CS 3780/5780: Introduction to Machine Learning **(Due: 12/09/24)**

Creative Project: Emotion Classification of Natural Language

Instructor: Sarah Dean, Thorsten Joachims, John Thickstun

Course Policy: Read all the instructions below carefully before you start working on the creative project, and before you make a submission.

1 Introduction

The creative project is about conducting a real-world machine learning project on your own, with everything that is involved. Unlike in the programming projects 1-5, where we gave you all the scaffolding and you just filled in the blanks, you now start from scratch. The past programming projects provide templates for how to do this (and you can reuse part of your code if you wish), and the lectures provide some of the methods you can use. So, this creative project brings realism to how you will use machine learning in the real world.

The task you will work on is classifying texts to human emotions. Through words, humans express feelings, articulate thoughts, and communicate our deepest needs and desires. Language helps us interpret the nuances of joy, sadness, anger, and love, allowing us to connect with others on a deeper level. Are you able to train an ML model that recognizes the human emotions expressed in a piece of text?

2 Dataset

We have three parts of the dataset: a training sample, a public test sample and a private test sample. The training sample contains 10,000 data points. The public test sample contains 10,000 data points. The private test sample contains 5,000 data points. We are given natural language text x that express a certain emotion y, which belongs to one of a fixed set of the emotion classes. There are 28 classes in total.

We provide both text x and label y for the training sample, while we only provide text x for the public and private test samples. You will submit your predictions for the entire test sample, and we will take care of splitting them into the public and private part.

We are using Kaggle as the platform for running this project, and you can find a link to our Kaggle project at the end of this document. The following files are provides on the Kaggle platform:

- **train.csv**: The first column contains the texts. The second column contains the labels. Each row contains a piece of text that corresponds to one of the 28 emotions, which is represented by an index from 0 to 27.
- **test.csv**: The first column contains the ids. The second column contains the texts. You should classify a text to one specific emotion class. The submission format is as in sample_submission.csv.
- **notebook_template.ipynb**: Please follow the structure of this template, and make changes following its instructions. We will use this file for grading. We ask you to submit the PDF version of the Jupyter Notebook on Gradescope, and the executable IPYNB version on Canvas.
- **sample_submission.csv**: The first column contains the ids, which are the same ids as in test.csv. The second column contains your model's classification results for the test sample, where the entries should be integers in between 0 to 27.
- **starter.py**: This file contains the starter code for loading the training set and test set data using the Notebook on Kaggle.
- **submission.py**: This file contains the code for saving your model prediction into the correct format that will be used for grading.

3 Your Task

You are required to submit the predictions $h(x)$ of for each data point x in the test set. You can use ML methods that you learned in class to train a model on the training sample, and to extract meaningful underlying representations from natural language that express human emotions. We encourage you to start from the basics, where you represent each piece of text using the bag-of-words representation you have seen multiple times in class and then apply classification models. Then, you can take the challenge to try out representations or variants of algorithms that may work better, like changes to the bag-of-words representation through feature selection or like text embedding models based on (small) LLMs. But keep in mind that your computational resources are limited, and that big LLMs are infeasible. Be creative, and see if you can perform better than the basic methods.

4 Collaboration

This is a group project, and you can work on this project in a group of 2-3 students. You cannot discuss ideas or share code with other groups. If you are stuck in any part and you cannot make progress on your own, make sure to seek help and discuss your issue with the course staff during office hours.

5 Important Rules and Academic Integrity

The following are important rules concerning academic integrity for this project:

- Sharing code or ideas with other groups is not allowed and is considered an academic integrity violation.
- Do not copy substantial pieces of code directly from the internet or from anybody outside your group. Looking for examples of similar applications on popular libraries (like scikit or numpy) is encouraged, but outright copying a complex solution is not allowed.
- You could theoretically label the test sample without machine learning (e.g., manual labeling, external tools), but that is of course not allowed; we expect your notebook when run directly to produce the same output as your competition submission, and you will need to submit an operational version of your notebook that produces the classifications.
- Don't make your team-name anything you wouldn't be willing to shout in front of the CS faculty or your parents (e.g., vulgar language).
- It is fine to use resources like reading published papers and code that implements these papers. But make sure to reference and cite any resource you used in your notebook.

6 Grading

The first 85% of your grade will come from how well you craft a basic solution to the learning problem. You are required to do some kind of train and validation split for model selection and provide test-sample predictions for at least two machine learning algorithms from class. In more detail, this includes:

- Load and preprocess the dataset, and design a machine learning approach to this problem.
- Perform model selection via some form of train and validation split.
- Use at least two different training algorithms from class.
- On the public leaderboard, the goal of your basic solution is to beat "Tiny Piney" with at least one of your methods. You will receive points for reaching that accuracy goal, and for performing and explaining the steps that got you there. We will also award partial credit, if you didn't beat 'Tiny Piney" but show good reasoning for doing the previous steps. So, it is very important to write up in words your thought process in the Notebook, since part of your grade is convincing us of the validity of your approach!

The remaining 15% will be given for creative ideas that go beyond the basics. Again, this gives some realism to the creative project, since no machine learning project is like the other, and they all require some creativity. Here are some ideas of what you could try, but there is really no limit to your creativity in improving on the basic solution.

- Create new features from the features you already have, or apply other learning algorithms that are better suited for this task, or modify some of the learning algorithms to better model this particular problem (e.g. choice of loss function to train with).
- In addition to running the experiments, clearly write up your reasoning and how each thing you tried improved - or did not improve - the results.
- On the public leaderboard, the goal of your creative solution is to beat "Zero Hero". Again, explaining your approach convincingly is part of the grade. We also award partial credit if your reasoning is creative and valid, but you do not beat the baseline.

Be creative, clearly describe what you tried, and report your results in a convincing way in the notebook you submit. If you try creative and well-argued ideas that do not pan out in better prediction performance, this is a perfectly fine outcome and can be an excellent project.

We are also planning to give up to 10% extra credit for particular strong performing algorithms on the private test set. Note that the test set is split to two parts in Kaggle, the leaderboard you see on Kaggle is the public part while the score on the private part of your submission is not visible until the final project is over. We will use the public accuracy for the ordinary grading. But for the extra credit, you have to do well on the private test set. Hence performing well on the public leader board does not guarantee the extra credit, which is again part of the realism of working in machine learning. The proof is in how well your method does in the real world – and the more robust your model selection etc., the more likely your learned rule will do well.

7 Due Date

The final project will be due on Monday, December 9th at 5pm. By that time, you will need to make the following submissions.

- You need to submit an operational version of your Jupyter Notebook on Canvas (e.g., as a ZIP file with all the code that you wrote).
- You need to submit a readable PDF version of your Jupyter Notebook on Gradescope.
- You need to submit all the required test sample predictions (both basic and creative) on Kaggle.

Note that December 9th at 5pm Ithaca time is a hard deadline for this project! No late submissions will be accepted and you cannot use any of your unused slip days. So we highly recommend that you keep submitting preliminary versions well before the deadline, so that you do not end up with an empty submission that will get you zero points due to random network flukes, power outages, snowstorms, illness, notebook-eating AI dogs, etc..

8 Kaggle Competition

Be ready to start the journey! You need to use your Cornell Netid email to log in (or sign up) to Kaggle. There are instructions that lead you to explore the data and the compute resources on Kaggle. Here is the link to the Kaggle competition: https://www.kaggle.com/t/58bc32dd94ca46ac845d29c4e98d1751. Enjoy!