DATA641 Final Project

Connor Laughland, Jaehyun Yoon, Stratis Aloimonos

May 17, 2021

1 Individual Code

Connor Laughland: **Click Here**
Stratis Aloimonos: **Click Here**
Jaehyun Yoon: **Click Here**

2 Introduction

The point of the experiment was to use written data, taken from posts made on Twitter by 250 different people, to identify whether these people were considered neurotic. The data came from a dataset called myPersonality. This dataset includes the user ID's of people who post, the posts themselves, the numeric (numbers between 1 and 5) and nominal (numbers converted to yes/no) measures of Big-5 personality traits (OCEAN: openness, conscientiousness, extraversion, agreeableness and neuroticism), as well as the date the post was made, among other statistics about the posts.

3 Initial Data Analysis

3.1 Pre-Processing

For some classifers (the logistic regressor, the svm with a linear kernel and the svm with an RBF kernel) the AUTHID, posts, and cNEU statuses were used as the data. These were downloaded to their own separate files, and then used to train a model to predict neuroticism based on the posts.

One way this was done was by looking at each post by itself, and if it was used in the training set, its label was its cNEU status. We also collected the posts from each user into one longer post, which was associated with the label of the user (the AUTHID). Since these users were all either neurotic or not, all the posts from each user had the same cNEU status, so there was no problem associating each longer post with the cNEU status of the posts which combined to make it, they all had the same cNEU status.

For the BERT model, the posts were truncated into one post for each unique user ID, associated with the cNEU status of this user. A model and tokenizer were loaded and trained on 90 percent of the posts and cNEU labels, and this model was tested on the remaining 10 percent of the posts, using a unique function. It was also tested on the Essays dataset.

3.2 Stopwords

For the logistic regressor, the svm with a linear kernel and the svm with an RBF kernel, the stopwords were not removed. The reason for this was that they could have been important in predicting neuroticism, as well as the fact that capitalization and use of characters other than words, such as punctuation which could combine to make emojis, could have been more important in determining neuroticism.

4 Baseline Classifier

We now turn to the problem of building a baseline binary classifier to label individual users using our personality dataset. Our default setting is that we apply the logistic regression model using unigrams and bigrams as we got from the previous analysis. We split our data set for our first approach using a single train-test split using a test size of 0.3 to hold out 30% of our data as test-set and the random state of 42. We also use a K-Fold cross validation and Stratified object to split the dataset. The number of folds that we choose is 5 and 10 which there are no different results of the accuracy between the two numbers. The random state remains the same as 42. As the result of splitting the dataset, we have the result shown below.

Training set label counts: Counter({'n': 106, 'y': 69})
Test set label counts: Counter({'n': 45, 'y': 30})

For our baseline classifier, we use a frequency-based word expression method that does not consider the order of appearance of words which is "Bag of words (BOW)" as the feature extraction. We import and use sci-kit learn's CountVectorizer class to create a BOW. For the parameters of CountVectorizer class, we were provided the list of stop words and features made of word n-gram.

We started by building a few classifiers, a logistic regressor, an svm with a linear kernel and an svm with an RBF kernel. These classifiers were trained on the posts given. The parameters these classifiers used were then adjusted, specifically the sizes of the training sets, the number of folds used for validation, and whether or not stratification was done (i.e. training set adjusted to have roughly the same proportions of each label in the set). These classifiers were trained on either The parameters used, as well as whether the set used all the posts, or one line of posts per user, are shown below, along with the accuracy found when testing the classifier with said parameters, are shown below.

4.1 Separating Vs. Combining User Posts

When the posts were combined, the accuracy was lower than for when they were not combined. However, this may have been because the training and testing datasets were significantly larger when the posts were not combined. It is also worth noting that the differences between accuracies for non-combined posts and those for combined posts were much smaller for the svm with an RBF kernel than for the svm with a linear kernel and the logistic regressor, suggesting that the svm with an RBF kernel is a better classifier. The accuracy for the svm with an RBF kernel is noticeably higher than that for the svm with a linear kernel and the logistic regressor when the posts are not combined into one post per user.We visualize our result into three subplots of different number of folds with the two lines of the stratificated data and non-stratificated data.

