Exercise 1: Classification

VU Machine Learning
Winter Semester 2022

Goals

In this exercise, you will perform classification using multiple algorithms on a number of different datasets. The goal is to experiment with these different combinations and settings, and go through the whole machine learning process from getting your data, preparing it, making your predictions, to evaluating and comparing the results, discussing them and drawing final conclusions. You will experience how the size of your dataset affects the runtime and how different preprocessing strategies and parameters impact the performance of classifiers. You will learn how to measure and compare these changes, and thus improve the overall performance. You should investigate such changes, provide comparisons of your results and analyse them, to be able to discuss your findings.
You will also participate in our own Kaggle Competition within this course, in which you might earn bonus points for the exercise.

Groups

Groups of exactly 3 students

- It is beneficial to remain in the groups from Exercise 0, but you may switch
- **Work as a team**

Deliverables

Submission package, Name: Group00_NameNameName

- 10-15 page report (including tables & diagrams)
- Code & scripts

Presentation

- Presentation dates will be published in the upcoming weeks
  In total, you will present 2 exercises out of 3; either Exercise 1 OR 2, AND exercise 3.

Tools

Generally, you shall use APIs to enable repeatable, scalable experiments. Tools you can use include:

- Python / scikit
- WEKA (http://www.cs.waikato.ac.nz/ml/weka/) - use the API, not GUI
- R (http://www.r-project.org/) – advanced & powerful software – if you know R already, or if you really want to learn it
- MATLAB - same as with R, use it if you know it

Exercise

Datasets:

You will use 4 datasets:

- 2 from our own Kaggle competition
- 2 from Exercise 0 (unless you encounter issues)

IMPORTANT: You must choose diverse datasets, with different properties

- number of samples – small vs. large
- number of dimensions – low vs. high dimensional
- number of classes – few vs. many classes
- pre-processing needed…

Classifiers:

You will choose 3 different classifiers from different types of learning algorithms

- So in the end you will have 4 × 3 combinations of datasets × classifiers
- You can use any classifier - ones already covered in the lecture, ones that will be covered in the lecture, and also those that won't be covered in the lecture
- But make sure that you choose from at least two different "types" / "paradigms" - i.e. do not choose 3 tree-based classifiers, or 3 NN based classifiers, or 3 ensembles, ….

Task:

Perform necessary steps as presented in the lecture:
Importing the data
Data Exploration and Preprocessing (missing values, outliers, scaling, encoding, etc.)
Carry out the **classification**:

- Run classifiers, and **Experiment** with:
    - Different classifiers and your datasets
    - Different parameter settings (= several results per classifier per dataset, not only random/best)

Evaluate and analyse the **performance** (primarily effectiveness, but also provide basic details on efficiency):

- Choose **suitable, multiple performance measures**
- Make **valid** comparisons (among the classifiers, across your datasets, parameters, preprocessing effects…)
- (How) can you improve the results?
- Can you identify any patterns/trends?
    - Which methods work well and which did not, is there e.g. one method outperforming the others on all datasets?
    - How do the results change when preprocessing strategies change? How sensitive is an algorithm to parameter settings?
    - Are there differences across the datasets? Design your experiments so that you can investigate the influence of single parameters.

Compare **holdout** to **cross-validation**

- Pay attention to your splits and settings
  Are there differences? Why? In which metrics? What could have caused it?
- Compare/document changes in **runtime** behaviour with the changing e.g. dataset size

Summarise your results in **tables, figures**!
Document your **findings, issues** in your report

Upload your best results to **Kaggle competition** (more information below)

You do not need to implement the algorithms, rely on libraries/modules
- Code just for loading data, pre-processing, running configurations, processing/aggregating results, ...

