Thema 9: Introduction to Data Mining

Introduction

In this project you will get to know several aspects of Data Mining (DM) / Machine Learning (ML). These two terms are not quite the same but are often used as if they are. By the end of the course, you will know what the difference is.

You will work individually on a ML problem of your own choice, but one or more other students will analyze the same dataset. This will make troubleshooting easier and opens the possibility of peer review (analysis and code), which is also an important skill to master: providing help and critical feedback to colleagues.

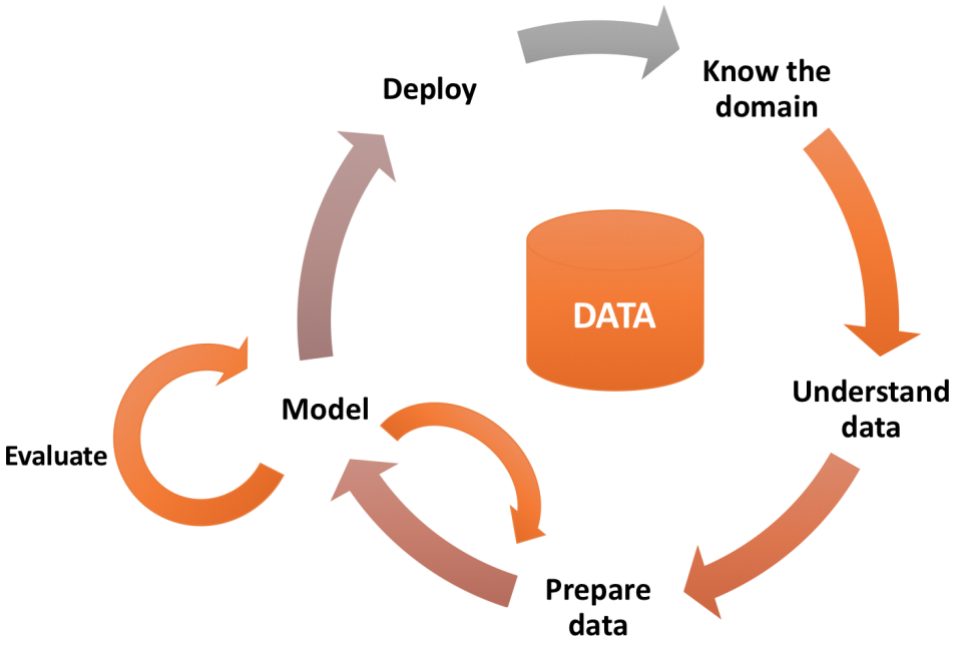
Contents, Learning outcomes & Assessment criteria

See Course Description ("Vakomschrijving") here.

Course program

For exact deadlines, submission details and peer review rules, refer to the Blackboard course. Make sure you have recent versions of R, RStudio and Weka installed (also on your home computer).

In this project, we will explore the entire data mining cycle:



****All** activities up to week 5 should be logged using an RMarkdown document with reproducibility and controllability as focus points.** This document will be reviewed by your teacher and (partly) by your fellow students (peer review) and should be maintained, together with all your data and supporting analysis code, in a Bitbucket/Github repository.

****All** source code files should have a license note.** In Java, Python and plain R source files as a header at the top and with RMarkdown as a chunk below the header. It should look like this (comment chars vary depending on the programming language):

```
/*
 * Copyright (c) 2018 <Your Name>.
 * Licensed under GPLv3. See gpl.md
 */
```

In other languages you use other code comment styles of course. Always put a copy of the license in the root of your code folder. You can find it here: http://www.gnu.org/licenses/gpl.md. IntelliJ and PyCharm have tutorials on how to do this automatically: https://www.jetbrains.com/help/idea/copyright.html. We will discuss the implications of this in class.

Activities marked in bold and with the "package" icon represent deliverables that will be assessed by the teacher or fellow students. It looks like this (and this is a real deliverable, not just an example):

Deliverable: at the end of the course, you will need to submit a log of your complete analysis in the form of a well-maintained RMarkdown document, as well as the pdf generated from it. The log will include the EDA (exploratory analysis) as well as the machine learning phase of the project.