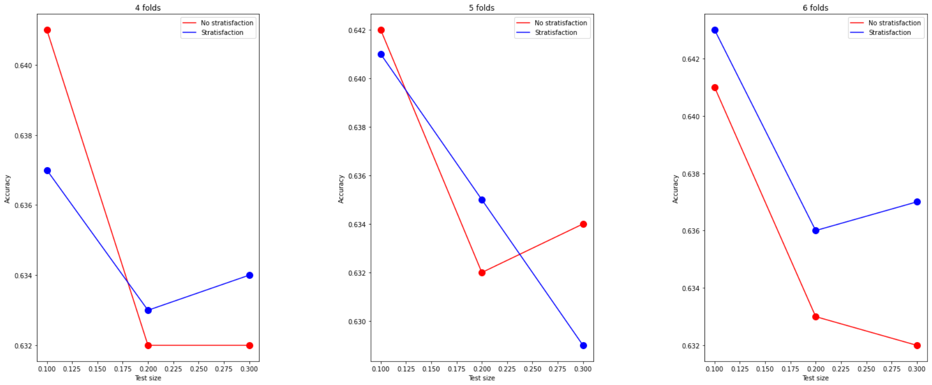

Figure 1: STATS of Logistic Regressor - Not Adjusted to 1 row per ID

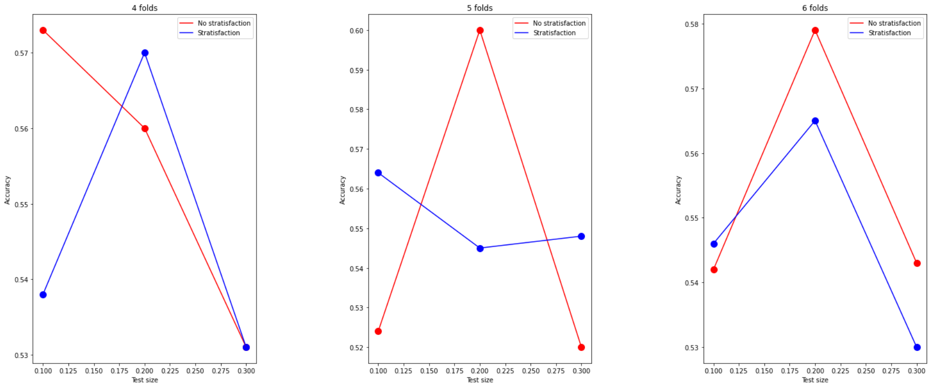

Figure 2: STATS of Logistic Regressor - Adjusted to 1 row per ID

Figure 1 and Figure 2 are the graphs that we take the approach of the logistic regression model for our baseline classifier. Figure 1 is that we treat each as a separate item and Figure 2 is that we collect all an individual's posts into a single long document. We have higher accuracies when we treat each post as an individual item than collecting them into one feature.

