

COMP-598: Applied Machine Learning

Mini-project #2: Text Classification

Due on October 14, 11:59pm.

Background:

For this project, you will participate in an in-class Kaggle competition on Text Classification. The goal is to devise a machine learning algorithm to analyze abstracts from the arXiv website, and automatically classify them according to their topic (stats, math, physics or CS). You will be able to download a training set, including labels, as well as a test set that doesn't include labels.

The competition, including the data, is available here (you must create a Kaggle account using your *mail.mcgill.ca* account.): <https://inclass.kaggle.com/c/abstract-classification>

A description of the dataset is included: <https://inclass.kaggle.com/c/abstract-classification/data>

Performance on the Kaggle Leaderboard will be calculated based on % of instances in the test set that are correctly classified.

The project should be completed in a group of 3. Remember: you must work with different team members on each mini-project. You can use the class discussion board on myCourses to find team members. A facebook group was also created to facilitate finding team members (see discussion board for the link).

Requirements:

To participate in the competition, you must submit a list of predicted outputs for the test instances on the Kaggle website (see example file on Kaggle website for the file format). You can submit multiple prediction entries throughout the competition, and track your performance on the Kaggle Leaderboard. The test set is divided into two parts; one set (the public set) is used to update the scoreboard, one set (the private set) is used to calculate the final score of each participant.

To solve the problem, you should try the following methods (the 3rd one is optional):

- 1) A baseline algorithm (from lectures 2-5), such as Naïve Bayes, fully implemented by your team.
- 2) A standard algorithm (from lectures 8-12), such as SVM, nearest neighbor, or decision trees, fully implemented by your team.
- 3) Any other machine learning method of your choice. Existing packages can be used, e.g. *scikit-learn*, if you use them appropriately, and provide appropriate references in your report.

Your written report should include results from all methods considered (so at least 1 *baseline* algorithm and 1 *standard* algorithm). For the Kaggle competition, you can submit results from your best performing method, from any of these categories.

You must submit a report describing your approach. It should contain the following:

- Your team name (feature in the title).
- A list of team members (enter them as “authors” on the submission website).
- Main text with the following sections:
 - o Introduction (overview of approach)
 - o Related work (previous literature related to this problem)
 - o Data pre-processing methods
 - o Feature design/selection methods
 - o Algorithm selection (for each of the categories *baseline/standard/advanced*)
 - o Optimization (if required for the algorithm)
 - o Parameter selection method (model order, learning rate, etc.)
 - o Testing and validation (detailed analysis of your results, outside of Kaggle)
 - o Discussion (pros/cons of your approach & methodology).
- When appropriate, use figures, tables and graphs to illustrate your work. Always include captions, axes labels, etc.
- Use appropriate referencing style throughout your report (with references listed in a separate section near the end, usually after the appendix).
- Before the references, add the following statement: “We hereby state that all the work presented in this report is that of the authors.” Make sure this statement is truthful!
- Spell-check and proof-read carefully.

The main text of the report should not exceed 6 pages. The appendix and references can be in excess of the 6 pages. The format should be double-column, 10pt font, min. 1” margins. You can use the standard IEEE conference format, e.g.

ewh.ieee.org/soc/dei/ceidp/docs/CEIDPFormat.doc. Only acceptable file format is .pdf.

Evaluation criteria:

Marks will be attributed based on: 33% for performance on the private test set in the competition; 67% for the report including a clear description of the methods. The code will not be marked, but may be used to validate the other components.

For the competition, the performance grade will be calculated as follows: The top team, according to the score on the private test set, will receive 100%. A Random predictor, entered by the instructor, will score 0%. All other grades will be calculated according to interpolation of the Leaderboard scores between those two extremes.

For the report, the evaluation criteria include:

- Quality of review of related work
- Technical soundness of proposed methodology (feature selection, algorithms, optimization, validation plan)
- Clarity of methodology description
- Overall organization and writing.

The same evaluation criteria will be used for peer-reviews and evaluation by TAs and instructor. The final grade will be attributed using 50% weight from peer-reviews, 50% grade from TAs and instructor. In the case of missing peer-reviews, the balance of the weight will be shifted to the TAs and instructor evaluation.

Final grades and any late penalties will be attributed per team (i.e. all team members will get the same grade.) You can discuss methods and technical issues with members of others teams, but you cannot share any code or data with other teams.

Any team found to cheat (e.g. use external information, use resources without proper references) on either the code or report will receive a score of 0 for both the report and competition.

Submission instructions:

Predictions on the test set must be submitted on Kaggle:
<https://inclass.kaggle.com/c/abstract-classification/submit>

For the report, we will be using the same online conference management system:
<https://easychair.org/conferences/?conf=comp598>

You should use the same account as for the previous project, but submit to a new track, called “Mini-project #2”. The new report should be submitted as a “New Submission” (one per group), linking other team members as co-authors.

You should submit the code developed for this project as an “Attachment”. Acceptable file formats for this are *.zip*, *.gz*, *.tar*, *.tgz*. Make sure that the code is set up so that we can run it (e.g. include a README file).