Assignment 4

Classification: practical issues

In this lab we will use two types data to perform classification tasks. The first dataset has been adapted from data provided by Wisconsin University, where computer-derived nuclear features are used to distinguish malignant from benign breast cytology (that is, breast cancer). Features are computed from digitized images of a fine needle aspirate (FNA) of a breast mass: they describe characteristics of the cell nuclei present in the image. The original images were annotated by an expert with labels M and B. The second dataset will be constructed starting from a set of news documents from the BBC, already used in the previous assignment (acknowledgments in the data folder).

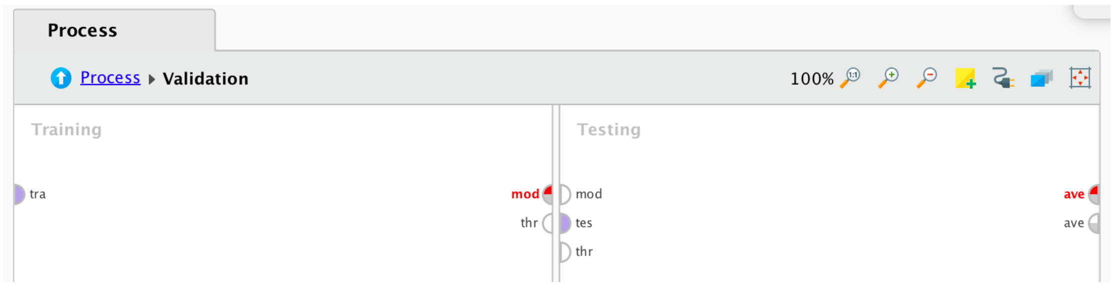
The objective of this lab is to address some typical problems related to the practical usage of classification algorithms. You will start with a **naïve classification** approach, just applying a data mining algorithm on the original data, and see the accuracy you would obtain in this case. To do this you will have to set up a model evaluation process. Then, your objective is to apply the knowledge learned in the first part of the course, and partially applied during the first assignment, to **improve this baseline accuracy** as much as you can. For your information, the scientific literature reports a best accuracy of 97.5% on these data, computed with repeated 10-fold cross-validations. The assignment concludes with a classification of non-tabular data, where you can test your ability to apply general classification concepts to different types of data.

TASK 0: warm up) Read the cancer.csv data into Rapid Miner and have a quick look at them. (We assume that) you are not an oncologist, so you are not required to understand the meaning of all attributes: the objective of this assignment is to use automated methods to identify if some of them are useful to classify the data. Of course if this were not a training exercise you would work in collaboration with domain experts. However, some of the names should be easily understandable. Attributes prefixed by "LARGE" indicate the largest values over all cells in the original image, and attributes prefixed by "SE" indicate the standard deviation of that attribute's values over all cells in the original image.

You can start by configuring the Read CSV operator using the "Import Configuration Wizard", so that you can directly specify the roles of the *special* attributes (id and label) at this stage. You should be familiar with data importing from previous assignments.

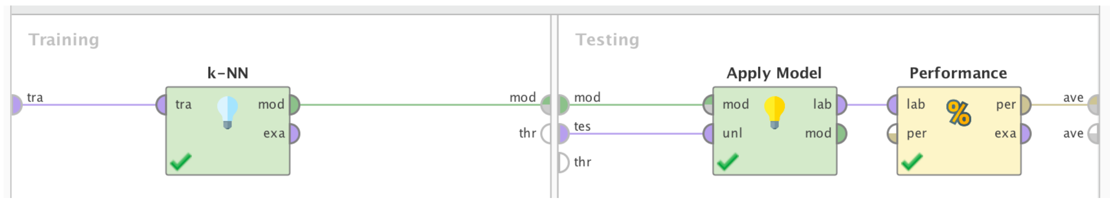
TASK 1: basic k-NN classification) Compute the accuracy you can obtain running a simple k-NN classifier on the original data.

In Rapid Miner, to compute the performance of a classifier we can use a Validation operator, which is a sub-process that can be expanded. Start adding a Split Validation operator, and inspect its internal structure by double-clicking on it.



You will find two phases inside it: training and testing. This operator produces a training (*tra*) and a test (*tes*) dataset from its input. You will also see input ports for training and test datasets (the operator will automatically take care of splitting the original data into these two sets).