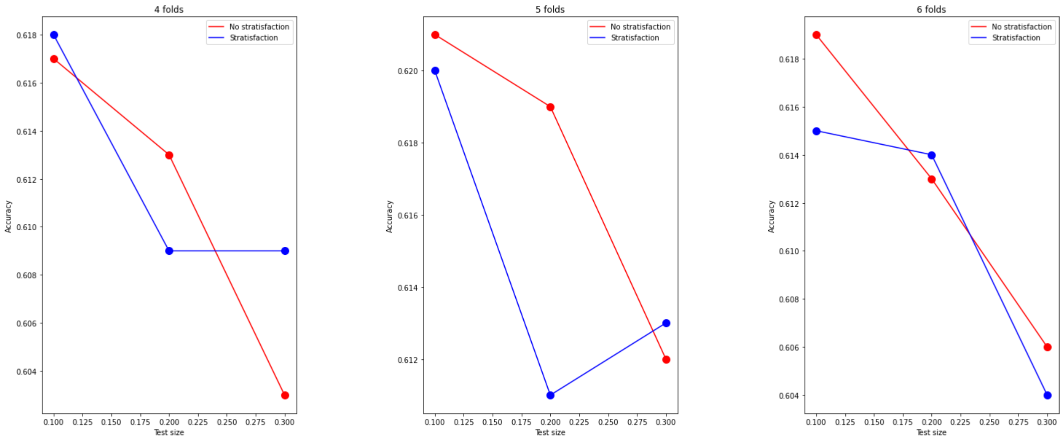

Figure 3: STATS of Linear SVM- Not Adjusted to 1 row per ID

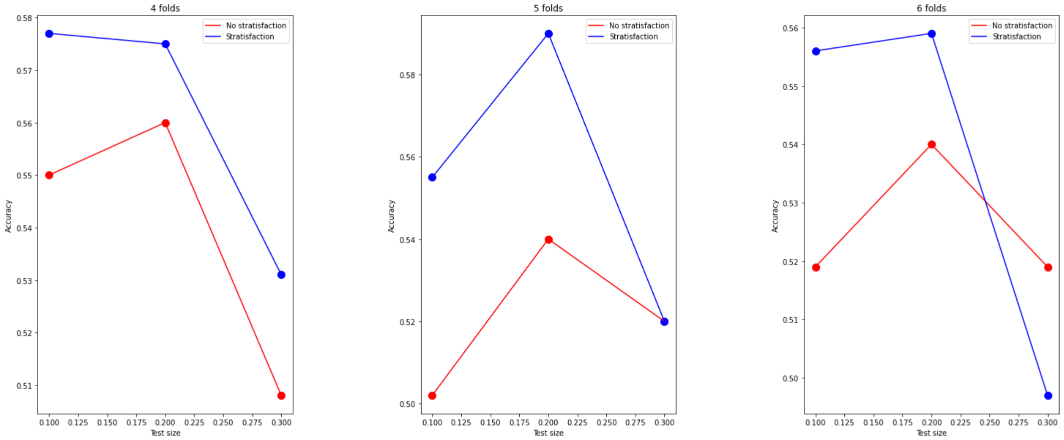

Figure 4: STATS of Linear SVM - Adjusted to 1 row per ID

Figure 3 and Figure 4 is the graphs that we apply the linear support vector machine (SVM) model. Figure 3 is that we treat each as a separate item and

Figure 4 is that we collect all an individual's posts into a single long document. We also have higher accuracies when we treat each post as an individual item then collecting them into one feature overall.

