Northeastern University, Khoury College of Computer Science

CS 6220 Data Mining — Assignment 6

Due: November 18, 2024(100 points)

YOUR NAME
YOUR E-MAIL

Mining Reviews in Small-ish Datasets

The question in this section is best started with the starter kit Colab. You can copy and past the cells to a `*.py` Python3 file or or simply use the Colab in your own Google Drive.

We'll process datasets that are small enough that they can fit into CPU memory. We'll review the Amazon purchasing reviews dataset. Amazon has collaborated with Universities to aggregate an extensive set of buyer reviews data. Most of the datasets (including the subcategory datasets) cannot fit into CPU RAM, and an even smaller subset of machine learning algorithms can actually process them.

With this dataset, you will need to do the following for question number 2. Typically we would split the dataset into training / validation / testing dataset, but we will only split into training / testing dataset in this assignment.

1. With the Amazon magazines dataset, do the following. (Feel free to use the `scikit-learn` library).

   - Plot a histogram of the data with the number of positive and negative reviews.
   - Balance the data so that you are training with equal probabilities, $P(\text{Liked}) = P(\text{Not Liked}) = 0.5$. We will use the original distribution for the evaluation dataset.
   - Use machine learning models to predict whether or not a review as written will result in a "Liked" rating. Try changing different parameters, including the maximum vocabulary size. Also try normalization.
   - Try out Naïve Bayes, Decision Trees, Random Forests, and Logistic Regression machine learning algorithms
   - Print out your best accuracy, precision, and recall numbers for each algorithm.

How do they compare? Which algorithms are overfitting?

Tips and Tricks

You can preprocess the data using files in `cs6220hw5.py`, which you can download here. This will remove *stop words*, punctuation, and ensure that the texts are all lowercase. Caution that this implementation is exceedingly slow and un-optimized. (Extra credit to anyone who implements a faster version.)

Featurize your inputs to the algorithm with `CountVectorizer()`. Make sure to set the maximum vocabulary size / number of features, `max_features`.

Submission Instructions

Submit your work to Gradescope. There, you will need to upload a PDF file with the written answers (including the plots) to all the questions above. As well, there will be a link to upload your Python code. Make sure that the signature matches the above signature as we will check it against other types of data.