You will only receive a grade at the end of the course if your portfolio of submitted work is complete.

Exact deadlines of the weekly assignments will be published on MS Teams.

Weekly tasks and activities

Week 1

Introduction

Intro on Machine Learning, exploratory data analysis and reproducible research (presentations).

Start log and repo

Start your log (an RMarkdown document) that you will use during all phases of this project: the EDA (see below) and the machine learning activities. Put the log in a folder that will also contain the data files and other resources. Initialize a git repo in this folder. Add a .gitignore file to the repo for files that should not be synced to the remote (e.g. .Rhistory). Also add a decent Readme.md to the root of the folder. After this, commit the first version.
Create a "remote": a public repo on Github, and push your first version to your repo. Note, when you have proprietary data, you should discuss how to do this step with the teacher. Make sure you commit (with a clear commit message) your work after every session you spend on the project.

Choose research topic and dataset

You can either choose from this listing, or find a dataset on UCI or Kaggle:

ILST Research projects

1. Effects of alcohol on lipid bilayer composition. Dataset provided by Tsjerk Wassenaar, RuG.
2. Diffuse cutaneous systemic sclerosis (dcSSc), Limited cutaneous systemic sclerosis (lcSSc). Prediction of diagnosis and/or therapy. Dataset provided by Wynand Alkema, on Blackboard

Kaggle https://www.kaggle.com/datasets

3. Urinary biomarkers for pancreatic cancer https://www.kaggle.com/johnjdavisiv/urinary-biomarkers-for-pancreatic-cancer
4. Birds' Songs Numeric Dataset https://www.kaggle.com/fleanend/birds-songs-numeric-dataset
5. CpG Values of Smoking and Non Smoking Patients https://www.kaggle.com/thomaskonstantin/cpg-values-of-smoking-and-non-smoking-patients
6. Crab body metrics https://www.kaggle.com/inputblackboxoutput/crab-body-metrics
7. Splice-junction Gene Sequences Dataset https://www.kaggle.com/muhammetvarl/splicejunction-gene-sequences-dataset

8. COVID-19/SARS B-cell Epitope Prediction - A simple dataset for epitope prediction used in vaccine development. https://www.kaggle.com/futurecorporation/epitope-prediction
9. Breast cancer proteomes - Dividing breast cancer patients into separate sub-classes https://www.kaggle.com/piotrgrabo/breastcancerproteomes?select=clinical_data_breast_cancer.csv

UCI website (https://archive.ics.uci.edu/ml/index.php)

10. Myocardial infarction complications https://archive.ics.uci.edu/ml/datasets/Myocardial+infarction+complications
11. Breast cancer diagnosis: benign or malignant?: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29
12. Epileptic seizure recognition (challenge set): https://archive.ics.uci.edu/ml/datasets/Epileptic+Seizure+Recognition
13. Yeast protein localization: https://archive.ics.uci.edu/ml/datasets/Yeast
14. Thyroid disease prediction: https://archive.ics.uci.edu/ml/datasets/Thyroid+Disease
15. Predicting hospital readmission of diabetes patients: https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008

Other sources

16. Substance abuse and mental health data (https://datafiles.samhsa.gov/study-dataset/teds-d-2017-ds0001-teds-d-2017-ds0001-nid18480)

If you want to investigate some other dataset of your own choosing. Here are some criteria for it:

- It is a supervised learning problem (there are **class labels** for a training set)
- It already consists of textual/tabular data, or it is obvious how to obtain these by transformations.
- It is a dataset from the life sciences in broader sense: biology, chemistry, medicine, and "quantified self" datasets are eligible.
- It has at least 7 attributes (but preferably more) and at least several hundred examples (but preferably more).
- Preferably, there is at least one publication on the data or the corresponding research.

Your chosen project will need to be approved by the teacher.

