DATA11002 Introduction to Machine Learning (Autumn 2025)
Term Project: Classifying New Particle Formation

Matias Loukojärvi & Lauri Seppäläinen & Arash Jamshidi & Ananth Mahadevan & Kai Puolamäki

28 October 2025

Invitation to the term project Kaggle challenge (please do not share the link)

- IML25 Kaggle Invitation

The Kaggle challenge page includes technical instructions for submission.

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

- A submission to the Kaggle competition, i.e., the predictions for the test set are to be submitted to the course Kaggle page.
- A preliminary version of your project report as a single PDF file on Moodle.
- Term project presentation—if you are asked to present your work.
- The final report as a single PDF file on Moodle.

The data are provided solely for the coursework. You **may not distribute the data** or use it for other purposes. Please do not publish the code to solve the term project as the idea is that the groups will do the work independently.

Check the deadline for each of these on Moodle.

Figure 1: The SMEAR II mast

About the Data

The term project is on a data set about *new particle formation* (NPF). NPF occurs on some days when small particles (starting with individual molecules) begin to form larger particles. The particles then spread out of the forest, affecting cloud formation and weather. In urban environments, air pollution is often caused by larger particles forming from anthropogenic sources, such as car exhaust, 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 `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 on SMEAR homepage or on SMEAR XWiki. Note that the data may not be published.

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) 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 provided data is relatively clean (e.g., no missing values), so you can focus on the machine learning aspects rather than extensive data cleaning. However, note that in `test.csv`, the `date` column has been set to `None` and the test data has been randomly shuffled. The column is included solely for exploratory data analysis, and your goal is to model the atmospheric conditions influencing NPF independently of time.

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 `test.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 in the real world.

NOTE: 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`—can be achieved 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 term project report.

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 performance score. You should 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 performance may indicate something fishy in your approach (which can affect grading).

The Online Challenge

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

This competition uses the following three metrics for the Kaggle leaderboard:

- **Binary accuracy** ("class2"). The fraction of days classified correctly as event (`Ia`, `Ib`, or `II`) or `nonevent` days. Larger accuracy (closer to 1.0) is 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 $p_i = 0.1023$; or if your method says there is no NPF event on a day j with probability 0.3000, then $p_i = 0.7000$. Smaller perplexity is better. The possible perplexity values are in $[1, \infty)$. Perplexity of 2 corresponds to coin flipping that predicts uniform probability of 0.5 for "class2", reflecting maximum uncertainty in its binary classification of `event` vs. `nonevent` days.
- **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.

These metrics will be combined into one equally weighted aggregate score defined as:

$$\boxed{\text{score} = \frac{1}{3}\left[\text{binary accuracy} + \text{multi-class accuracy} + \max(0, \min(1, 2 - \text{perplexity}))\right] \in [0, 1]}$$

The last term normalizes perplexity such that a score of 1 corresponds to the best possible perplexity, i.e., the model classifies the binary class ("class2") correctly for all days and is fully confident in its predictions (assigning probability 1 to the `event` days and probability 0 to `nonevent` days). Score of 0 corresponds to perplexity of 2.

The metrics are computed by comparing your predictions on the test data to the correct labels (which we have, but you don't).

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 submission, although the multi-class accuracy component of the total score won't obviously be optimal.
- 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 submission, although the perplexity component of the total score won't most likely be optimal.
- Notice an equal number of `event` and `nonevent` days ("class2") in the training data. The test data days are randomly sampled from the actual data, in which the number of `event` and `nonevent` days differs (`nonevent` days are slightly more frequent). This is typical of a real-life scenario: the class distributions in the training data may differ from those observed when the classifier is applied. You may want to consider the slight class imbalance, but ignoring it may not have a substantial effect on the final results; it is your choice how to handle this issue.

Since we are using Kaggle to collect your submissions, we will use a subset of test samples for the private leaderboard. For those unfamiliar with Kaggle, the submission's score on the private test data rows will be used to determine the final standings. This "private leaderboard" is only viewable by the competition host until the competition deadline, after which we publish it for participants.

We have two deadlines for the term project (see Moodle for deadlines). For the first deliverable, you should submit the following:

- Your predictions for Kaggle.

- A preliminary version of your report to Moodle.

After this deadline, we will publish the private leaderboard scores on Kaggle.

The preliminary report should describe the work completed to date. This report does not need to be 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. However, please do not simply copy the method used by the teams with good performance in the competition!

The Final Report

You should submit the final report as a PDF file via Moodle (see the deadline on Moodle).

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

- The names of the group members.
- The name of the team you used to submit the predictions on Kaggle.
- The stages of your data analysis, including how you looked at the data to understand it (visualisations, unsupervised learning methods, etc.).
- Description of the considered machine learning approaches and the 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 classification model performed.
- As a final section, please include a self-grading report (at most 1 page) that suggests a grade for the term project (integer 0–5) by using the attached grading instructions (see below).