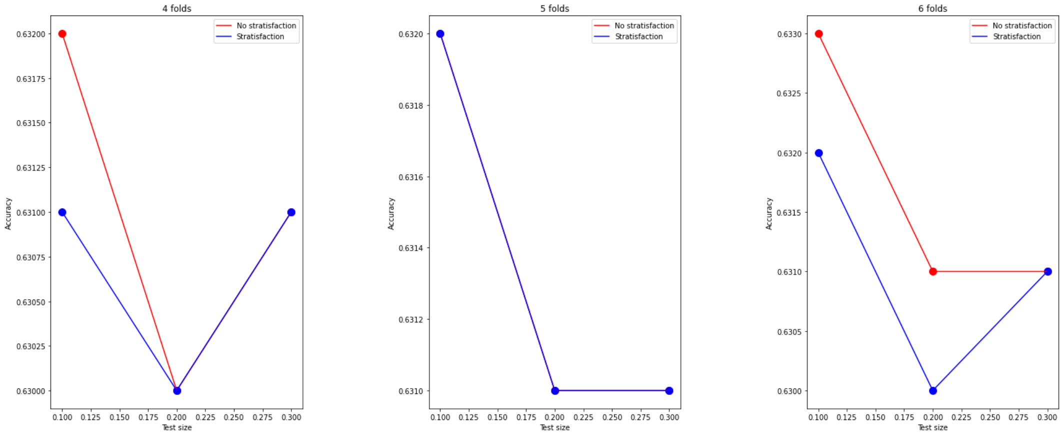

Figure 5: STATS of RBF Kernel SVM- Not Adjusted to 1 row per ID*

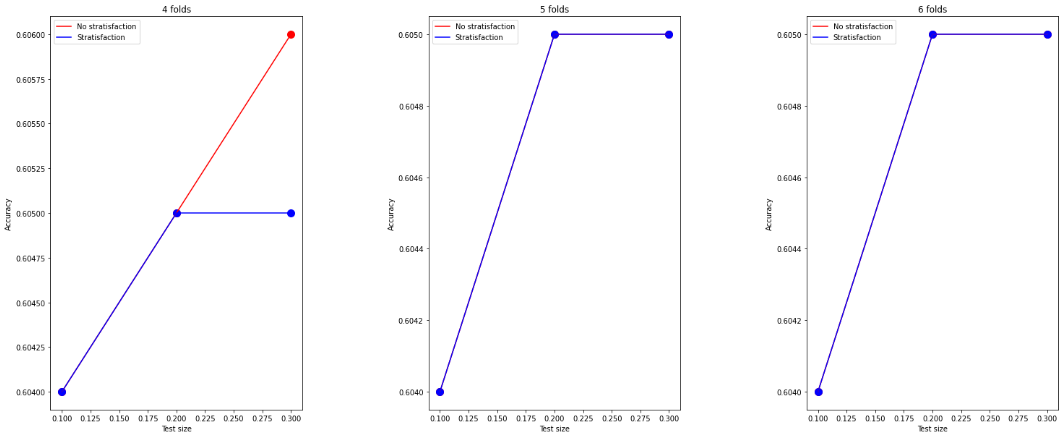

Figure 6: STATS of RBF Kernel SVM - Adjusted to 1 row per ID*

*Two lines of the stratificated data and non-stratificated data are shown overlapped

Figure 5 and Figure 6 are the graphs that we apply the Radial basis function (RBF) Kernel support vector machine (SVM) model. Figure 5 is that we treat each as a separate item and Figure 6 is that we collect all an individual's posts into a single long document. For this model, we have higher accuracies on collecting documents into one feature than considering them as an individual item overall.

5 Improving Classification

5.1 Classifiers and Feature Detection Methods

Using scikit learn, we were able to compare the baseline logistic regression classifier to a few of many other classifiers available to us. The two other classifiers represented in the graphs below are "SVM" (using a linear kernel), and "Decision Tree". For the SVM classifier, we could have tested all of the kernels against different feature detection methods. However, the classification results for each kernel didn't vary enough to warrant including each on their own.

For the feature detection methods, we used "Bag of Words" (BOW), "Term Frequency–Inverse Document Frequency" (TF-IDF) model, "Linguistic Inquiry and Word Count" (LIWC) model, and "Empath". All of these text representations are described below, as they inform the specific features detected by the classifiers.

Constant Parameters: For each of these trials, there were a few constant parameters. The posts were consolidated into a single document per user. For BOW and TF-IDF, the ranges remained consistent between unigrams and trigrams. For both LIWC and Empath, the feature range wasn't limited, and included all of the topic specific feature categories. For TF-IDF, the DF range was between 10 and 0.1.

While the Logistic Regression and SVM linear classifiers performed similarly to one another, Decision Tree performed decidedly worse. What we can glean from this is that whatever features are unique to each class can be separate linearly. If they weren't linearly separable, logistic regression would have performed significantly worse than SVM, which projects data into higher dimensions to find a linear division that doesn't exist in the existing dimensional space.

