NLP 220 Assignment 1

University of California, Santa Cruz

Part A: Feature Engineering for Naive Bayes, SVM & Decision Tree (23 pt.)

In this first problem, you’ll implement features which can be used with the Naive Bayes, Support Vector Machine and Decision Tree classifiers implemented in scikit-learn.

Dataset

We’ve prepared a subset of the [Amazon Book ReviewsLinks to an external site.](https://www.kaggle.com/datasets/mohamedbakhet/amazon-books-reviews) dataset, and the subset is available  in the file **small\_books\_rating.csv**.

Dataset Format

The dataset contains reviews of books from Amazon:

review/text: holds the content of the review

review/score: holds a score (1-5 stars)

Title: refers to the title of the book

review/summary: holds a summary or title of the review

Other columns may provide useful information for learning to predict scores.

**You are required to build three distinct features which process review/text**, but may build additional features using the other columns, or apply your features to column combinations (e.g. your three features could instead process the concatenation of review/summary and review/text).

Dataset Tasks & Deliverables (5 pt.)

1. From this dataset, create a binary classification dataset: consider any review score of 4 or higher to be positive, and any score of 2 or less to be negative. Ignore scores of 3.
2. Create a train/test split using the train\_test\_split method from sklearn.model\_selection, and having it shuffle the data. Use a fixed random seed of your choice for reproducibility. Use 85% of the data for training and the remaining 15% for test.
3. Print or plot the distribution of your classes/labels, and include these in the report.

Feature Engineering

For this binary classification problem, apply 3 feature engineering techniques of your choice to a Naive Bayes, SVM and Decision Tree classifier. Use the implementations of each included in scikit-learn. In total, you’ll create 3 distinct Naive Bayes, 3 distinct SVM and 3 distinct Decision Tree classifier models in this section.

Feature Engineering Deliverables (12pt)

1. Briefly (1-2 sentences each) describe the features you created.
2. Compare the accuracy and F1 scores across your 9 models on the test data. Compute macro-average F1 for each case. Print confusion matrix.
3. Discuss any features/models that outperformed others. If some feature(s) out-performed others in one or both models, consider/discuss why, if there is a particular reason you might expect this. There may not be an immediately clear reason, there is no requirement to analyze further here or provide a reason if doing so would take significant analysis of the data/models.

**Training/Inference time comparison (6)**

Compare training and prediction/inference time of each of your classifiers. Which one would you recommend based on accuracy and train/inference time?

Part B: Sentiment Analysis on Stanford’s total movie review corpus (15 pts)

Dataset

Download and use the dataset here: [https://ai.stanford.edu/~amaas/data/sentiment/ Links to an external site.](https://ai.stanford.edu/~amaas/data/sentiment/)

This is also available in data subfolder of Files section (aclmdb\_v1.tar.gz)

The dataset includes a README, read this to understand the format, and location of train and test data. You’ll need to design your own method for parsing this to a dataset you can use. I recommend doing this once, and creating a CSV or similar type which you can re-use in remaining experiments.

Tasks & Deliverables (15 pt)

1. Use ngram for feature extraction (n, whether to use stopwords, other design decisions up to you)
2. The training dataset contains 25,000 total examples and is balanced. Further split this using train\_test\_split to create a training split (90%) and a held out validation split (10%), with a fixed random seed for reproducibility.
3. Briefly describe the models you used, and the ngram features you created.
4. Use 5 models to classify sentiment, and report accuracy on the validation split for each. You can use this validation split to help choose hyper parameters for each of the 5 models (you only need to report accuracy of best of each of the 5 model types).
5. Report the test set accuracy of the best of your 5 models. (The test set being the original held out 25,000).

**Part C (**17 pt**)**

Repeat part A with your "own" naive bayes classifier. You need to implement your own naive bayes classifier and then do the steps asked in part A. Compare the accuracy of your own naive bayes with sklearn's naive bayes.