Assignment Reproducible Research

Reproducibility is guaranteed by the following aspects of an analysis:

1. There is a log of all steps of the analysis
2. The log is in English and is easy to read (spelling/grammar/formulations)
3. The data are described in detail. It is known
    - where they came from and how they were collected
    - what the variables are and what abbreviations mean
    - what the data types of the variables are, what values they may have, and in what units they were measured
    - what the dependent / class variable is
4. There are no data processing steps missing in the log (or its rendered result!)
5. It is clear what the sequence of processing or analysis steps is (also chronologically), and why they were undertaken in that particular order
6. Every step has
    - an intro: why is it carried out
    - a result: presented in clear tables or figures or other relevant means
    - a conclusion: was the step successful, are the results as expected, what action should additionally be done, what questions do arise as a result etc.

The blackboard site of this course has an item *Assignment Reproducible Research*. Unpack the zip. It is your task to study this EDA and describe weaknesses with respect to reproducibility. Study its contents in pairs or trios and report flaws, errors, weaknesses and possible solutions/repairs at the end of the session.

Document status of knowledge and nature of the data

When you have selected a project to work on, find out where the data was originally published. Read this paper and summarize its findings. This will be the basis of your introduction of the report.

Describe in detail: how many attributes are there, what is the class attribute and what are its possible values, what data type have the attributes and what is their range (possible values) and distribution. In what way were the attributes measured/collected?

Formulate own research question and project goals

Given what you now know about the research you have chosen, formulate a research question you think is interesting and relevant. Here are some guidelines to help you formulate your own:

Questions should in some way. . .

Be worth investigating
Contribute knowledge & value to the field
Be valuable to society

Characteristics of a good research question:

The question is feasible (timeframe, technical and financial constraints).
The question is clear and precise (not too broad or narrow).
The question is measurable (answerable) – can it be supported or contradicted?

Also see this reference for writing research questions
https://cirt.gcu.edu/research/developmentresources/tutorials/question.

For instance, if you were to investigate the Wine Reviews dataset (see https://www.kaggle.com/zynicide/wine-reviews), a question could be "Is it possible use machine learning to predict wine quality based on word contents of the wine description field?".

Your research question will be peer reviewed by other students
Submit your own research question in Blackboard as plain, pasted text and provide feedback to the question of two others.
When reviewing other's research questions, consider the abovementioned characteristics. Use 150-200 words to give your feedback. Give positive and constructive feedback.

Carry out an exploratory data analysis (start)

Using R, perform a so-called Exploratory Data Analysis (EDA). Use good-looking and well-annotated tables and figures to present your results in your RMarkdown log file and discuss your findings critically. For figures, you should only use the ggplot2 package or its extensions, not the base plotting functionality, and for tables use Kable(extra), Xtable, Huxtable, Pander or any other suitable package. Do NOT show console output when a readable table is possible.

The general process of EDA is a cyclic iterative process revolving around these aspects:

1. ***Generate Questions about your data***
2. ***Search for answers by visualizing, transforming, and modeling your data***
3. ***Use what you learn to refine your questions and/or generate new questions***

This process always contains these elements:

- Are there **missing data**? If so, are there many, what is their relevance/impact and how do you propose to do deal with these instances and these missing values?
  Typical visualizations: Table
- What is the **variation** within your data; what is the underlying **distribution**? Split on the class label to see whether an attribute will likely be valuable (informative) in the modeling process. Is a (log) transformation appropriate? Are there outliers? If so, what do you propose to do with them?
  Typical visualizations, where distinction of different classes may be applied: Histogram, Density plot, Boxplot
- What is the **class distribution** - are the classes evenly or unevenly represented? (Only in ML analyses)
  Typical visualizations: Bar chart, Pie chart, Table

- Are there correlated -i.e. dependent- attributes? Describe these correlations, also in relation to the class attribute. This is an especially important aspect, since many machine learning algorithms assume that all attributes are independent.
  Typical visualizations, where distinction of different classes may be applied: Scatter plot (pairs plot), Heatmap of correlation matrix
- Can data be clustered? Perform clustering (Hierarchical, kMeans) and Principal Components Analysis. Visualize these to discover apparent patterns that can help you understand your data and the modelling process that will follow.
  Typical visualizations: Tree, Scatter plot, PCA plot

Week 2

Finish EDA

Finish the EDA you started last week.