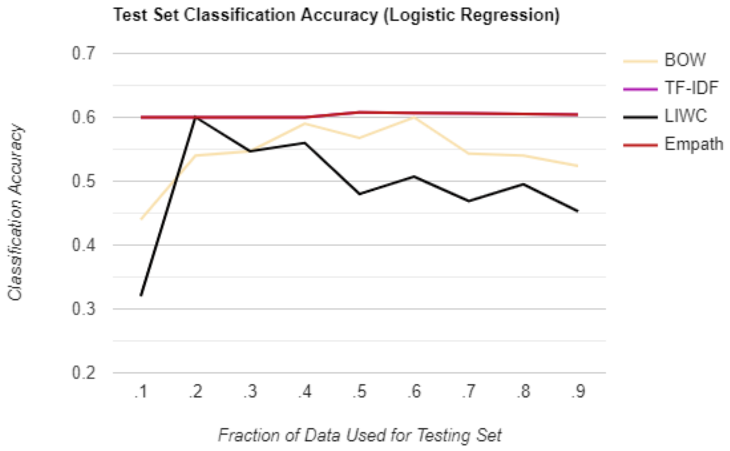

Figure 7: The classification accuracy values of different feature detection methods, using the logistic regression classifier. Note: For this classifier, TF-IDF and Empath values are identical

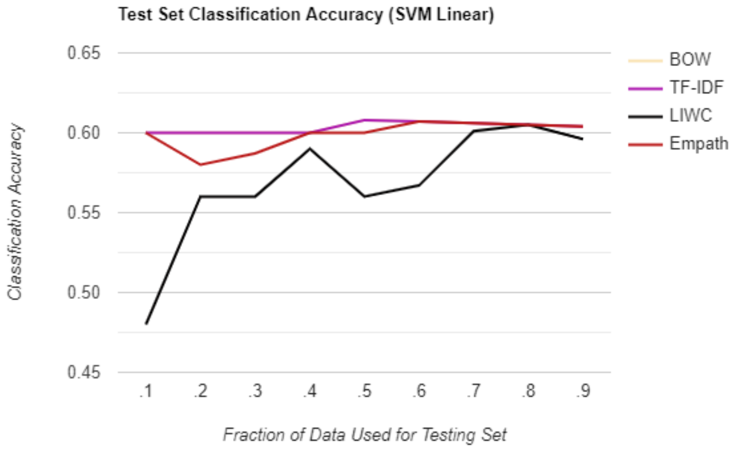

Figure 8: The classification accuracy values of different feature detection methods, using the SVM (linear kernel) classifier. Note: For this classifier, TF-IDF and BOW values are identical

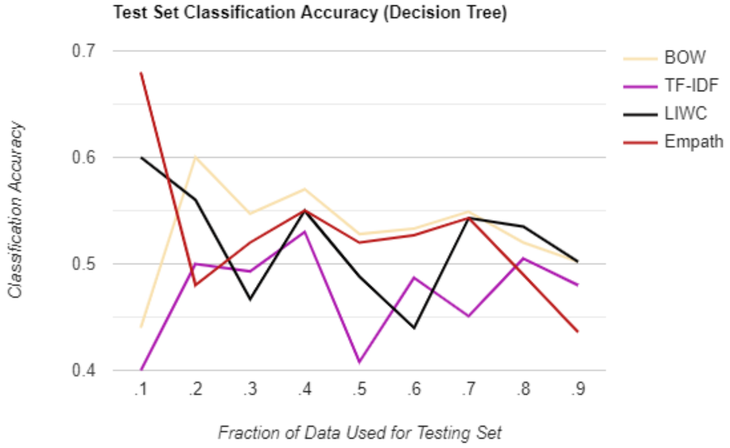

Figure 9: The classification accuracy values of different feature detection methods, using the Decision Tree classifier.

5.2 Most Informative Features

To be clear, the most informative features in *Table 1* are pulled from all of the documents combined. This means that while they are highly associated with a certain classification label, they aren't necessarily found often. Given how exploratory this classification experiment was, there was no guarantee that any of the documents document would have strong positive/negative indicators of neuroticism. Additionally, these features are informed by the parameters used. In the following case, a .1 test/train split was used for all of the methods.

Note: Feature extraction could have also been conducted for the SVM classifier, although the original script would have required fine tuning. This is because the features would need to take into account the fact that the kernel is creating many different binary classifications between features, in effect creating a nonlinear decision boundary between the classes. Given that the SVM classifier didn't significantly improve the classification results, we can trust that the most informative features are similar to those for logistic regression.

