Homework 5

You can submit in groups of 2. Due 4/16, 1pm.

All assignments need to be submitted via github classroom:
https://classroom.github.com/g/3cBbhmej
and via gradescope.

In this homework, we solve an imbalanced text classification task.
You can download the dataset here:
https://www.dropbox.com/s/6ot9w3on66gp129/hw5_data_test.csv?dl=1
https://www.dropbox.com/s/71q00c3r3vmsz9j/hw5_data_train.csv?dl=1

The dataset consists of Women's fashion online shop reviews, consisting of a title, a review text, and whether the review author would recommend the product. We are trying to determine whether a reviewer will recommend a product or not based on review title and review.
In a real application this might allow us to find out what is good or bad about certain products or to feature more typical reviews (like a very critical and a very positive one).

For these tasks, using the memory option of the Pipeline object might decrease runtimes for grid-searches!

For all tasks, use cross-validation to evaluate the results. Use a metric that's appropriate for imbalanced classification (AUC or average precision for example), and inspect all models by visualizing the coefficients.
For each tasks select a single best model, explain your choice, and evaluate this model using the test set.
We will only try out some combination of preprocessing methods and models. While that doesn't guarantee the best possible model, we try to get a reasonable trade-off of results and model development time.

Task 1 Title and Body (30Pts)

We will look at four ways to use the data

1) Use the title only
2) Use the review body only
3) Concatenate the title and review to a single text and analyze that (discarding the information which words were in the title and which in the body)
4) Vectorizing title and review individually and concatenating the vector representations.

Use CountVectorizer with the default settings and train a linear classifier. Visualize the 20 most important features in the linear model. Tune the regularization parameter of the classifier, and visualize the 20 most important features after regularization.
Do this for all 4 settings. Which one works best?
For the simplicity, for the remaining tasks, we will work with option 3), concatenating the texts.

Task 2 Feature Tuning (30Pts)

2.1 Try using TfidfVectorizer instead of CountVectorizer. Does it change the score? Does it change the important coefficients?
2.2 Remember that TfidfVectorizer uses normalization by default. Does using a Normalizer with CountVectorizer change the outcome?
2.3 Try using stop-word. Do the standard English stop-words help? Why / why not?
2.4 Limit the vocabulary using min_df or max_df. How to these impact the number of features, and how do they impact the scores?

Task 3 n-grams (30Pts)

3.1 Using your current best model, try changing from unigrams to n-grams of varying length. What provides the best performance? Visualize the coefficients. Try visualizing only the higher-order n-grams that are important.
3.2 Try using character n-grams. Visualize the coefficients. Can we learn something from this?
3.3 Investigate how min_df and the use of stop-words changes the number of features when using word n-grams, and how they change the score.

Task 4 Model Tuning (again) 10 Pts

Revisit your choice of model. Compare different linear models with L1 and L2 penalty on the best performing features from Task 3.
Are there any other obvious features to try, or combinations to try out? (Don’t perform them, just list them).