November 18, 2021
Elliott Chen
COMP 4448

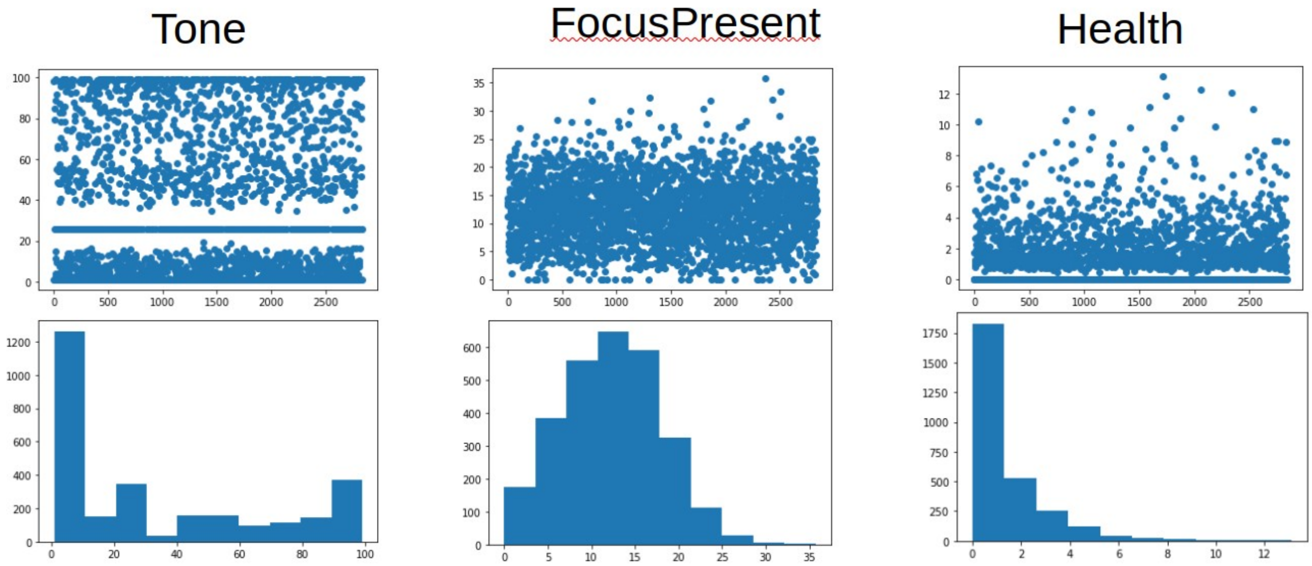
Stress Classification, Linguistics, and Social Media

This paper covers a machine learning project that compares and evaluates classification model performances on an online stress dataset. The dataset was sourced from Kaggle but was created by Elsbeth Turcan and Kathleen McKeown of Columbia University's Department of Computer Science in their 2019 paper, *Dreaddit: A Reddit Dataset for Stress Analysis in Social Media*. The dataset is pre-partitioned into a training set of 2,838 instances and a testing set of 715 instances; each instance is a collection of 116 columns. The dataset focuses on personal posts on the popular content agglomeration website and forum, Reddit, and whether the post authors were experiencing unhealthy levels of stress.

