# Kaggle Competition IFT3395/6390

September 23, 2020

## 1 Background

For this project, you will take part in a Kaggle competition based on text classification. The goal is to design a machine learning algorithm that can automatically sort short texts into a pre-determined set of categories. The dataset that we have prepared contains abstracts from scientific papers on arXiv. We have selected 15 categories from arXiv: these will be our 15 pre-determined categories (classes). We have sampled 500 papers from each category to be used as the training set, and 1,000 papers as test set. You will implement and train a few different classifiers and will be evaluated on the test accuracy that your trained models achieve.

The competition, including the data, is available here: `https://www.kaggle.com/c/ift3395-6390-arxiv`.

## 2 Important dates and information

- **Sept 28th 23:59** Deadline for IFT3395 students to enter the competition and form teams. Deadline for IFT6390 students to enter the competition on Kaggle.
- **Oct 19th 23:59** Competition ends. No more Kaggle submissions are allowed.
- **Oct 27th 23:59** Reports and code are due on Gradescope.

**Note on sharing and plagiarism**: You are allowed to discuss general techniques with other teams. You are NOT allowed to share any of your code. This behavior constitutes plagiarism and it is very easy to detect. All teams involved in sharing code will receive a grade of 0 in the data competition."

## 3 Enter the competition and team formation

**IFT6390** students must do the competition alone (1-person team). **IFT3395** students will work in teams of 2 or 3.

### 3.1 Kaggle Team formation (IFT3395 students only)

To form a team:

- Enter the competition and create a Kaggle account if you are not registered yet by following the link: `https://www.kaggle.com/t/6200fceaac08449e96eb4b68d9718f51`
- In the "Invite Others" section, enter your teammates' names, or team name.
- Your teammate has the option to accept your merge.
- Fill out the google form `https://forms.gle/kxYbUyFweXUtBJbs7` with your team information by **Sept 28th at 23:59**. Any teams not registered or registered late will not be graded.

**Important note:** The maximum amount of submissions is 2 per day, per TEAM. Any team whose individual members have a submission count larger than what is allowed up to-date will be UNABLE to form a team. Example: Today is the first day of competition. A,B,C are three teammates who haven't formed a team yet.

- A submitted 0 times.
- B submitted 2 times.
- C submitted 1 time.

Because the maximum amount of submissions is 2 per team per day, the total possible submissions for a team is 2. However, the cumulative submission count for A,B,C is 3. Therefore, they will be unable to form a team (They will need to wait for tomorrow, and not submit any submissions for the next day).

You can start submitting solutions before you form a team, as long as you are careful about the above limitation when forming teams.

### 3.2 Enter the competition (IFT6390 students only)

- Enter the competition and create a Kaggle account if you are not registered yet by following the link: `https://www.kaggle.com/t/6200fceaac08449e96eb4b68d9718f51`

## 4 Naive Bayes baselines

You will need to build a Naïve Bayes classifier using Bag of Words features, and beat the baselines highlighted in the leaderboard. These baselines are:

- a Random classifier that randomly picks a class for each test example
- a Naïve Bayes classifier using Bag of words features
- the TA's best baseline

To participate in the competition, you must provide a list of predicted outputs for the instances on the Kaggle website. You can submit 2 predictions per day over the course of the competition, so we suggest you start early, allowing yourself enough time to submit multiple times and get a sense of how well you are doing.

For each of the 3 baselines that you beat, you get extra points.

Your Naive Bayes classifier must be implemented from scratch. Apart from standard Python libraries, the only libraries allowed are `numpy`, sparse matrices from scipy (`scipy.sparse`), and `pandas`.

# 5 Other models

You must try at least **1 other model** than Naive Bayes, and compare their performances. You are encouraged to implement techniques studied during the course, and look up for other ways to solve this task. Here are a couple of possibilities:

- Kernelized SVM using string kernels
- Random Forests
- Hand-crafted features and logistic regression
- any other algorithm of your choice...

The goal is to design the best performing method as measured by submitting predictions for the test set on Kaggle. Your final performance on Kaggle will count as a criterion for evaluation (see below). If a tested model does not perform well, you can still add it in your report and explain why you think it is not appropriate for this task. This kind of discussion is an important feature that we will be using to evaluate your final competition report.

For this part, you are free to use any library of your choice.

# 6 Report

In addition to your methods, you must write up a report that details the preprocessing, validation, algorithmic, and optimization techniques, as well as providing results that help you compare different methods/models. The report should contain the following sections and elements. You will lose points for not following these guidelines.

- Project title
- Team name on Kaggle, as well as the list of team members, including their full name and student number.
- Introduction: briefly describe the problem and summarize your approach and results.
- Feature Design: Describe and justify your pre-processing methods, and how you designed and selected your features.

- Algorithms: Give an overview of the learning algorithms used without going into too much detail, unless necessary to understand other details.
- Methodology: Include any decisions about training/validation split, distribution choice for naïve bayes, regularization strategy, any optimization tricks, setting hyper-parameters, etc.
- Results: Present a detailed analysis of your results, including graphs and tables as appropriate. This analysis should be broader than just the Kaggle result: include a short comparison of the most important hyperparameters and all methods (at least 2) you implemented.
- Discussion: Discuss the pros/cons of your approach & methodology and suggest ideas for improvement.
- Statement of Contributions. Briefly describe the contributions of each team member towards each of the components of the project (e.g. defining the problem, developing the methodology, coding the solution, performing the data analysis, writing the report, etc.) At the end of the Statement of Contributions, add the following statement: We hereby state that all the work presented in this report is that of the authors.
- References (very important if you use ideas and methods that you found in some paper or online; it is a matter of academic integrity).
- Appendix (optional). Here you can include additional results, more details of the methods, etc.

**The main text of the report should not exceed 6 pages.** References and appendix can be in excess of the 6 pages.

Your must submit your report and your code on Gradescope before ~~**Oct 16th 23:59**~~ **Oct 27th 23:59**.

## Submission Instructions

- You must submit the code developed during the project. The code must be well-documented. The code should include a README file containing instructions on how to run the code.
- The prediction file containing your predictions on the test set must be submitted online at the Kaggle website.
- The report in pdf format (written according to the general layout described above) and the code should be uploaded on Gradescope.

# 7 Evaluation Criteria

Marks will be attributed based on the following criteria:

1. You will be assigned points for each one of the 3 baselines that you beat.
2. You will be assigned points depending on your final performance at the end of the competition, given by your ranking in the private leaderboard.
3. You will be assigned points depending on the quality and technical soundness of your final report (see above).