Introduction to Data Mining: Project

See Toledo:

G0Y13a Lecture

G0Y14a Exercises

G0Y15a Project

Questions? Send an email to the TA at lorenzo.cascioli@kuleuven.be.

The project of Introduction to Data Mining consists of **two** parts. Both parts should be handed in on Toledo. You are free to hand in at any point during the semester. The hard deadline is **Friday May** 23^{th} **at 17:00 CET**. No late submission will be accepted. For any questions, please contact the TA.

Collaboration policy:
The project is individual. **No working together!** You must come up with how to solve the problems independently. Do not discuss specifics of how you structure your solution, etc. You cannot share solution ideas, pseudocode, code, reports, etc.
Beyond what is provided, you may not use any other code that is available online. The use of language models such as ChatGPT or any other AI-based systems for completing this programming assignment is strictly prohibited. If you are unsure about the policy, ask the professor in charge or the TAs.

Part 1: Data Exploration (5 points)

You will be exploring a dataset of sports activities performed by different users of a sports tracker during January and February 2022. Each row in the dataset corresponds to one activity. The columns are the following variables:

- Date
- Activity (run or ride)
- Duration
- Distance

- Type (workout or race)
- User ID
- Age
- Gender
- Weight
- Location of the user

Your task is to load the data and explore it to find interesting observations. These findings can be either anomalies in the data that should be taken into consideration, or actual patterns and relationships between the variables. **At least one** of your findings should involve the interplay among two or more variables. There are errors in the data and you must identify and propose a correction to **at least one** of these (hint: you may need to use or lookup "common" or "domain" knowledge). Make sure that your findings are non-redundant, i.e., each finding should not already be covered by another one. You are free to perform any analysis or data manipulation you want for this task.

Report

The report of this part should contain **no more than 4 pages** of text, including figures, tables etc. You should mention **at least five** different interesting things that you found in the data. You should state what the finding is, and provide some evidence. Examples of evidence would be a graph or some sample data records. The report can briefly mention any challenges and/or problems you have encountered.

Grading

This part is worth five points and will be graded on the accuracy, quality (e.g., interestingness) and diversity of the findings, and the quality of the proof you present. Examples of **uninteresting** findings would be the number of variables and the number of examples in the dataset.

What to turn in

- Your report as a PDF file.
- You do *not* need to hand in code for this part.

Part 2: Feature Construction, Classification and Evaluation (15 points)

You will be working on a data set of web pages from computer science departments of various universities. The 3710 pages correspond to the personal web pages of either students (1642), faculty members (1125) or courses (931) within these departments.

The files are organized into a training and a test set. Within the training set, you'll find one directory for each class. These directories in turn contain the web pages. The file name of each page corresponds to its URL, where '/' was replaced with '^'. Note that the pages start with a MIME-header. Some of the pages may not contain useful information (e.g. redirection pages).

Your task is to build a model that is able to correctly classify web pages from the test set as belonging to one of the possible classes. To that end, the **three key steps** are:

1. Converting the raw data into features.
2. Training models based on the features.
3. Evaluating the performance of the models.

Feature engineering

Your first task will be to think about interesting features that might capture key aspects of the problem. Then you have to use your scripting knowledge to actually compute the features to build a feature-vector that can be used to train and evaluate your classifiers. The **minimal requirement** is to go beyond what has been seen in the exercise sessions and to use **at least three** different types of features (the more the better). An example of a type of feature is using bag of words.

Models

Your second task is to train models and make predictions for the web pages of the test set. You should experiment with **at least three** different classifiers, corresponding to three different classes of models. Examples of classes of models are linear classifiers, probabilistic classifiers, ensembles, etc. Please take care to ensure that you use the proper experimental methodology (e.g., no training on the test data).

Evaluation

Your final task is to evaluate the performance of your models, once again according to a sound methodology. For that, you will use **at least two** performance measures: the classification accuracy and another one of your choice. These two performance measures have to be reported for **all** the models you try, as well as for a **dummy baseline** model that simply predicts the majority class for each test instance. The accuracy of all models (yours and the baseline) should be accompanied by a **95% confidence interval**. Finally, you should perform a **t-test** on the differences of accuracies of your models to identify which model(s) perform significantly better than the others. Concluding that the dummy baseline model is the best performer is **not allowed**.

Report

The report should contain **three sections** and may optionally contain a fourth section. The first section should clearly describe all the features that you have decided to construct and why you have decided that these features may be interesting. The second section should describe the experimental setup, i.e., what algorithms did you select and which parameters did you select. You should very carefully describe your setup, i.e., how is the evaluation carried out. Furthermore, you should pose some experimental questions such as:

- Which classifier produces the most accurate predictions?
- Which features contribute most to the performance of the classifier?

The third section should present your findings and provide some interpretation to your findings. Finally, the report can optionally include a fourth section that briefly mentions any challenges and/or problems you have encountered. Please limit the text to **no more than six pages**, including the optional fourth section and any figures or tables you may wish to include. Note that all the requested parts must be mentioned in the report.

Grading

The assignment will be graded with respect towards accomplishing the tasks, the creativity and insight shown in carrying out the task (e.g., on feature construction), the correctness of the experimental setup, and the quality of the report among others.

What to turn in

- Your report as a PDF file.
- Your code with a `README` file showing how to execute your code.

Here are some requirements for us to be able to test your code:

- Your software should read the training data from a directory with three subdirectories (one for each class), and read the test data from a single directory.
- The path to these directories should be easy to specify (command line ideally, or configuration file or at least some clearly identifiable variables at a the beginning of the code; this should be specified in your `README`).
- At the end of the feature engineering step, your entire data matrix X containing the features you have created should be written to a file called `featurematrix.csv`. The first line of that file should contain the name of the columns of the matrix.