Term project description

Kai Puolamäki & Anton Björklund & Rafael Savvides

2 November 2022

DATA11002 Introduction to Machine Learning (Autumn 2022)

In this term project, you will train a classifier on a data set of atmospheric measurements. To complete the project, you should deliver:

- **Sun 11 December:** Predictions for the test set and your estimate of accuracy (`answers.csv`, see below), and a preliminary version of your project report as a single pdf file.
- **Wed 14 December:** A video presentation about your solution.
- **Sun 23 December:** The final report as a single pdf file.

About the data

Figure 1: *The SMEAR II mast.*

The term project is on a data set about *new particle formation* (NPF). NPF happens on some days when small particles (starting from individual molecules) begin to form larger new particles. The particles then spread out of the forest and affect cloud formation and weather. In urban environments, air pollution is often caused by larger particles forming from anthropogenic sources, such as car exhausts, with dire consequences for human health. An interesting research question is under what conditions NPF happens, which is what your classifier will try to model. The NPF formation process is one of the significant scientific outputs from the University of Helsinki in atmospheric sciences. You can find a detailed explanation of the NPF phenomenon from Kerminen et al. (2018).

The training data file npf_train.csv contains several variables measured on different days (rows of the file) at the Hyytiälä forestry field station (mainly at the SMEAR II mast, shown in the picture). The variables are daily means and standard deviations of various measurements between sunrise and sunset. The variable names typically refer to the height of the measurement device in the mast. For example, T48 refers to the temperature at 4.8 meters above the mast base and T672 at 67.2 meters above the mast base (as you can see, the mast is quite tall). CS is the condensation sink in units of 1/s; for details, see Kulmala et al. (2012). You can find more details about the variables at https://smear.avaa.csc.fi/ or via https://wiki.helsinki.fi/x/XYiKDg.

On each day, an NPF event can occur. The type of NPF event is given in column "class4": nonevent means no NPF event took place, and Ia, Ib, and II are different NPF event types. For the event classification schema and explanation of event types, see Dal Maso et al. (2005). For more information about the classification task, see Hyvönen et al. (2005) or Joutsensaari et al. (2018).

An in-depth understanding of the datasets or articles mentioned above is optional for completing the task (from a machine learning viewpoint, this is a relatively standard classification task!). However, understanding your modelling processes allows you to build better models and sanity-check the results.

The data files are pretty "clean" (e.g., no missing values), so you can hopefully concentrate on the machine learning part instead of fighting with the data.

Your task

You should work in groups of 1-3 students.

Your task is to build and apply a classifier to predict the event types for days listed in the test data file npf_test_hidden.csv. The "primary" task is to make a binary classifier (event vs nonevent) for a new variable "class2", defined as follows: "class2" = nonevent if "class4" is nonevent and "class2" = event if "class4" is one of Ia, Ib, or II. You don't need to, *and usually should not*, code your classifier from scratch! You should use various machine learning libraries as in "real life".

Remember that this is a non-trivial classification task. It is possible to do it in many ways. The most straightforward binary classification task, classifying events vs. nonevents, is possible with reasonable accuracy using any decent machine learning library with little effort. Multi-label classification is more complex but should still be doable. However, you should also do the data exploration, preprocessing, feature selection, model selection, classification accuracy estimation etc., appropriately since you will report and analyse your choices and results in the project deliverable.

The project's purpose is not to (even try to!) replicate any methods in the literature, make a super-complex best-performing classifier that beats everything else or attempt to use other data sources etc., to obtain the best possible classification accuracy. Do not use any method that you do not understand yourself! Accuracy of the predictions on the test data is not a grading criterion by itself, even though a terrible accuracy may indicate something else fishy in your approach (which could affect grading).

The Challenge (DL 11 December)

We are organising a non-serious competition (or "challenge") to make the project more interesting.

The challenge uses the following four performance measures, all computed by comparing your predictions on the test data to the correct labels (which we have, but you don't):

Binary accuracy (class2). The fraction of days classified correctly as event ("Ia", "Ib", or "II") or non-event days. This is the main error measure used in the challenge. Larger accuracy (closer to 1.0) is better.

Accuracy of your estimate of accuracy (class2). Estimating how accurate your classifier will be when used in real life is essential. In your answer file, you should estimate your accuracy in the test set. The one who estimates their accuracy most accurately wins this sub-competition. :) As an example, if you say that your binary classification accuracy on the test set will be 0.7 and it is 0.8 you make an error of $|0.7-0.8| = 0.1$. The smaller the difference the better.

Perplexity (class2). A measure for probabilistic predictions, defined as $P = \exp(-\text{mean}(\ln(p_i)))$, where p_i is the probability given by your method for the day i having the correct class, and the mean is over the test set. For example, if your method says there is an NPF event on the day i with probability 0.1023 and there is an NPF event, then $p_i = 0.1023$. Smaller perplexity (closer to 0.0) is better.