In Blackboard, submit your log presenting the EDA, knitted to pdf. This log will be peer-reviewed by another student. The primary focus of this review is on readability and reproducibility.

Week 3

Create a clean dataset

Given your findings of last week, modify your dataset so that it is "clean" and ready for machine learning experiments. This may involve

- Transformations. If variables range across orders of magnitude, it is often a good idea to log(2) transform them.
- Removing instances. Obviously erroneously recorded instances.
- Removing variables. Obviously non-informative attributes should be removed (e.g. identifiers, attributes with little or no variation). Also, with highly correlated attributes, one should usually be removed, or they should be combined into a single value (e.g. 'length' and 'width' can be combined into a 'surface' attribute).
- Recoding attributes. For instance, the `age of first symptoms of dementia` numeric attribute could be recoded into a factor with levels 'pre-adult', 'young adult', 'adult', 'senior', 'ancient'.

Of course, you log this carefully with any arguments you think of for doing so!

Write Results and Discussion & Conclusions on the EDA

Write the Results and Discussion & Conclusion sections of a scientific paper to publish your results. This should span the data exploration and cleaning phase. Liven up your results section using figures and tables. A results section is always a combination of text and accompanying figures! A well-annotated figure has clear axis labels, with quantity and units. For different data series, a legend is added (and the different labels and lines are readily discernable). It also has a caption with title and description. In short: a well-annotated figure tells a story in itself. Here is an example, generated with the given code:

```
ggplot(data = iris,
      mapping = aes(
            x = Sepal.Length,
            y = Sepal.Width,
            color = Species)) +
   geom_jitter(alpha = 0.5) +
   geom_smooth(method = "lm", se = FALSE) +
   labs(
      x = "Sepal length (mm)",
      y = "Sepal width (mm)",
      colour = "Iris species")
```

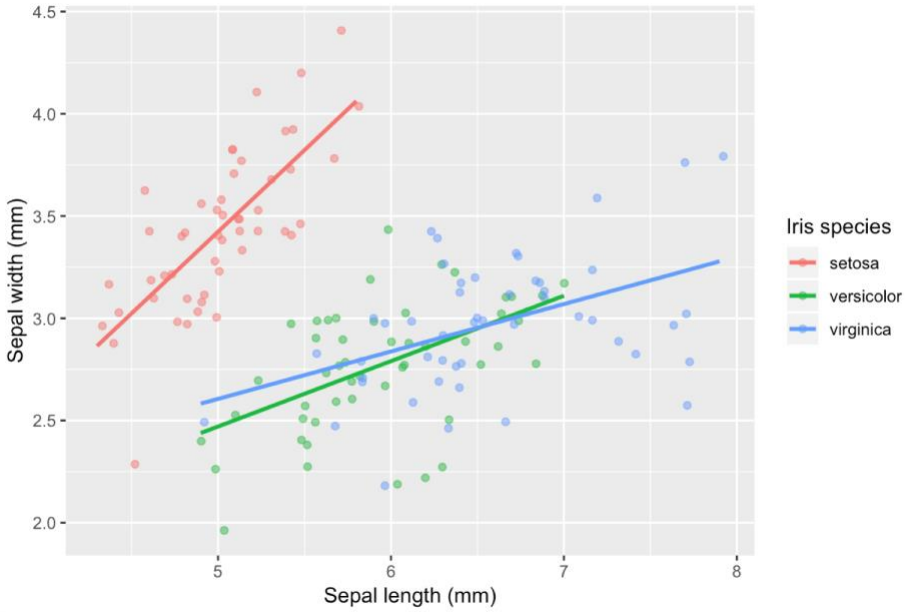


Figure 1. *Relationship between length and width of the sepal in three different Iris species. A linear regression line is added for each species.*

