CS5785 / ORIE5750 / ECE5414 Midterm Project

This midterm project is due on **Thursday, October 31st, 2024 at 11:59PM ET**, uploaded to Gradescope (Canvas->Gradescope) and Kaggle. Your submission will have three parts:

1. Join the class Kaggle competition through this link: https://www.kaggle.com/t/6977532b78b349268b47c7a09274b83a
2. A write-up as a single `.pdf` file. Submit this under the `midterm-report` assignment in Gradescope.
3. Source code and data files for all of your experiments (AND figures) in `.ipynb` files (file format for IPython Jupyter Notebook). These files should be placed in a folder titled `midterm` and uploaded to the `midterm-code` assignment in Gradescope. You could use colab notebook template as a starting point.

The write-up should contain a general summary of what you did, how well your solution works, any insights you found, etc. On the cover page, include the class name, and the names of the entire team. In addition, briefly describe the responsibility of each team member. You are responsible for submitting clear, organized answers to the questions. You could use online LaTeX templates from Overleaf, under "Homework Assignment" and "Project / Lab Report".

Please include all relevant information for a question, including text response, equations, figures, graphs, output, etc. If you include graphs, be sure to include the source code that generated them. Please pay attention to Canvas for announcements, policy changes, etc. and Piazza for homework related questions.

IF YOU NEED HELP

There are several strategies available to you.

- Please ask clarification questions on Ed. That way, your solutions will be available to other students in the class.
- You might be able to get some hints by attending the professor and TAs office hours, but the hints are not guaranteed.
- You are allowed to use well known libraries such as `scikit-learn`, `scikit-image`, `numpy`, `scipy`, etc. for this assignment (including implementations of machine learning algorithms), unless we explicitly say that you cannot in a particular question. Any reference or copy of public code repositories should be properly cited in your submission (examples include Github, Wikipedia, Blogs).

PROJECT GUIDELINES

1. **Project Overview** In class, we have worked with text datasets with labels for supervised learning based binary classification, and unsupervised learning techniques. But what about semi-supervised learning tasks? Semi-supervised learning is a machine learning paradigm in which a model is trained on a dataset that contains both labeled and unlabeled examples. In this project, you will investigate semi-supervised learning techniques applied to the multiclass classification task involved in sentiment analysis, where the labels are incomplete. This is common in the real world, where data acquisition devices/sensors fail, or human annotators are too expensive/inaccessible. This project will therefore consist of two parts:
   (a) Part 1: Use unsupervised (or supervised or hybrid) learning techniques to automate the addition of labels to the data examples with missing labels (i.e., augment your dataset).
   (b) Part 2: Apply a supervised learning technique on this augmented dataset, and see if the sentiment classification accuracy/F1-score/any other metrics improves (i.e., does this sentiment classification task via supervised learning improve after augmenting the dataset?)

2. **Project Motivation** The project is intended to give you experience working on both unsupervised and supervised learning methods in a more realistic setting. It is also intended to give you experience working with some of the topics covered in the last five weeks of class, for which this project is the evaluation. This means, in this project, you should evaluate at least one of the unsupervised algorithms proposed in the grading rubric at the end of this document, or maybe even machine learning methods we haven't explicitly covered in class. Here is a list of things that you should **not** do in this project:
   (a) Please do not try to reverse engineer the labels on the test set.
   (b) Please do not try to find the data source online and train your model on the fully labeled dataset.
   (c) You may manually label a subset of data, but please do not manually label more than 100 data points.
   If we discover that you did one of the above, you will receive a severe penalty for this project.

3. **Python Packages You Should Use** The goal will be to optimize the classification performance on sentiment analysis, and hopefully doing so by leveraging the entire dataset (even those data examples without any labels). You are allowed to use well known libraries such as `scikit-learn`, `scikit-image`, `numpy`, `scipy`. When required, we only ask that you develop your algorithms in `PyTorch` instead of `Keras`, `TensorFlow`, etc.