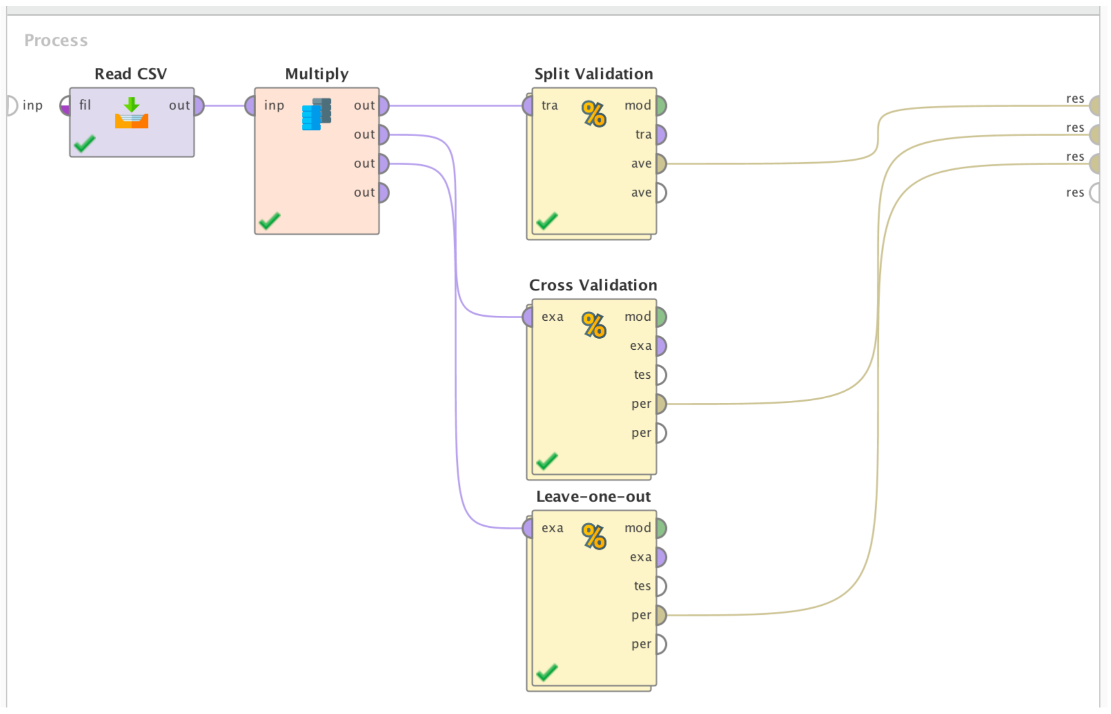
For now use k-NN with k=1 and Euclidean distance as the numerical distance (as you have seen, all regular attributes are numerical) in the training phase. Also use the Apply Model operator in the testing phase. Inside the testing phase use also a Performance operator to evaluate the labels predicted by Apply Model – accuracy is the default performance evaluation metric computed by the Performance operator, but you can also compute others.



The result of the Performance operator can be connected to the *ave* output port. Try to execute it and see if the classification result is satisfactory.

You can now try different ways of producing test data. In particular:

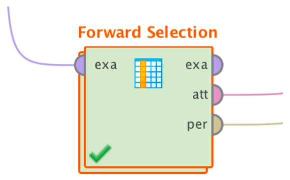
1) Holdout, 70/30 (called Split Validation in Rapid Miner: the one you just used).
2) 10-fold cross validation. Use the sampling method you prefer (e.g., shuffled, building the folds uniformly at random) but please try to relate the method to your knowledge and motivate your choice.
3) Leave-one-out.



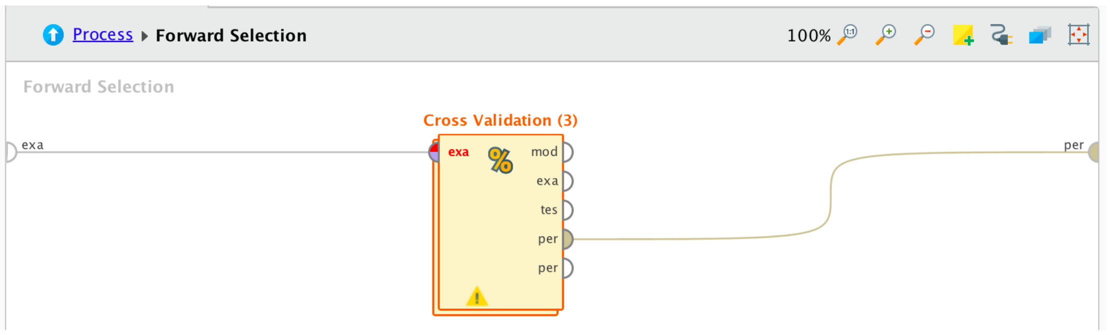
To get an idea of the potential impact of K on these results, choose one of the validation approaches (e.g., cross validation) and set K to 10 in its internal k-NN operator. Check the change in accuracy. (We will get back to how to find a good value of K later.)