Write a discussion section. Describe your general view of the quality of the dataset - is it good for partitioning the data in classes, does it seem corrupted, are sufficient examples recorded, are attributes independent…(etc)?
Conclude whether you think you have obtained the best possible 'clean' version of the dataset and whether it is a now suitable for the ML process. Also discuss how you think the dataset could have been ameliorated, for instance by obtaining more examples, other variables, or better measurements.

Of course, if your research log was well maintained, writing these two chapters won't be that much of a challenge!

In Blackboard, submit the Results and Discussion & Conclusion paper for review by the teacher. It should contain about 3-4 pages of text (excluding figure and table space).

Week 4

Determine quality metrics relevant for your research

Read this Wikipedia page: https://en.wikipedia.org/wiki/Sensitivity_and_specificity (and/or the corresponding remarks in the book: page 180-181). When evaluating ML algorithm performance, accuracy is the default quality metric. However, for your particular application, other metrics may be more accurate or relevant. Reflect on these and describe which are most important for your project, why this is so, and how you can measure them.

Related but different are the performance aspects of

- Speed and Scalability; is it important that classification is fast or not? For instance, Facebook uses models to filter out nude photographs and they should obviously be very fast and running on cloud-like systems
- Feasibility of online classifications (as opposed to batch processing).

Describe and define criteria that are important for your research. Also describe how you are going to determine and evaluate these aspects.

Investigate performance of ML algorithms

Taking your clean dataset, start investigating the performance of all standard ML algorithms. You should always include ZeroR and OneR to measure baseline performance. Besides these, you should include at the least include representatives of all classifier categories: Naïve Bayes, Simple Logistic, SVM (SMO), Nearest Neighbor (IBk), Decision Trees (J48/C4.5) and Random Forest.
Carry out classifications using 10-fold cross validation, and record relevant quality metrics: speed, accuracy, TP, FP, TN, FN (the confusion matrix) and of course the quality metrics you have chosen yourself. It is quite easy to do this using the Weka Experimenter.
Do not forget to use the cost-sensitive classifier!
Store these data in a table that you can read into R.

Record and discuss your findings in your log and argue which line of research -i.e. which algorithm(s)- will probably be most effective to investigate further.

There is an extensive collection of YouTube videos demonstrating the how-to of Weka; see https://www.youtube.com/channel/UCXYXSGq6Oz21b43hpW2DCvw and -more specifically – the More data Mining with Weka course:
https://www.youtube.com/playlist?list=PLm4W7_iX_v4OMSgc8xowC2h70s-unJKCp

Week 5

Explore Meta-learners and optimize a selection of algorithms and

Using the Weka experimenter, investigate the effect of different algorithm settings with the goal of improving algorithm performance, on at least 2 machine learning algorithms. Also investigate the effect of Attribute Selection methods (see paragraph 8.1 Attribute Selection of the Data Mining book). This video introduces the experimenter:
https://www.youtube.com/watch?v=h3eQpC8n8yA
Apply appropriate statistical tests (the Experimenter supports this). Again, take into account the quality metrics you specified. Record, present and discuss your findings. Also investigate some Meta learners (Stacking, Bagging, Boosting). Have a look at the tutorial "More Data Mining with Weka (5.4: Meta-learners for performance optimization)":
https://www.youtube.com/watch?v=dfUZdxXl_kU&index=30&list=PLm4W7_iX_v4OMSgc8xowC2h70s-unJKCp&t=0s

ROC and learning curve analysis

Create a ROC curve visualization of one or two final algorithms with optimal settings. Is the result satisfying? Take into account the quality metrics you have defined in an earlier stage. Create a learning curve as well. How much data do you need to get a reasonable performance estimate?

