# Kaggle Competition 1 IFT 6390/3395

November 4, 2022

## 1 Introduction

For this project, you will take part in a Kaggle competition about detection of handwritten digits. The goal is to design a machine learning algorithm that can automatically classify images. Figure 1 shows an example image from the training set. All the images (including held out test samples) are of dimension $56 \times 28$. The class for a image is denoted by the sum of digits in the image. For example, the class of image in figure 1 is 10. We have generated this dataset from the popular MNIST dataset. The training and held-out test set for this data consists of 50,000 samples and 10,000 samples respectively. You are required

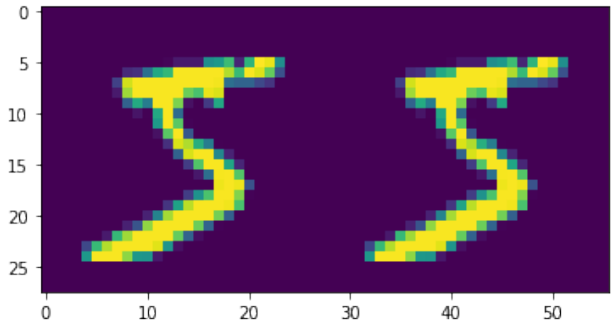


Figure 1: Example image from the training set

to implement and train several classification algorithms based on the dataset provided. The evaluation will be based on the performance on the held out test set and a written report.

The competition, including the data, is available here:

https://www.kaggle.com/t/0d0b1c033ece47ffa1dbc8bd374689ae

## 2 Important dates and information

Please take into consideration the following important deadlines:

- **October 5th 23:59** Competition released

- **October 12th 23:59** Deadline to enter the competition on Kaggle.
- **October 26th 23:59** Competition ends. No more Kaggle submissions are allowed.
- **November 3rd 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 Join the competition

**IFT6390** students must do the competition alone (1-person team) (**IFT3395** students will participate 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/0d0b1c033ece47ffa1dbc8bd374689ae
- 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/Jrscpc5p6ee3x9Vw7` with your team information by **October 12th at 23:59**. Any teams not registered or registered late will not be graded.

**Important note:** The maximum amount of submissions per day and per team is 2. 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.

# 4 Baselines

We ask you to build a **logistic regression** classifier and beat the baselines highlighted in the leaderboard. These baselines are:

- a dummy classifier that simply predicts the most frequent class in the training set.
- a logistic regression classifier.
- 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 logistic regression 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** besides logistic regression, 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 few ideas:

- Support Vector Machines
- Naïve Bayes
- Random Forests
- Hand-crafted features that take into account the nature of the data

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 Section 6). 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 pre-processing, 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:

- Project title
- Full name, student number and Kaggle username.
- 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, regularization strategies, any optimization tricks, choice of 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 hyper-parameters and all methods (at least 2) you implemented.
- Discussion: discuss the pros/cons of your approach and methodology and suggest ideas for improvement.
- Statement of Contributions. add the following statement: "I hereby state that all the work presented in this report is that of the author".
- 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.

You will lose points for not following these guidelines. **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 **November 3rd 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).