Multi-class accuracy (class4). The fraction of days classified correctly to all four classes in class4 ("Ia", "Ib", "II", and "nonevent"). Larger accuracy (closer to 1.0) is better.

Some practical tips:

- If your binary classifier doesn't easily twist into multi-class problems, a simple solution is to predict the majority event class (Ia, Ib, or II) for days predicted to be event days. This will give you a valid `answers.csv` file, even though you probably won't win the multi-class accuracy competition. :)
- If your classifier doesn't output probabilities for event vs nonevent days, a simple solution is to smartly guess (i) a probability $p_1 \in [0.5, 1.0]$ for days predicted to be event days and (ii) a probability $p_0 \in [0.0, 0.5]$ for days predicted to be nonevent days. All event days would then have the same probability p_1, and all nonevent days would have the same probability p_0. This will give you a valid `answers.csv` file, even though you probably won't win the perplexity competition (but who knows?).
- Notice an equal number of `event` and `nonevent` days ("class2") in the training data. The test data days are sampled randomly from the actual data, in which the number of `event` and `nonevent` days differs (nonevent days are a bit more frequent). This is typical of a real-life scenario: the class distributions on the training data may vary from what is actually observed when the classifier is applied. You may want to consider this slight class imbalance, but ignoring it may not have a substantial effect on the final results.

At the latest by 11 December, you should submit to Moodle:

- your predictions, and
- a preliminary version of your report.

After this deadline, we will publish the prediction scores of every team and the correct labels for the test set.

The preliminary report should describe the work done so far. This report can be a little polished or complete, but it should already contain the basic ideas used in the solution. The teams are allowed to modify their approach and report before they submit their final report by 23 December. However, please do not simply copy the method used by the teams with good performance in the competition!

Video presentation (DL 14 December)

Every group will present their term project solutions in the final lecture. Instead of live presentations, we will use prerecorded videos. Please make a 90-180 second video about your term project and submit it on Moodle at the latest by 14 December!

The video should contain a short pitch about your term project solution in which you briefly tell about:

- Overview of your approach
- Your chosen classification algorithm
- Steps you took to select good features and model parameters
- Summary of your results
- Any insights

You can prepare a couple of slides to support your presentation. Make sure that at the beginning of the video, you include the **full names (first name + surname) of your group members and the name of your group (as in Moodle)**. When making the video, you can assume that the audience is familiar with the topic and setup: the audience is your fellow students who have done the same task.

The final report (DL 23 December)

You should submit the final report as a single pdf file via Moodle by 23 December.

The final report should contain, among other things, the following:

- The names of the group members.
- The stages of your data analysis, including how you looked at the data to understand it (visualisations, unsupervised learning methods, etc.).
- Description of considered machine learning approaches and pros and cons of the chosen approach for this application.
- Steps you took to select good features and model parameters.
- Summary of your results, insights learned, and how the classifier performed.
- As a final section, please include a self-grading report (at most 1 page) that suggests a grade for yourself (integer 0-5) by using the attached grading instructions (see below).

To pass the project, it is enough to use one of the basic algorithms, do the feature and model selection parts as instructed (you should probably use cross-validation!), and prepare a well-written report.

Practical instructions for writing the report:

- Your report should read like a self-contained blog post or scientific article that is understandable and without any task description. You should explain what you have done and why you have done it so that a person familiar with machine learning can understand what you have done and could, in principle, reproduce what you have done based on your report alone. Put some emphasis on presentation and readability (one of the grading criteria): imagine that the report's reader would be your future boss, who appreciates a clear and concise presentation.

- You are not required to hand in any program code. Your report should not look like a code listing! Your report may contain code snippets if you explain what the reader is supposed to conclude from your code. We may look at them, but we won't go fishing for results and missing details from your code. In other words: all relevant parts of your report should be understandable without going through any code. If you need to include more significant chunks of code, please put them in an appendix so we can easily skip them when grading your report.

- Your report may include tables or figures. Always explain in detail what the tables or figures show and what the reader expects to conclude from them. If you have a figure or table, the text should refer to it at least once.

- You can use suitable typesetting software that produces legible pdf output (LaTeX, Word, R Markdown, etc.). There is no strict page limit, so you can use a readable font (e.g., 12 pt serif font), margins, and appropriately sized figures. Note that Jupyter Notebooks often lead to poorly formatted pdf. Out of curiosity, I took a random sample of 16 similar final reports that got total points from other courses I lectured. The task was identical to this one, but without self-grading (which may add max. 1 page). The page counts of these final reports were: 7, 7, 9, 10, 10, 10, 12, 12, 13, 13, 14, 14, 14, 14, 14, and 14. The reports had between 7 and 14 pages, the median being 12.5.

Even though you are allowed to modify your approach and adjust your algorithms for the final report, you are not required to (and probably should not) make significant changes. The idea is to polish the report and complete whatever steps you planned but needed more time in the preliminary report.

The term project (final report, video, and challenge submission) will be graded on an integer scale from 0 to 5 (1-5 = pass). See the grading criteria below and "Practical arrangements" for how this affects the course grade computation.