BOW: The range of ngrams used for this analysis was 1 to 3, as shown by the maximum length of the features in *Table 1*. However, any range could have been used. BOW has the benefit of easy implementation, where the feature vector of each document is easily constructed based off of its individual word frequencies. While this method can become computationally taxing for larger corpa, this one was small enough that the feature vectors could easily be processed by the classifiers we used. The results, however, to a human eye, aren't very intelligible. Because this classification experiment is exploratory, we don't have any particular tokens in mind for defining neuroticism. So, the ngrams that are found to be most/least associated with neuroticism aren't easy to understand out of context. Because BOW doesn't consider a words position in the sentence, or its syntax, the context of the sentence is lost.

> The features lack a clear narrative, relative to the other feature detection methods. By this, I mean that the words don't strike one as strongly/weakly associated with neuroticism. Many of the neurotic features are associated with phones. Without any deeper analysis of the documents, one could speculate that it could be associated with a phone recently being released, or some kind of public drama. But drawing a clear conclusion would require more inspection of the individual documents.

TF-IDF: This method is similar to BOW, in that it rates the frequency of each word in every document. However, it differs in that it adjusts the frequency according to whether they're unique to that individual document or are found frequently among all of the documents. The TF–IDF value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. As figure 2 shows, TF-IDF significantly improves the classification accuracy. In BOW, less common ngrams are assigned very low frequency values, and can easily be ignored by the classifier, despite the fact that they might distinguish neuroticism better than more frequent words. TF-IDF, however, weights those unique words with higher frequencies, giving them more influence on the linear classification. The scores for the TF-IDF ngrams are much higher than for BOW, meaning that stronger associations were found. This meant that more specific contexts surfaced, as discussed below.

> Neurotic: The most associated ngram was a text short-hand for an emoji, in the form of ”xd”. Other top ngrams consisted of ”propname”, which indicates that the words were rendered null during the tokenizing process. This could be because there were too many misspellings, or that their were random characters in a row. There were also some expletives, including ”fuck” and ”fucking”. These are words often associated with rage and negative emotions. There are a lot of words on this list that require more context, though, like ”frog”, and ”black”.
>
> Not Neurotic: Because words that were both infrequent and highly associated rose to the top of these feature detection results, certain specific positive contexts can be accounted for. These include ngrams like ”birthday wishes”, ”vegas”, ”beach”, ”birthday”, and

> ”packing”. These are all words associated with events and travel, which are usually associated with positive emotions.

LIWC: This representation tokenizes the text in each document, and then determines the number of unigrams in the document that are associated with each sentiment/topic category in the LIWC dictionary. The dictionary associates a group of words with a sentiment, where any word can be included in multiple categories. This set of instances becomes a document feature vector that can be compared to all of the other documents. Note that there is a bigram model for LIWC, but we chose not to try implementing it. Luckily, the most informative features are much easier for a human to interpret, as categories provide more context than individual ngrams. This method performed poorly for every classifier at first, and required optimization (explored in the section below).

> Neurotic: ”Nonfl” is the most highly associated feature. This refers to non-fluent strings like ”uh”, ”rr”, etc. This could be referring to strings like ”uh-oh” and ”uhhhh”, which are often used in reference to distressing topics, but it would require looking at the particular documents. Other features include ”see” (seeing verbs), ”family”, and ”anger”. Topics surrounding family and anger can be particularly stressful, and elicit strong emotions from people.
>
> Not Neurotic: ”Future” is the most highly associated feature. This is an interesting one, as ”past” is associated with neuroticism. At face value, one can speculate that talking about the future is more strongly associated with hope and joy, whereas talking about the past can often be associated with ruminating and dwelling on uncomfortable events. Again, these are only face value interpretations. Beyond that one, however, the other categories reflect how poorly it performs (using a .1 test/train split) during classification. Sadness, positive emotions, and negative emotions are all associated with not being neurotic, which doesn’t track with the other feature detection methods.