TASK 2: data scaling) One potential problem with k-NN is the presence of significantly different scales in the different attributes. You should now know how to rescale the data, so we do not provide additional details here: what we want to focus on is the impact of preprocessing on classification. Check if this improves accuracy, and also have a look at the rescaled data.

TASK 3: Feature selection) Now, try a different approach. The data might contain some attributes that are not useful for the classification, but can make the distance computation less accurate. Instead of rescaling the attributes, try to select a small subset of them. You can do it using the Forward Selection operator. The Forward Selection operator starts with an empty selection of attributes and, in each round, it adds each unused attribute. For each added attribute, the performance is estimated using some validation operator (in this case just use the 10-fold cross-validation, that you can copy and paste into the Forward Selection sub-process). Only the attribute giving the highest increase in performance is added to the selection. Then a new round is started.



As shown in the figure above, send also the attributes chosen by this approach (output port *att*) to the output of the process, so that you can see which ones have been selected. Notice that the Forward Selection operator must contain a Validation operator, the same that you have been using so far; as mentioned above, you can just copy and paste one of the Validation operators you have previously created into Forward Selection – double click on it to open it.



TASK 4: Combining the two methods) Now that you have tried the practical potential impact of these approaches, combine them together to see if their combination can further improve your results: first rescale the attributes, then apply forward selection to the normalized data.

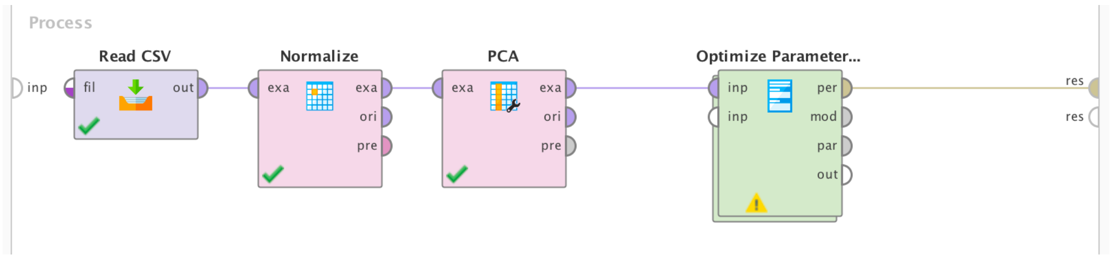
TASK 5: Principal Component Analysis) Now, you can try PCA as an alternative approach to reduce the dimensionality of the data. Also in this case, the objective is not to learn about the operator (that you already know) but to examine its effect on a classification task.

In summary, please replace the Forward selection operator with a PCA transformation.

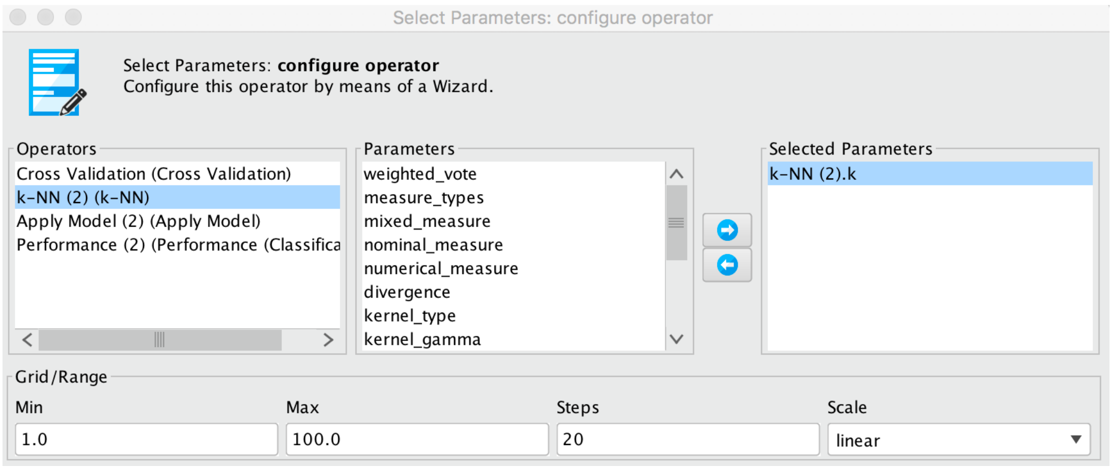
First, have a look at the result of the PCA transformation, without reducing the dimensionality of the data. In particular, check the variance captured by each of the components. Then, use the transformed data for classification. Reduce their dimensionality keeping only some of the components. Try with 1, 5, 10, 15, 20, and check the corresponding accuracy.

Also double-check what happens to the accuracy if you invert Normalize and PCA, that is, you first compute the principal components and then normalize them. Before doing it: what would you expect? Was the effect the expected one?