The final report will be processed by the Ouriginal plagiarism detection system.

Grading of the term project

At the end of the course, you will be asked to give your project deliverables (final report, presentation, and challenge submission) an integer grade on a scale from 0 (fail) to 5 (excellent). You should attach the grading comments as the last section of your final report ("grading section"). The length of the grading section should not exceed 1 page.

All group members will usually receive the same grade for this part of the course. (The group members may receive different grades if there are substantial problems with the contributions of some group members. Please contact the lecturer as soon as possible if there is any problem resolving them!) The course staff will consider this self-review when giving you the grade for the term project.

Grade for the deliverables

Please use the following grading guidelines to grade your group's deliverables (final report, presentation, and challenge submission) with a single integer grade from 0 to 5. **Please state the grade you gave yourself clearly at the beginning of the grading section!** Your deliverables may have shortcomings in one area, which better results in another area can compensate. You should try to balance any weaknesses and strengths and produce one grade that faithfully describes your group's deliverables.

Notice about the challenge submission: the accuracy (and other performance measures) of the predictions on the test data is not a grading criterion by itself, even though low accuracy may indicate that there is something else fishy in your approach which could affect grading.

In addition to the numeric grade, you should explain briefly (max. 1 page) the reasons for your grading by using the grading criteria described below. The grading criteria are like the Data Science Master's Thesis assessment criteria. Please do not just repeat the grading criteria; tell how they apply and are related to your work.

Grade 5 (excellent): The treatment of the topics shows in-depth understanding, the relevant source material is used and cited, and the discussions show maturity. Appropriate machine learning and other methods have been chosen and applied correctly. The methods used have been analysed sufficiently. The reporting is to the point and exact. The conclusions drawn are in-depth and to the end. The discussion of findings shows an aptitude for independent, critical, and innovative research and thinking. The reports and presentations are polished and "camera-ready." The work has been creative and independent and progressed within the given schedule. The deliverables have been done by using the instructions provided.

Grade 3 (good): The treatment of the topic shows an understanding. The subject and literature are mainly analysed critically. The research material and methods (incl. machine learning methods) are suitable for the problem, and their use is well-argued. The findings have been reported in a primarily clear manner. The research questions are answered feasibly. The language is exact, and the terms used have been defined. The presentation is accurate, although the style may vary. The work has primarily proceeded to the planned timetable. The deliverables mostly follow the instructions given.

Grade 1 (passable): The topic and scope have not been motivated clearly, nor have the subject and goals fully understood. The work shows significant shortcomings in domain knowledge, and the cited sources are generally few or sub-quality. The reporting and analysis of the results have substantial weaknesses. Conclusions and discussions do not follow the scientific style. The deliverables are unpolished. The work has not progressed as planned. A substantial portion of the instructions given was not followed. However, the work still satisfies minimal requirements to be accepted.

Grade 0 (fail): The deliverables fail to satisfy the minimal requirements.

Grade for the group as a whole

Please also give your group a single integer grade from 1 to 5 and briefly (typically 1 paragraph of text) explain your grading. You can use the following rubric as a guideline, even though you do not need to grade each criterion separately. This grade does not directly affect the computation of your course grade. If you did the term project alone, you do not need to do this part.

Criteria	Grade: 5	Grade: 3	Grade: 1
Discussions about the content	The group has analytic and critical discussions. The discussion includes insights from the group members' own experiences. There is little irrelevant chatter.	The discussions are mainly about the topic of the project. There are examples from own experiences. Off-topic discussions are limited.	There are some discussions about the topic of the project. Some examples of own experiences are discussed, but they remain separate from the rest of the work. There are many off-topic discussions or discussions about topics of little relevance.
Setting the objectives and working towards the objectives	The group has a common goal which considers the individual objectives of the group members. The group works so that all the objectives are reached, and the objectives are – if necessary – adjusted during the progression of the work.	The group has a common objective that considers individual objectives to some extent. The group works towards objectives in an organised manner, even though all of the objectives may not be reached.	The group does not have a common objective. The group members work separately and do not share their responsibilities equally. A group has members who do not do their fair share of the work.
Participation, taking responsibility, interaction, atmosphere	Everyone participates actively in the discussions and group work. All group members take responsibility for the group work but also give room for the ideas of the others. Responsibilities are distributed fairly. The atmosphere of the group encourages to learn and do the work. Any conflict situations are resolved and learned from.	The group members participate in the meetings actively. The responsibilities and workload have been distributed fairly. The atmosphere is good, and an attempt is made to resolve conflicts.	The group has difficulties agreeing on meeting times, and all group members do not participate in the meetings. The distribution of responsibilities and work is uneven. Some group members do most of the work, while others do almost nothing. The atmosphere in the group does not encourage to learn in the group. The conflicts are not resolved.
Results and added benefits from the group work	The group work substantially contributes to the group members' learning outcomes.	The group advances the quality of the learning of its members to some degree.	The group brings no additional value to the learning of its members.