Mini-Project 2: Machine Learning

CP322-A, Fall-2023, Wilfrid Laurier University

General information

- **Due on 12-Nov-2023 at 11:30 pm**. Late work will be automatically subject to a 20% penalty and can be submitted up to 5 days after the deadline. No submissions will be accepted after these 5 days.
- all members of a group will receive the same grade.
- You will submit your assignment on MyLS as a group. The group leader is responsible for the assignment submission. Other group members should make sure that their leader has submitted the assignment on time before the deadline
- You must submit two separate files to MyLS (**using the exact filenames and file types outlined below**):
  - **CP322-A2-GroupID-code.zip**: Your data processing, classification, and evaluation code (.py and .ipynb files).
  - **CP322-A2-GroupID-writeup.pdf**: Your (max 5-page) project write-up as a pdf (details below).
- Except where explicitly noted, you are free to use any Python library or utility for this project.

Problem definition

In this mini-project, you will develop models to classify textual data, the input is text documents, and the output is categoricalvariables (class labels).

Datasets

Use the following datasets in your experiments.

- 20 newsgroup dataset. Use the default train subset (subset='train', and remove=(['headers', 'footers', 'quotes']) in sklearn.datasets) to train the models and report the final performance on the test subset. Note: you need to start with the text data and convert the text to feature vectors. Please refer to `https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html` for a tutorial on the stepsneeded for this.
- IMDB Reviews: `http://ai.stanford.edu/~amaas/data/sentiment/` Here, you need to use only reviews in the train folder for training and report the performance from the test folder. You need to work with the text documents to build your own features and ignore the pre-formatted feature files.

Models

Apply and compare the performance of the following models:

- Logistic regression: sklearn.linear_model.LogisticRegression
- Decision trees: sklearn.tree.DecisionTreeClassifier

- Support vector machines: sklearn.svm.LinearSVC
- Ada boost: sklearn.ensemble.AdaBoostClassifier
- Random forest: sklearn.ensemble.RandomForestClassifier

You are welcome and encouraged to try any other model covered in the class, and you are free to implement them yourself or use any Python library that has their implementation, e.g., the above links from the SciKit learn package. You need to still understand what is the exact model being used. You are also free to use any Python libraries you like to extract features and preprocess the data, and to tune the hyper-parameters.

Validation

Develop a model validation pipeline (e.g., using k-fold cross-validation or a held-out validation set) and study the effect of different hyperparameters or design choices. In a single table, compare and report the performance of the above-mentioned models (with their best hyperparameters), and mark the winner for each dataset and overall.

Write-up instructions

The project write-up is a five-page PDF document (single-spaced, 10pt font or larger; extra pages allowed for references and appendices). Using LaTeX is recommended, and overleaf is suggested for easy collaboration. *You are free to structure the report how you see fit*; below are general guidelines and recommendations, *but this is only a suggested structure and you may deviate from it as you see fit.*

- Abstract (100-250 words) Summarize the project task and your most important findings.
- Introduction (5+ sentences) Summarize the project task, the dataset, and your most important findings. This should be similar to the abstract but more detailed.
- Related work (4+ sentences) Summarize previous literature related to the multi-class classification problem and text classification.
- Dataset and setup (3+ sentences) Very briefly describe the dataset and explain how you extracted features and other data pre-processing methods that are common to all your approaches.
- Proposed approach (7+ sentences) Briefly describe the different models you implemented/compared and the features you designed, providing citations as necessary. *If you use or build upon an existing model based on previously published work, it is essential that you properly cite and acknowledge this previous work.* Discuss algorithm selection and implementation. Include any decisions about training/validation split, regularization strategies, any optimization tricks, setting hyper-parameters, etc. It is not necessary to provide detailed derivations for the models you use, but you should provide at least a few sentences of background (and motivation) foreach model.
- Results (7+ sentences, possibly with figures or tables) Provide results on the different models you implemented (e.g., accuracy on the validation set, runtimes).
- Discussion and Conclusion (3+ sentences) Summarize the key takeaways from the project and possibly directions for future investigation.
- Statement of Contributions (1-3 sentences) State the breakdown of the workload.

Evaluation

The mini-project is out of 10 points, and the evaluation breakdown is as follows:

- Completeness (3 points)
  - Did you submit all the materials?
  - Did you run all the required experiments?
  - Did you follow the guidelines for the project write-up?
- Correctness (4 points)
  - Are your models implemented correctly?
  - Are your reported accuracies close to the reference solutions?
  - Do your proposed features improve performance, or do you adequately demonstrate that it was not possible to improve performance?
  - Do you observe the correct trends in the experiments (e.g., comparing learning rates)?
- Writing quality (2 points)
  - Is your report clear and free of grammatical errors and typos?
  - Did you go beyond the bare minimum requirements for the write-up (e.g., by including a discussion of related work in the introduction)?
  - Do you effectively present numerical results (e.g., via tables or figures)?
- Originality/creativity (1 point)
  - Did you go beyond the bare minimum requirements for the experiments? For example, you could investigate different stopping criteria for logistic regression, investigate which features are the most useful (e.g., using correlation metrics), or propose an automated approach to select a good subset of features.
  - **Note:** Simply adding in a random new experiment will not guarantee a high grade on this section! You should be thoughtful and organized in your report.
- Best performance bonus (+1 point)
  - The top-performing group (groups, if there are ties) will receive a bonus.

Final remarks

You are expected to display initiative, creativity, scientific rigor, critical thinking, and good communication skills. You don't need to restrict yourself to the requirements listed above - feel free to go beyond, and explore further.

You can discuss methods and technical issues with members of other teams, but **you cannot share any code or data with other teams.**