what you see here. You should delete W-J48, Apply Model, and Performance, and add X-Validation. You will get some error messages. Don't worry about those for now. In many cases, you'll want to do Batch X-Validation instead of X-Validation. Batch-X-Validation allows you to do student-level cross-validation, or item-level cross-validation, or population-level cross-validation. Regular X-validation supports flat cross-validation, as talked about it the video lecture.

    Note the options over to the right, which allow you to do k-fold cross-validation (by default, set up to do 10-fold cross-validation), or to do leave-one-out cross-validation.

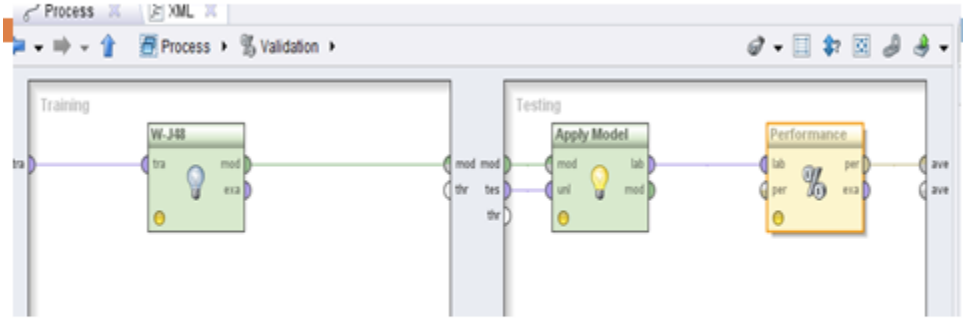


21. Now double click on the validation box (the tall yellow one).

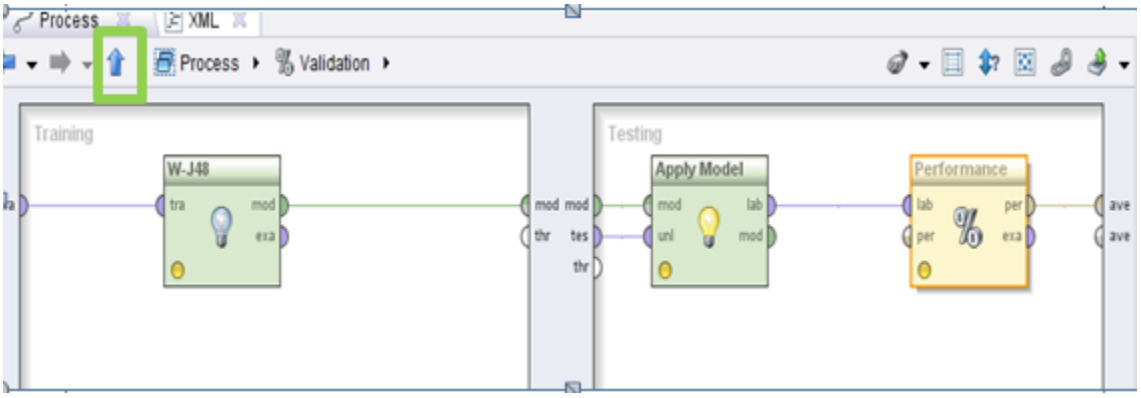
    It will bring you to another screen. Add operators as shown here –
    the same ones you just deleted.

    The left box represents what you do with the training folds – build a model.
    And the right box represents what you do with the test folds – apply the model, and see how well it does.

    Set up everything the same way you did before, e.g. with Performance (Binomial Classification) and the kappa statistic.



22. You can click the up arrow to go back to the main screen



23. Click to run the model. You should get this. Note that kappa is a lot lower once we're cross-validating.

kappa: 0.442 - 0.153 (mikro: 0.445)

	true N	true Y	class precision
red. N	325	70	82.28%
red. Y	63	106	62.72%
class recall	83.76%	60.23%	

24. So now you've built a model and validated it. There's a lot more things you could do.

You could

- Use student-level cross-validation (you would have to add the variable student back in)
- Try different algorithms, such as W-Jrip, W-KStar, KNN, Logistic Regression, Linear Regression (which gives you Step Regression for binomial data)
- Try creating new features (try Generate Attributes) or removing features (try Remove Correlated Attributes)

Have fun!