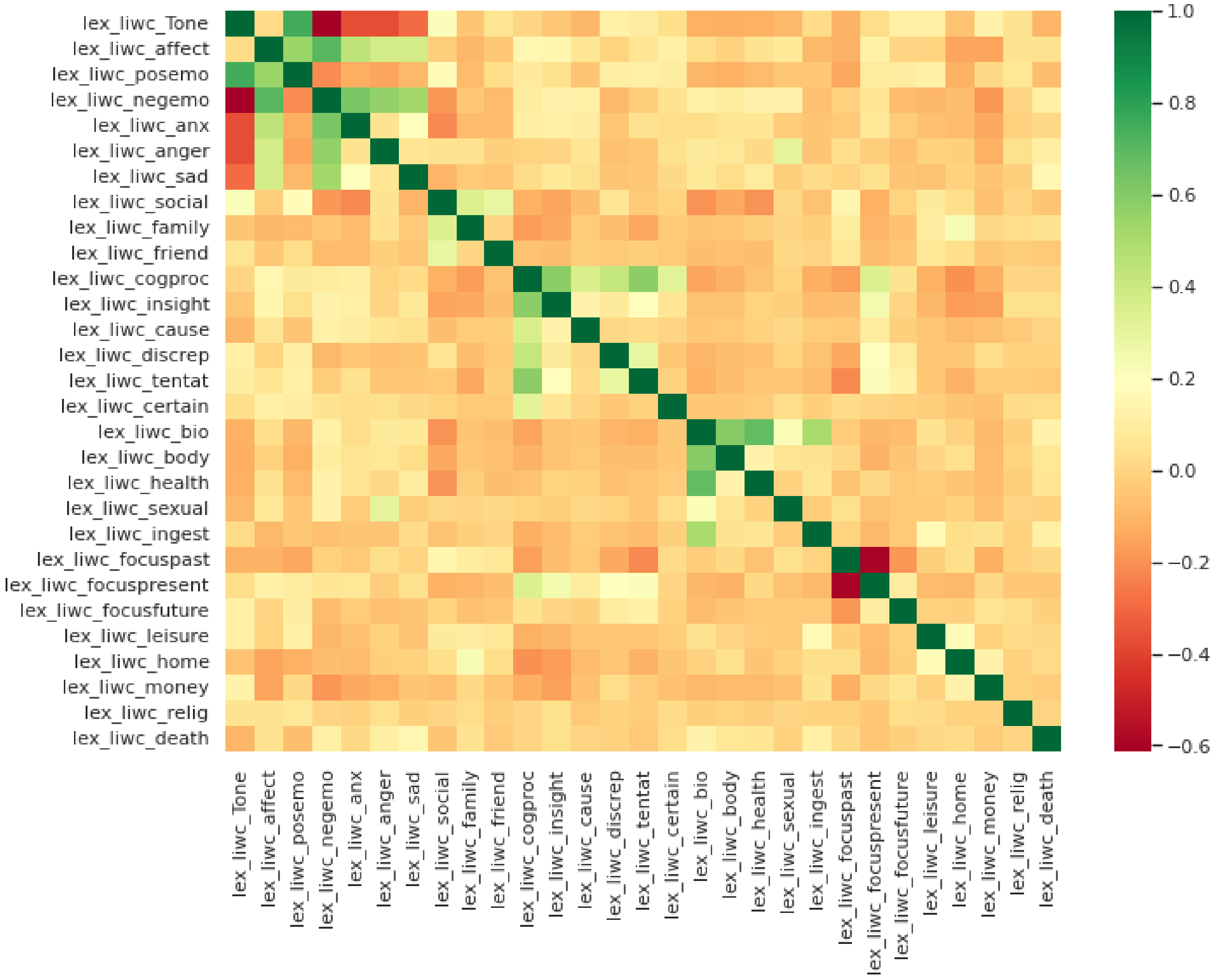
The outcome variable is a binary label, with 1s and 0s describing whether a given post's author was stressed or not, respectively. The other 115 columns are a diverse set of features. Some are Reddit metadata, such as post ID and other site specific information such as social karma and number of upvotes. These are utterly irrelevant to our project and can be safely discarded. One feature is the raw text content of the post, and will be useful. The vast majority of the rest are columns generated by Turcan and McKeown using Linguistic Inquiry and Wordcount (LIWC), a linguistics software tool. These features mostly track information that is not relevant to our analysis—for example, some include words per sentence, total word count, and the number of a given part of speech such as pronoun or verb—but some of them also track the proportion of words in a post that relate to certain themes. These themes include subjects such as anger, health, leisure, and family; we therefore are interested in them as parts of the authors' emotional lives and therefore could play into whether they are stressed or not. These columns are percentages ranging from 0 to 100.

With this in mind, we can employ various different types of classification algorithms with this dataset and compare them. We use and evaluate a logistic regression classifier, a decision tree classifier, a random forest classifier, and a Naive Bayesian classifier that is fed with a TF-IDF vectorizer. The goal is to identify which classifier performs the best at the task of predicting whether the author of a given Reddit post is experiencing unhealthy levels of stress.