TASK 6: optimizing the parameters – K) In the previous task, you had to manually check different parameters for the "number of components". This operation can be automated. We will try this here for the K parameter of the k-NN classifier: how can we choose the best K (that is, the one leading to the best accuracy computed on the test data)? You can do this by including the Validation operator inside an Optimize Parameter sub-process. We will use the "Grid" version of this operator, which runs the validation for multiple values of the specified parameters:



In particular, when you set up Optimize Parameter, you can choose among all the parameters contained inside it. Clicking on the operator you will see the tab to set its parameters (as usual), and clicking on Edit Parameter Settings you can select the parameters to optimize and which values to try. Choose K, and specify which values should be tested (in the following example, we test 21 values of K between 1 and 100):


Select Parameters: configure operator

Select Parameters: **configure operator**
Configure this operator by means of a Wizard.

Operators
Cross Validation (Cross Validation)
k-NN (2) (k-NN)
Apply Model (2) (Apply Model)
Performance (2) (Performance (Classifica

Parameters
weighted_vote
measure_types
mixed_measure
nominal_measure
numerical_measure
divergence
kernel_type
kernel_gamma

Selected Parameters
k-NN (2).k

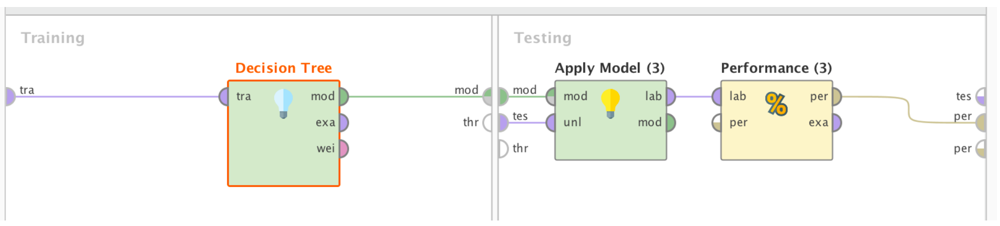
Grid/Range

Min	Max	Steps	Scale
1.0	100.0	20	linear

After executing the process, in the Results panel you can look at the accuracies obtained using different values of K. Use the Charts view to plot them.

For now, we only optimize for K. You are free to do the same for other parameters as well, for example the number of components kept after PCA. This is not required for this assignment though.

TASK 7: Decision tree) The analysis you have performed so far has told you something about the potential issues in the original data: scale and attribute relevance. This suggests that you should also try using an approach that – as you know – is less sensitive to these problems: decision trees. Replace the k-NN operator inside the cross validation. At this point in the course you should be able to understand the meaning of all the parameters of the decision tree: go through them and see if you know what they mean, but you can just leave them as they are for now. (We trust that you would be able to optimize these as well, as done for Task 6).



As decision trees are not affected by scaling problems and can *to some extent* autonomously perform the selection of relevant features, you can now simplify the external process by removing these operations and perform the validation directly on the original data, without standardization nor PCA. Check the impact on accuracy.

Have a look at the generated model (the tree). Try to understand what discriminates between a benign and a malignant case.

As a final change, modify the "minimal leaf size" parameter of the decision tree to 20. This should result in a more compact tree. How many nodes does it contain? How does it relate to the previous tree you obtained?

TASK 8) This task is intended to be performed independently, so we only provide limited instructions. The knowledge acquired during the lectures and from the assignments should be sufficient for you to set up this process from the beginning to the end. To perform this task you need the Text processing extension.

You will use the data *bbc.zip*, containing five folders (business, tech, sport, …), each containing several news articles about the same topic of the folder. The objective of the task is to set up and validate a classifier to identify the topic of the articles based on their text.

We recommend to try different options, so that you can reflect about their impact (or absence of impact) on this classification task.

TASK 9 - optional) In this assignment we have used only two types of classifiers, to test the impact of different choices on the result. You (may) have learned other types of classifiers in other courses, or be interested in trying others. You are of course free and welcome to try any other model available in RapidMiner, either among the operators available in the basic installation or others you may find in the extensions.

CONCLUSION: Reflect on your starting point, that is, the direct application of a classification algorithm (k-NN) and the corresponding accuracy you could obtain, and what you have been able to achieve at the end of the assignment in terms of: 1) potential problems, 2) possible solutions, 3) improvement in accuracy and 4) understanding of the significant information that can be used to discriminate between the two classes. Also recall the optimization methods you have used to examine the role of different attributes and of the parameter K. Feel free to experiment with other parameters. Finally, reflect about the impact of different choices on the classification of text documents, and about the differences between working with text and tabular data.