Kaggle Competition 2 IFT6390 students

November 16, 2022

1 Background

For this project, you will take part in a 2nd Kaggle competition based on text classification. You are provided with over 1 millions texts which are annotated as negative, positive and neutral. The task is to create features based on the provided data and use machine learning algorithms to classify them.

You are required 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/6c98b08e131d49abaa8915175de22ced

2 Important dates and information

Please take into consideration the following important deadlines:

- **November 16th** Competition released
- **November 22th 23:59** Deadline to enter the competition on Kaggle.
- **December 9th 23:59** Competition ends. No more Kaggle submissions are allowed.
- **December 14th 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 form teams

IFT6390 students will participate in teams of 2.

3.1 Kaggle Team formation

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/6c98b08e131d49abaa8915175de22ced
- 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/j96iySG3TJAFrXBg8` with your team information by **November 22nd 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 3. 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 1 times.
- B submitted 2 times.
- C submitted 1 time.

Because the maximum amount of submissions is 3 per team per day, the total possible submissions for a team is 3. However, the cumulative submission count for A,B,C is 4. 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 Requirements

For this competition, you are allowed to use any library function of your choice. To get full grades, you are asked to implement **the following 4 algorithms**, and report their performances. However, even if you use a library, it is important that you discuss each hyperparameter, and how you chose their values. Additionally, you will be uploading only the best performing model on Kaggle. But, it is important to report to accuracy of each of these models on the on the validation and test sample (you can report the score you received on kaggle as you don't see the complete held-out test set).

Here are the algorithms to implement:

1. a Naïve Bayes classifier using Bag of words features
2. Kernelized SVM using string kernels

3. Neural Nets

4. 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), as well as the number of baselines that you beat. 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.

4.1 Explainability in Machine Learning

Machine learning models are often treated as black-boxes where the reason why a decision is made by the classifier is unknown. Hence, algorithms such as Local Interpretable Model-Agnostic Explanations(LIME) [1] or Grad-CAM [2] are helpful.

In addition to the Kaggle competition, you are expected to implement any explainable algorithm of your choice and explain the decision made by the model you trained. You are allowed to used any library function for this implementation.

5 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, 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 3) 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 8 pages. References and appendix can be in excess of the 10 pages.

Your must submit your report and your code on Gradescope before **Dec 14th 23:59**.

Please note: We will not be accepting submissions for the outputs other than ones that are directly made on Kaggle.

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.
- You can submit *.ipynb files but it is compulsory to submit the associated *.py file.
- Please submit the *.ipynb file associated with the explainability of your model.

6 Evaluation Criteria

Marks will be attributed based on the following criteria:

1. You will be assigned points if you beat the baseline.
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).

4. The complete breakdown of grading is - Competition : 35 (Algorithm 1-4: 30, Ranking on the Leaderboard: 5); Report : 40 (Format : 5, Algorithms : 10, Methodology : 15, Discussion of results (including explainability) : 10); Explainability: 15; Code : 10

References

[1] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[2] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.