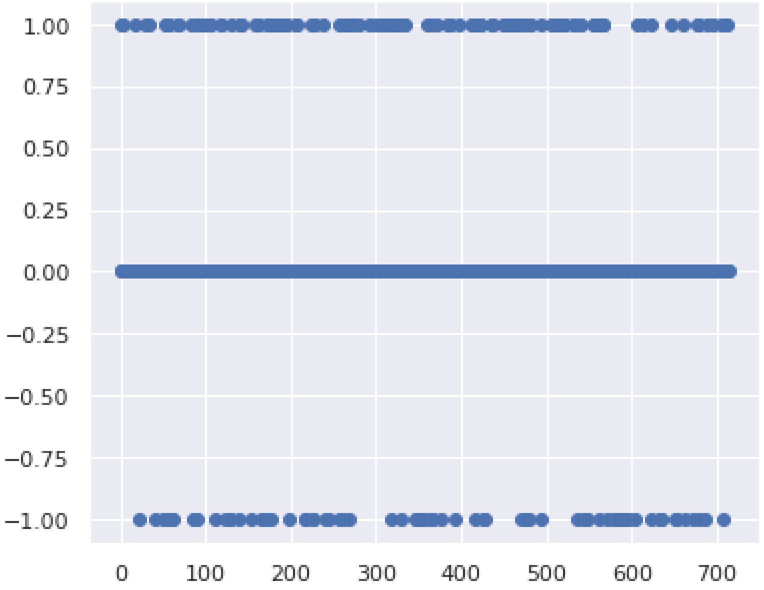
Our proposed Naive Bayesian classifier only requires the raw text feature of our column and involves a separate data preprocessing workflow, so we deal with it separately. Of our other three algorithms, only logistic regression requires serious preprocessing once we make sure there are no missing values. After discarding our unnecessary columns, we find that we have 29 continuous features, all of which were generated by LIWC. There are too many to describe each one in detail here. However, each ranged from 0 to 100 as noted above and, as the following representative scatterplots and histograms show, are typically not normally distributed and do not have large outlying values:



Therefore, we conclude that the min-max scaler imported from sci-kit learn is our best choice to scale the features. We then check for multicollinearity in our dataset. Initially we do visually so with a seaborn heatmap, but given the size of our dataset, we would get more empirical results with another tool. We employ statsmodels' variance inflation factor function and filter out the features that score higher than 7. Taken together with the heatmap shown on the next page, the results indicate that lex_liwc_tone, lex_liwc_posemo (positive emotion), and lex_liwc_negemo (negative emotion) are strongly correlated with each other, and lex_liwc_bio and lex_liwc_health are strongly correlated with each other. That these features might correlate with each other makes intuitive sense; we drop lex_liwc_posemo, lex_liwc_negemo, and lex_liwc_bio. This leaves us with 26 continuous columns that range from 0 to 1 with most arithmetic means less than 0.1 after scaling. As mentioned previously, our data was delivered pre-partitioned, so we are ready to train our models.



We use grid search cross validation to find the best hyperparameters for our decision tree based algorithms. We settle on a max_depth of 3 and max_features of 0.6 for our decision tree classifier, and a max_depth of 20, max_features of 0.2, and n_estimators of 50 for our ensemble learning random forest classifier. The results we get are clear. Both the decision tree and random forest classifiers suffer from extensive overfitting; in particular, the random forest classifier achieved a 99.9% accuracy rate on its training set but only a 52.7% accuracy rate on the testing set—a difference from near perfection to little better than guesswork. The logistic regressor, in comparison, achieved a 73.6% accuracy on the training set and 74.3% accuracy on the testing set. It clearly defeated the decision tree based algorithms. We inspect the residuals from our logistic regressor to make sure that they are independent of X, and are gratified to see that this is the case.



Moving on to Naive Bayes, the only preprocessing we need to perform is cleaning the text data of English stopwords and ignoring instances where the text sample is insufficiently large. Once that is done, we again use grid search cross validation to find the optimal hyperparameters for our TF-IDF vectorizer. Once that is done and we train a Naive Bayesian model, we find some degree of overfitting—accuracy dropped from 86.8% to 70.6% as we switched from training to testing—but we still achieved an accuracy level comparable to that of the logistic regressor with our test set. Furthermore, when we stop to consider other metrics, our Naive Bayesian model looks even better. Considering the relative cost of false positives vs false negatives in a situation where intervention may be desirable, we conclude that recall is an important statistic to consider. Our logistic regressor achieves a recall of 69.6%, compared to the Bayesian model's 86.2%. Depending on how much one cares about the false negative rate, either the logistic regressor and the Naive Bayesian classifier perform very similarly.

Bibliography

https://www.researchgate.net/publication/246699633_Linguistic_inquiry_and_word_count_LIWC

https://www.kaggle.com/ruchi798/stress-analysis-in-social-media

https://aclanthology.org/D19-6213/

http://liwc.wpengine.com/interpreting-liwc-output/