Create Java wrapper program for your learned model

OK, so you have a nice model - so what? You have to publish it in a user-friendly way so that others can use your model to predict the class of new, unknown instances. That is why you now have to create a command-line Java application that wraps your final optimized model. This will make it possible to quickly classify new instances. The program should be able to classify new instances having the required attributes – not necessarily all attributes that were present in the original dataset! The program should be able to classify single instances fed from the command line, or batches fed through an input file. If your data was transformed/processed in any way prior to building the model, you should of course do the same with new unseen data before it is fed to the model for classification!
Maybe your model should even be able to perform online (streaming) classification? As a bonus, your program could be extended to give some classification statistics when working batch-wise, or to output probabilities instead (or as well as) class labels.

Use the Weka API. This repo contains some demo code on how to get going with building your program around a serialized Weka classifier:
https://bitbucket.org/minoba/wekaapidemo/overview. Use the Apache CLI API for processing command-line arguments (see http://commons.apache.org/proper/commons-cli/ and a demo program at https://bitbucket.org/minoba/clidemo. This is also an example on how to start and build a Gradle-managed Java project in IntelliJ.

Put this project in a separate Bitbucket repo.
Take special care with the Readme; it is the first item interested parties see when visiting your repo. Do not make it a scare-off!

Week 6

Continue Java wrapper program

Week 7

Finish wrapper program

Finish your command-line application and any loose ends that may be present. Make sure your program is on Bitbucket and has a good readme.md describing your algorithm, what it can be used for, and how to install (set up) and run your program. Do NOT forget to describe all possible command-line arguments and their possible values (and what they default to), and of course the dataformat your program should be fed with. I encourage you to do a Google search and have a look at some good Readme.md examples.

Pitch your results

Present a 90-second summary of your work that will convince others to use your program or visit your oral presentation in a (fictitious) conference.
You have to convince the audience your work is really interesting and valuable for humanity as a whole. You will have to do this in the classroom, without presentation materials (e.g. Powerpoint) or other digital support.

Write report

You have been given feedback on the Results and Discussion sections of your EDA scientific paper. Use this feedback to improve it, and to write an A-grade Results section on the entire project (EDA, machine learning, deployment).
Also add a description of a possible minor project to be carried out in one of the bioinformatics minors (Application Design or High performance / High throughput Biocomputing. This should include a goal, purpose and a deliverable. Can you maybe even think of (commercial) partners that may be interested in this classifier?

The final paper should have these contents (pages are an approximation):
Introduction (½ page) with your research goal (improved with feedback from peer review)
Materials & Methods, containing links to both your repositories (1 page)
Results (5 pages maximum text content)
Discussion & Conclusions (1 page)
Project proposal for minor (½ page)
References

Project proposal for minor

The project proposal for the minor should describe what project would be suitable to enter as candidate into one of the minors of bioinformatics: Application Design (AD) or High-Throughput High Performance Biocomputing (HTHPB). Select one of these minors for which you will write a proposal. The proposal should describe in more or less detail

- what the goal is;
- the desired product or outcome;

- implementation details: technology of choice (e.g. Desktop app, web app, Android, etc.);
- the target audience for which the application should be developed;
- inputs, outputs and visualizations and such.

This is the final portfolio that should be submitted (via email):

- **Your final report**, written in RMarkdown, knitted as pdf document. In the Materials and Methods section links to both repos (see below).
- **Repo 1: Analysis repo**. Contains your log (with EDA and Machine Learning phases), in Rmd and pdf format (html is not accepted!), a Readme.md describing the repo and its contents, and data that you have used. Make sure it has a well-organised structure and does not contain files or folders that should not be there (e.g. .RData files).
- **Repo 2: The Java wrapper repo**. Contains your Java project and a Readme.md describing the repo and its contents, and data or model classes that you have used. Make sure it has a well-organised structure and does not contain files or folders that should not be there (e.g. .gradle folder). Make sure all public Java methods have well-written Javadoc!

Only portfolios that are complete and organized according to these instructions will be graded.