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

Practical instructions for writing the report:

- Your report should read like a self-contained blog post or technical report that is understandable to a data science professional unfamiliar with this specific assignment. 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 the work based on your report alone. Put some emphasis on readability (one of the grading criteria): imagine that the report's reader would be your future boss, who appreciates a clear and concise presentation.
- A strong report goes beyond simply listing what you did. For each major decision in your project (e.g., choice of algorithm, feature engineering approach, hyperparameter selection method), explain *why* you made that choice, *how* you reached that decision (e.g., what alternatives you considered, what criteria you used, computational experiments you ran), and reflect critically on the outcomes. For example, instead of writing "We used a random forest classifier," write "We chose a random forest classifier because [reasoning], comparing it against [alternatives] using [criteria]. This choice proved [effective/problematic] because [reflection on results]." (This is just an illustrative example to demonstrate the point: in a real report, you probably want to use more than one sentence to reflect these issues!) Please read the grading instructions below, especially the criteria for grade 5, on what a good report should look like.
- The performance measures of the predictions on the test data are not a grading criterion by themselves, even though poor performance may indicate that there is something wrong in your approach, which could affect grading. Make sure to reflect on your performance in the challenge final report (was the performance as you expected; if not, why?).
- You are not required to hand in any program code. Therefore, your report should not look like a code dump! Your report may contain code snippets *if* you explain what the reader is supposed to conclude

from your code. If you want to include more significant chunks of code, please put them in an appendix after the main report text. We may look at your code, but we won't go fishing for results or missing details from your code. In other words, all relevant parts of your report should be understandable without going through any code. We may grade your work without going through any code.

- Your report may include code snippets, tables, or figures. Always explain in detail what the code snippets, tables, or figures show and what the reader expects to conclude from them. If you have a code snippet, 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 any readable font (e.g., 12 pt serif font), margins, and appropriately sized figures. Note that Jupyter Notebooks often produce poorly formatted PDFs; we strongly recommend that you write your report with something other than "vanilla" notebooks. It is a good idea to learn a system for writing a proper report, also for future studies and life after graduation!
- Out of curiosity, I took a random sample of 16 similar final reports that got a grade of 5 from an earlier course. The task was identical to this one, but without self-grading (which may add a 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 ranged from 7 to 14 pages, with a median of 12.5.

Even though you can modify your approach and adjust your algorithms for the final report as compared to the preliminary 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 have planned.

The term project will be graded on a scale of 0 to 5 (1–5 = pass); see the grading criteria below.

Grading of the Term Project

At the end of the course, you will be asked to give your project deliverables (including final report, presentation—if you are asked to present your work, 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 grading section should be no more than 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 get in touch with the course staff 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 (including final report, presentation—if you are asked to present your work, 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, but better results in another area can compensate. You should try to balance your weaknesses and strengths and produce a single grade that accurately reflects your group's deliverables.

Notice about the challenge submission: the performance measures of the predictions on the test data are not a grading criterion by themselves, even though poor performance may indicate that there is something wrong in your approach, which could affect grading.

In addition to the numeric grade, explain briefly (max. 1 page) the reasons for your grading using the grading criteria described below. The grading criteria are similar to those for the Data Science Master's Thesis assessment. Please do not just repeat the grading criteria; tell how they apply and relate 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. The discussion of findings shows an aptitude for independent, critical, and innovative research and thinking. The reporting is polished and "camera-ready." The work has been creative and independent, and progressed within the given schedule. The deliverables have been completed using the provided instructions.

Grade 4 (very good): Exceeds 3 but does not meet 5.

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 reporting is accurate, although the style may vary. The work has primarily proceeded according to the planned timetable. The deliverables mostly follow the instructions given.

Grade 2 (satisfactory): Exceeds 1 but does not meet 3.

Grade 1 (passable): The topic and scope have not been motivated clearly, nor have the subject and goals been fully understood. The work shows significant shortcomings in domain knowledge, and the cited sources are generally few or of substandard quality. The reporting and analysis of the results have substantial weaknesses. The conclusions and discussion do not follow the scientific style. The deliverables are unpolished. The work has not progressed as planned. A significant portion of the instructions given was not observed. However, you have submitted the deliverables, and they meet the minimal requirements.

Grade 0 (fail): You have not submitted the deliverables, or they 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 explain your grading (typically 1 paragraph of text). 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 one's 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 that 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 not all of the objectives may 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 others. Responsibilities are distributed fairly. The atmosphere of the group encourages learning and doing the work. Any conflict situations are resolved and learned from.	The group members participate in the meetings actively. Responsibilities and workload are distributed fairly. The atmosphere is good, and conflicts are being resolved.	The group has difficulties agreeing on meeting times, and not all members participate. Responsibilities and workload are uneven. Some do most of the work, others almost nothing. The atmosphere does not encourage learning, and conflicts are unresolved.
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.