Grading scheme:
20% datasets & classifiers description/choice reasoning, preprocessing 30% classification 30% analysis of results, summary, interesting findings 10% presentation 10% submission package & report(formal requirements, clarity, structure)

Keep in mind that the grading categories are dependent on each other (e.g. if you do not use preprocessing when needed, your classification and overall analysis will suffer)
Your methodology and reasoning are more important for grading than just achieving the highest e.g. accuracy when performing classification

Pointers for your project

Apply the knowledge from the lectures
Document the whole process
Carefully design your experiments
- work out your **experiment design together as a group**

Important points:
- Explain your choice of **datasets**, introduce them, their characteristics
- Briefly describe the **preprocessing** steps and argue why you chose them
  - Evaluate their **impact** on the results (mainly scaling)
- Explain your choice of **classifiers**, describe their characteristics
  - there is no need to give lengthy explanation about how a classifier works (do not repeat what you heard in the lecture)
- Argue on your choice of **performance measures**
  - Think and find multiple, suitable measures, argue why you chose them (why are they necessary, what do they measure/tell us about the performance), and if they are sufficient
- In the report, include a paragraph briefly describing the steps you took to **ensure** that the **performance of the classifiers can be compared** (think if the comparison makes sense & research what needs to be fulfilled in order to e.g. compare the performance of multiple classifiers on one dataset, how to compare the impact of parameter changes etc.)
- **Discuss** your experimental results, compare them using **tables** and **figures**
- Provide an **aggregated** comparison of your results as well - i.e. a big table of the best settings and results for all combinations (and a summary/findings/conclusions!)

- The idea is to extract knowledge from your results, not just list everything without explanations

Hand-in discussion

Each group will meet with a tutor to discuss their solution. Dates will be published in TUWEL.

- 30 minute slots
- ~5 minutes: present your overall workflow and experiments, main findings. You can prepare, but don't need slides for that - you can do that along your report. Describe:
  - Your datasets & choice of classifiers
  - Your experiment setup (e.g. pre-processing, which parameters you tried, ...)
  - Organisation of work in your group
  - Main conclusions
    - Interesting findings, issues, conclusions
    - Usefulness of the algorithms for your datasets
    - Comparison of classifiers
  - Keep the overview of datasets, algorithms brief, do not repeat materials from the lectures
- Be prepared to show your code for specific parts of your analysis
- Every group member needs to know all aspects of your solution

Report

10-15 pages long, Name of the report file: Group00_NameNameName
Full report of your work, document the whole process

- Datasets, preprocessing, algorithms, experiments, explanation of your choices, arguments, comparisons, analysis, discussion, tables, figures, conclusion.....
- Structure & Readability
  - Important: structure, organisation, clarity and readability of the report
  - Pay attention to clarity and readability of the report, and your tables and visualisations (scale, legend, axis labels, etc.)
  - You are free to choose your own structure/outline, but make sure basic formal requirements are fulfilled (length, title, sections, page numbering, your names, etc.)
- Do not include code in the report (only in the your submission package)

Competition

We will use Kaggle In-class (https://inclass.kaggle.com) for a competition
Submission requires a simple CSV file

- for each sample in the test set: <id>,<predicted class>

Pick two of the datasets provided in TUWEL
Number of uploads to Kaggle per day is limited - start early!

- That way you also have early feedback on your results compared to other groups
  First try locally what works, only then upload to Kaggle
- 10% Bonus points for the top 3 teams!
  - +5% Bonus if you also have your notebooks running in Kaggle
  - Mind that the Bonus points will be awarded to the top 3 on the **private** leaderboard; ties are broken by Kaggle on the basis of the submission timestamp: earlier submissions will be ranked higher than equally well-performing later submissions.
  - The maximum of bonus points you can reach is 15%

    - Bonus points are % of your grade, hence you can get 15 bonus points only if your project grade is 100%
- Group name: Group00_NameNameName
- Your notebook's name: Group00_NameNameName_dataset[_optionalSuffixes] (no points otherwise!)
  See https://www.kaggle.com/notebooks
  Keep it private, and share it with the kaggle users "rmayer" & "AndreaSiposova"