JOHANNES KEPLER
UNIVERSITY LINZ

PR Computational Data Analytics (351.044) – Part 1 of 3

Summer term 2025 - Prof. Dr. Johannes Fürnkranz, Dr. Van Quoc Phuong Huynh

Deadline:	**25.04.2025, 23:59h**
Presentation Zoom Session:	**28.04.2025, 9:00h**

The aim of this practical course is to gain practical experience in data mining and computational data analytics. It can be taken by students of course *KV Computational Data Analytics (KV 351.008)* to accompany the lecture.

The exercises are split in three parts which will be uploaded when the corresponding lecture notes have also been uploaded. The tasks can be completed alone or in a small group (maximum three students, name and matriculation number must be visible in the submission). The submission should be in the form of a self-explanatory presentation (e.g. PDF, OpenOffice or PowerPoint) with a focus on interpretation or analysis. Results are only to be recorded to the extent that they are necessary to verify the statements. Further files (e.g. in TXT/ARFF/CVS) are not considered. The submissions are made in Moodle (one per group, all files in a zip folder named "group_xx.zip", e.g. group_01.zip) and must be made by 23:59 on the deadline date at the latest. Each group will have a presentation (10-15 minutes including presentation and Q&A) for each part via a Zoom meeting. The Zoom meeting link can be found on the Moodle.

We will start with using the data mining framework `Weka`[1].

1 Rule Learning: Application and Interpretation (10 points)

In this task you will explore and compare the results of different rule classifiers on different datasets in `Weka`. Particularly, you will compare `JRip`, (with and without pruning by setting `usePruning=True/False`) and `ConjunctiveRule` on different datasets. You will find `JRip`, a reimplementation of the most popular rule learner `Ripper`, and `ConjunctiveRule`, a learner which learns only one single rule. The latter must first be installed via the package manager. To do this, open it in the start menu via Tools > Package manager, search for `ensembleLibrary` and install the package. If the classifier `ConjunctiveRule` does not appear as an option when you choose the classifier, you might have to restart `Weka` before using it.

You will choose eight datasets from a package ("datasets-UCI") which can be downloaded from `https://waikato.github.io/weka-wiki/datasets/`. A short description of the datasets and

[1] `https://waikato.github.io/weka-wiki/downloading_weka/`

its attributes are usually included in the beginning of the `arff` file. Datasets should be selected so that you have some variances in the number of instances, the number of attributes and the attribute data types (categorical, numeric, mix) for more meaningful results. You will also additionally include bigger two datasets e.g. "adult" and "connect-4" downloaded from `https://archive.ics.uci.edu/datasets`. The total ten datasets should be shown in the increasing order of number of examples.

You should use the standard evaluation settings (10-fold cross-validation) for each rule classifier performing on each dataset.

1. Compare the accuracies and the number of rules, conditions and predicted classes of the resulting rule sets, and the running time with respect to
   - the datasets
   - the rule classifiers
2. Many rule learning algorithms use a so-called default rule, which is used for classification when no other rule fires. Is there a default rule for all algorithms? If so:
   - Which class is usually chosen as the head of the default rule?
   - How do you interpret the quality of the default rule?
3. On the basis of the previous subtasks, which of the datasets seem the easiest or best to learn?
4. Perform a Friedman-Nemenyi test on the results and check whether there is a significant difference between the performance of the classifiers.

2 Noise and Pruning (10 points)

Choose the dataset with the highest accuracy in the previous task and at least 50 instances. Disturb the class information in this dataset by adding different levels of noise (for example, 5%, 10%, 25%, 50%, 75%, 100%) with the filter `weka.filters.unsupervised.attribute.AddNoise` in "Preprocess" tab. To guarantee that the noised datasets will be prepared correctly, for each noised dataset you should (re)load the original dataset, change the percentage of noised examples then apply the filter setting. Observe the accuracy and size of the learned trees on the original and the noisy datasets for the tree classifier `J48`.

- with default parameters.
- without pruning (`unpruned=True` / `-U`) and minimum one instance per leaf (`minNumObj=1` / `-M 1`).

Experiment a little with the parameters `-C` (`confidenceFactor`) and `-M` for pruned trees and try to find the combination that gives the highest accuracy on the data disturbed with 10% noise.

Note: A x% noise level is created by replacing the example label at x% of all examples with a randomly selected label from one of the other classes. For two-class problems, you will notice that the performance at 100% noise is identical to the performance at 0% noise. In this case, adapt the bounds in an appropriate way (here 50% noise corresponds to random data).

3 Evaluation Methods (15 points)

In this task different evaluation methods using `Weka` are to be applied and their results discussed. Apply the rule classifier `JRip` to five datasets (e.g. a subset from task 1).

1. **Resubstitution Estimate:** Evaluate the learned rules for each dataset on the same data that were used for training by loading it via Set > Open file as the supplied test set.

2. **Hold-Out Sets:** Do a hold-out evaluation. To that end, first divide each dataset into two equal stratified parts. Both sets are used for training and later on also serve as a test set for the other set. A stratified split can be achieved with the filter `weka.filters.supervised.instance.StratifiedRemoveFolds`. Set the parameter `-N` (`numFolds`) to 2 and `-F` (`fold`) to 1, apply the filter and save the first part of the dataset. For the second part, you need to reload the original dataset, set `-F` 2, apply the filter and save the second part.

   Now train the learner on one dataset and determine the accuracy on the other set. Then swap the roles of training and test set and compare the results. Assuming that these test sets are real use cases, how do you assess the estimates of the evaluation methods from the previous two tasks?

3. **Cross-Validation:** Train `JRip` on each of these training sets and evaluate the accuracy of the resulting classifiers (without changing customized options like random seed) using:

   - 5-fold cross-validation
   - 10-fold cross-validation
   - 20-fold cross-validation
   - leave-one-out

   How do you assess the quality of the accuracy estimates obtained? Also compare them to the estimates obtained with hold-out set evaluation and the resubstitution estimate.

4. **Repeated Cross-Validation:** Repeat the previous task with the difference that you should now use a 10x10 cross-validation for evaluation. To do this, apply a 10-fold cross-validation ten times with ten different random seeds and average the achieved accuracies. Repeat the previous task another time using a 5x2 cross-validation, i.e., five repetitions of a 2-fold cross-validation. Compare the accuracy estimates obtained in this way with the estimates from the previous task. Are the results for diffeent random seeds stable?

5. **ROC Curves:** Select a sufficiently large dataset of a binary classification problem and compare the ROC curve and AUC for `J48` and `NaiveBayes`. The ROC curve can be created by right clicking in the result list and selecting Visualize threshold curve.