Final Assigment Data Mining

In the final assignment, you need to apply all that you have learned in order to build the best possible machine learning models for a real-world dataset.

This implies:
* Selecting machine learning algorithms and tuning their hyperparameters.
* Preprocessing the data to select features, normalize your data, etc...
* Handle issues such a unbalanced class distributions
* Evaluate your models using proper evaluation procedures (e.g. cross-validation) and evaluation measures (e.g. AUC, cost functions,...)
* Use evaluation results to make informed decisions about which learning algorithms, hyperparameters, or preprocessing techniques to try

Note that this assignment is partly collaborative. You will be able to check the solutions of other teams on OpenML and learn from them. However, you cannot just copy them (resubmissions will be ignored). OpenML keeps track of *when* you submit your solutions. In order to stand out, you should come up with good solutions (good models) and be the first to submit them. OpenML will keep a timeline of all solutions, clearly showing who is the first to try out a certain technique, and part of your grade will depend on how often your group was taking the lead in developing better and better models (e.g. by submitting good models or answering questions). However, the grade does not only depend on who built the best model, it also greatly depends on the range of techniques you try, which decisions you make, and how you interpret the results. Ultimately, we want to see that you fully understand the techniques that you are using.

You may also like to use OpenML to explore which models yielded good results on other (similar?) datasets and try them out on this dataset. You can also often check how fast models run on datasets of similar size to avoid problems (some models are really slow).

Uploading models

You can use WEKA for all experiments. You can use the WEKA Explorer for initial experimentation, but you need to use the OpenML Experimenter to submit all solutions (as in assignment 4). Use 1 shared OpenML API key per team for uploading, so that all runs that matter appear under the same user on OpenML. Note that, using the Experimenter, you have to wrap multi-step processes in meta-classifiers. For instance, if you want to do feature selection + k-Nearest Neighbors, you'll need to use the AttributeSelectedClassifier to wrap the IBK classifier. Likewise, you'll need to use the MultiSearch meta-classifier to do a grid search over hyperparameters.

Advanced students can also experiment with scikit-learn (Python) or mlr (R). These allow a lot more flexibility, and more complex pipelines. However, if you choose to do so, we expect you to work more independently, as we cannot cover these libraries in the course. You can of course still ask questions on Canvas. Each of these tools have their own OpenML integration. For more information on how to use them, see the OpenML website.

Report

You need to closely document the experiments that you have run and upload all experiments to OpenML. We expect you to report on at least the following:
* Which are the main algorithms that you have tried? How did you select these?

* How did you find the most important hyperparameters and their optimal values? If you run (local) experiments to find optimal values, report these experiments. Include line plots or heatmaps showing the effect of the hyperparameters.
* How did you preprocess the data? Try to be specific and data-driven. For instance, compare multiple methods for feature selection or normalization. Which combinations of preprocessing techniques and learning algorithms seem especially useful?
* How did you take class imbalances into account? What about other data defects (e.g. missing values)?
* Which unexpected challenges did you encounter? How did you (attempt to) solve them?

You will be evaluated on all 5 aspects listed here.

Finally, don't forget to incude your names, and provide an description of how the work was distributed (if at all), and who contributed what to the assignment. Also clearly indicate the name of the person who is uploading the results (whose API key is used), we will use this to check all uploaded experiments.

The report should be no more than 6 pages.

The data and task

The datasets that you need to analyse are:

* Spectrometer (https://www.openml.org/d/313). This is data from an infrared astronomy satellite. It is an unbalanced multi-class classification datasets with a mix of numeric and categorical features. It is rather small but has many features. You'll need to build the best model according to the area under the ROC curve (https://www.openml.org/t/145682). Hence, add task ID 145682 in the OpenML Experimenter to upload your experiments.
* Creditcard (https://www.openml.org/d/1597). This is real credit card data. Because of privacy concerns, the original data was processed with PCA, and the 28 principal components are used as features here. Only the amount of the transaction is kept unaltered. The goal is to detect which transactions are fraudulent. This is a cost-sensitive classification problem, where the cost of a false negative is 500 times that of a false positive (i.e. not detecting a fraudulent transaction is much worse than raising a false alarm). You'll need to use task 145685 (https://www.openml.org/t/145685) for this, and observe the total cost online (you want to minimize this cost). To train models, use the CostSensitiveClassifier in Weka and set the cost matrix to [[0,1],[500,0]]. Note that this is a larger dataset (250,000 instances), so you'll need to be more careful about which classifiers you actually run.

OpenML will automatically evaluate your models when you upload them, and all results (from all students) are gathered on the corresponding task page (links above).

This page has a few interesting tabs:
* Overview: shows a scatterplot of how each algorithm performs (irrespective of who submitted it). Every dot here is a run, and you can click on it to see the details.
* Leaderboard: shows a timeline and a ranking of all the people that have submitted runs to this task. Every dot on the timeline is a run which you can click to reveal the details. Also note that you can select the evaluation measure. OpenML computes a whole range of measures, but here we are interested in AREA UNDER ROC CURVE (for the first task) and TOTAL COST (for the second task). This should be pre-selected.
* All runs: shows an overview of everyone's runs in this task. A run is an execution of an algorithm on a dataset, and hence contains the model. You can click on any run to see the details (who submitted it, which algorithm and parameters were used, the models, and their evaluations). It also shows errors if something went wrong while you built the model.

Grading

The assignment will be evaluated based on your group report as discussed above. In your report, clearly mention your best submission for each of the tasks. We will also use the OpenML timeline and leaderboards to track your activity during the assignment, and reward the teams that are taking the lead. You are free to submit as many models as you want.

Your grade (on 50) will be based on different components:

- 40 points: Written report as detailed above.
- 4 points: Cross-validated performance as evaluated by OpenML, for AUC and total costs. I.e. how good are your models?
- Up to 3 points for excellent performance (top of the leaderboard or statistically equivalent).
    - Winning team wrt AUC (task 1) will get 1 extra point.
    - Winning team wrt Total Cost (task 2) will get 1 extra point.
    - Any team that at some point during the assignment earned 1st place on the leaderbord will get 1 extra point.
- Up to 3 points can be earned for going beyond built-in WEKA techniques. I.e. you can come up with your own approach and improve an existing WEKA algorithm specifically for this challenge, or you can use R or Python to write your own scripts.

Submission deadline

You have to submit only the report, containing information about your experience using canvas.tue.nl and before 21 April 2017, 23:59.

Appendix 1: Uploading runs from WEKA

OpenML is integrated with several machine learning environments, such as WEKA and R, and you can tell these tools to download certain OpenML tasks, so that you don't need to worry about formatting the data or building the correct train-test splits. You 'just' need to select the best algoritms and parameters. When you run experiments, OpenML will also automatically upload and evaluate them, and show them online.
Instructions:
* Go to http://www.openml.org
* Click on 'Sign In' to create an account (you will need this to submit results).
* If you want to use WEKA (recommended for most student in this course), you can find instructions here:
http://www.openml.org/guide#!plugin_weka
* If you want to use R (mlr), you can find a tutorial here:
http://www.openml.org/guide#!r
* If you want to use Python (scikit-learn), you can find a tutorial here:
http://www.openml.org/guide#!python

In all cases, you need to authenticate, and then indicate that you want to work on the right task, e.g. 145682. This makes sure that your experiments are correctly registered.