4. **Dataset** The dataset is available on Kaggle and on Canvas under Week 8, and is based on the "Rotten Tomatoes movie review dataset". The zip folder will contain three files: train.csv, val.csv and test.csv. All three files will contain data tables, with one column containing the input text data (X) and another column containing the output label (y). 60% of the data examples in the train.csv file will have missing labels (i.e. a label with value -100), and the remaining 40% will have valid labels (i.e. in the range [0,1,2,3,4]). All examples in the val.csv file will have valid labels (i.e. in the range [0,1,2,3,4]). You may use any preprocessing steps you like on the raw input text data (X) – you may choose to refer to your previous homework assignments. Once you are satisfied with your algorithm hyperparameter tuning on the val set, you will have to submit your predicted labels to kaggle to obtain a final test score, at this link. Please note that the Kaggle page states you can only make 20 submissions per day, so please plan ahead accordingly.

PROJECT GRADING

We will now outline the criterion by which we will grade the project. Note, some projects will be more creative than others; some projects will achieve higher performance than others. We will provide room for extraordinary work in one category to compensate for deficiencies in another category. These are the general areas we will look into. Concretely, the midterm project will be graded on a scale of 100 points, but each section is assigned points so that the sum total can exceed 100 points. Your final project score will be capped at 100 points. You should aim to do a good job in all areas.

1. **Creativity (35 points maximum)** How creative and/or diverse is the approach? Have you implemented / tried various algorithms? Are multiple classes of algorithms compared? An example of a project that we would count as creative is implementing and comparing at least two options for each of the below categories (2 data-augmentation methods, 2 supervised methods, and 2 data pre-processing methods). Alternatively, you can create *ensemble methods* that combine the predictions or decisions of multiple individual models. Here are some suggestions (you may use other suitable methods as well):

   (a) Distribution based clustering methods (e.g., GMM)
   (b) Density based clustering methods (e.g., DBSCAN, note that we have not covered DBSCAN in the class but you can try it out)
   (c) Centroid-based clustering methods (e.g., K-Means)

   Some combination of the following supervised methods:

   (a) Multinomial Naive Bayes
   (b) Softmax Regression
   (c) K Nearest Neighbors

   Some combination of the following unsupervised methods:

   (a) Bag of words (refer to your homework 2 solution)
   (b) N-gram (refer to your homework 2 solution)
   (c) GLoVe Embedding (refer to the midterm project template notebook provided on Canvas)

   To be fair to students who have not taken a ML class, please **do not** use deep learning methods in this midterm project. You will have the chance to do that in the final project.

2. **Insight (35 points maximum)** Does the project reveal some insight about the choice of hyperparameters, architectures, etc. on the performance of the algorithm? Is there reasonable insight / explanation / intuition into the results? (i.e., you should not just blindly apply different algorithms to a problem and compare them.)

3. **Performance (30 points maximum)** Does the project achieve relatively good performance on the problem, given the computational resource that you have access to? How do different algorithms compare? Did you to optimize the algorithms chosen instead of just randomly training a few different algorithms without optimizing them?

4. **Report Writeup (15 points maximum)** Are the approach, insight, and results clearly presented and explained? Dissemination of results is an important component to any project.

PROJECT REPORT WRITEUP

Each group should submit a writeup of their project work, exceeding no more than 5 pages including figures. References are excluded from the 5 pages (e.g., they may overflow onto a 6th page and be numerous). It is fine to be below the page limit; this is the maximum. We appreciate conciseness. We will also ask you to submit your **code** and **dataset**, so that we can validate your results.

The writeup must adhere to the following template, accessible via this link – so that we can judge all writeups in the same manner without having to worry about different font sizes, etc.

Your report writeup should contain the following sections:

1. **Abstract** A brief description of what you did in the project and the results observed.
2. **Methods** State the details about your models. This section should summarize all the methods that you have tested. Specifically, explain the details about 1) your model (e.g., how did you initialize the cluster centroids or GMM clusters) and 2) the training (e.g. for EM algorithm, the threshold parameter determined to stop the algorithm convergence, the number of iterations, etc.).
3. **Results** State the results of your experiments. This section should include a table summarizing the performance of your models.
4. **Discussion** Discuss insights gained from your project, e.g., what resulted in good performance, and any hypotheses for why this might be the case.
5. **References** List references used in your writeup. We expect you to cite all papers, books and websites used for ideas, code and phrasing. Please review the Canvas course syllabus for our policy on Academic Integrity.