**CS 590 NLP**

**HW4**

**Naïve Bayes**

**Due 03/02 11:59 pm**

**Overall Goal:**

In this homework you will work with Naïve Bayes to classify the same dataset from HW3.

|  |
| --- |
| **Useful python packages:** |
| For Naïve Bayes, there is a sklearn package (**Recommended**):  (<https://scikit-learn.org/stable/modules/naive_bayes.html)>  Or there is a nltk package: (<https://www.nltk.org/_modules/nltk/classify/naivebayes.html>)  You may also implement this manually, but this will require extra time and effort, so I leave it up to you. |

|  |
| --- |
| **Dataset** |
| The dataset that will be used is the toxic comment dataset. This consists of toxic comments, so be cautioned when viewing the dataset.  You can find the training and test sets here: <https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data>  The training set (**train.csv)** will be used for training the LM, while the test set (**test.csv**) and labels **(test\_labels.csv)** will be used for testing and analyzing your models.  Take time to understand how the training set and test set are laid out. |

|  |
| --- |
| **Naïve Bayes Model Tasks** |
| You will implement and test the Naïve Bayes model. This model will predict whether the text is toxic (toxic is 1) or not toxic (toxic is 0).  **Note that your NB model must train on the text itself (CountVectorizer if sklearn) and not use the TF-IDF vectors.**  You will create 2 required functions for NBs (you may create more functions for your own use, but you need to at least create these two as specified):  **train\_NB\_model(path\_to\_train\_file)**  This method trains a naïve bayes model on the training text and returns that trained model.  **The format for the train file should follow the same format as the training data file!**  **test\_NB\_model(path\_to\_test\_file, NB\_model)**  This method tests a trained NB model on some test file and outputs a test file in the same format as the input test file but with 2 columns added: 1) probability of that text being toxic, 2) class prediction (toxic, not toxic). **The format for the input file should follow that of the test file.**  Once these functions are implemented use them to accomplish the following tasks.   1. Train a NB model on the entire train set. 2. Test the trained model on the test set and produce predictions for all test texts. 3. Analyze your accuracy of your NB model and try to improve it. This may be done by: adding other features in, improving your preprocessing of the text, changing the hyperparameters of the NB model for training, etc. **Note: Beware of overfitting your models: e.g. if your model is only predicting one class, then it probably overfit, you may lose points if you do not fix this.** 4. Compare and contrast how the models did on different labeled data (ie toxic vs non toxic texts). 5. Write the analysis and attempts to improve the accuracy as observations in a report document to be handed in alongside code. Note that even if changing something causes the accuracy to drop, this should be reported along with your reasoning of the drop/improvement in accuracy. This report is how you can demonstrate that you are thinking through the problem and give you practice for your final project. |

|  |
| --- |
| **Additional Tips/Guidance** |
| 1. The accuracy you achieve will not determine your final grade for this homework (**except for overfitting)**. Rather your thoughtfulness in approaching and analyzing your models. This means that lack in analysis will receive lower scores as indicated in the grading scale. 2. Note that preprocessing the text is up to you. This assignment is not specifically evaluating preprocessing like the first assignment, but here you have a chance to practice any preprocessing for your final project and future assignments. 3. Get familiar with the python libraries on your own on small sets of data. Just copying and pasting the code without fully understanding it will end up being detrimental as you may not be able to accurately give proper analysis and observations of results (especially for the NB model where you are expected to make improvements/changes).    1. Do not just copy and past the sklearn class code. You may use portions from the sklearn classes, but too much unused code from classes will result in lost points due to poorly written code. Experiment with the code and only import what you need. 4. DO NOT SHARE CODE. I have linked libraries which will be useful for this assignment. You may also come to me to discuss/figure things out. 5. Start the assignment early. You will have 2 weeks to complete this assignment, which is plenty of time if you start early any familiarize yourself with the libraries. **IF YOU WAIT UNTIL THE LAST MINUTE TO START, YOU WILL BE LESS LIKELY TO DO WELL.** I won’t be granting any additional extensions so each late assignment with follow the late grade policy. 6. Ask questions/approach like a researcher. Think like this is your chance to explore NLP models and analyze their effectiveness. |

**Grading**

Assignment will be graded as follows:

|  |  |
| --- | --- |
| **Description** | **Points** |
| Code Runs | 5 |
| NB Implementation/Tasks | 20 |
| Report analysis and observations | 20 |
| Documentation (Comments, functions, etc) | 5 |
| **Total:** | **50** |