Empath: It’s very similar to LIWC in that each document is reduced to a feature vector describing the number of instances associated with a dictionary of lexical categories. While similar to LIWC, the classification results were far superior. For all three classifiers, it was either the best performing, or the second best (ties included).

> Neurotic: ”Negative emotion” is the most highly associated feature. This is what we would expect, as negative emotions are an integral feature of neuroticism. It evokes distress and discomfort. The other features are also easy to interpret, including ”swearing terms”, ”night”, ”social media”, etc. All of these terms, at face value, are regularly associated with negative, neurotic episodes. Social media elicits a lot of anxiety and rumination from people. And the

night is the best time for people to be left alone with their thoughts. Swearing terms hearken back to the expletives detected by the BOW method. The features are all, in one way or another, associated with anxiety.

Not Neurotic: ”Positive emotion” is the most highly associated feature. Again, it checks out. The other top features include ”celebration”, ”childish”, ”giving”, ”friends”,, ”love” etc. These all elicit feelings of joy and togetherness. People who feel socially connected are found to be happier and healthier, whereas loneliness can breed anxiety and depression. The terms are all strongly associated with activities, family and friends.

BOW Feature Detection

Score	Non-Neurotic Feature	Neurotic Feature	Score
-0.199	work	black	0.1366
-0.1225	internet	phone iphone	0.0828
-0.119	doesn	iphone	0.0828
-0.1084	writing	phone	0.0757
-0.1043	home	yesterday	0.0639
-0.1021	eat	propname propname	0.0595
-0.1	likes	smoking	0.0501
-0.0988	car	christmas	0.0495
-0.0975	inch	snow snow	0.0472
-0.0966	ice	dreaming	0.0472
-0.0935	dance	today day	0.0431
-0.0929	car encased	upgrade broke phone	0.0414
-0.0929	car encased inch	upgrade broke	0.0414
-0.0929	dance skirts	surrounded naysayers phone	0.0414
-0.0929	doesn work	surrounded naysayers	0.0414
-0.0929	doesn work car	phone iphone os	0.0414
-0.0929	dog	os upgrade broke	0.0414
-0.0929	dog eat	os upgrade	0.0414
-0.0929	dog eat expensive	os	0.0414
-0.0929	eat expensive	naysayers phone iphone	0.0414

TF-IDF Feature Detection

Score	Non-Neurotic Feature	Neurotic Feature	Score
-0.6917	birthday wishes	xd	0.6218
-0.521	lol	propname propname	0.6173
-0.5195	vegas	hates	0.5135
-0.4029	beach	propname propname propname	0.4933
-0.3952	red	kitty	0.4827
-0.3833	le	ppl	0.4253
-0.3817	interview	rock	0.4134
-0.3596	ticket	fucking	0.4035
-0.3484	birthday	black	0.3994
-0.3479	packing	2010	0.3992
-0.3423	pictures	passed	0.3829
-0.3306	chicago	fuck	0.3817
-0.3296	feelin	da	0.3763
-0.3109	side	mode	0.3705
-0.3052	check	frog	0.3668
-0.295	chinese	law	0.3484
-0.2939	loves	til	0.3441
-0.2936	wishes	studying hard	0.3181
-0.2922	happy birthday	en	0.3155
-0.2863	winter	bar	0.3148

LIWC Feature Detection

Score	Non-Neurotic Feature	Neurotic Feature	Score
-0.2938	future	nonfl	0.3236
-0.2206	feel	see	0.3194
-0.2193	sad	family	0.2822
-0.2086	cause	anger	0.2557
-0.1946	home	auxverb	0.2131
-0.1477	leisure	inhib	0.1711
-0.1474	verb	number	0.15
-0.1414	tentat	past	0.1366
-0.14	they	excl	0.1232
-0.1035	posemo	assent	0.1058
-0.0986	negemo	i	0.1051
-0.0895	death	cogmech	0.1028
-0.0869	adverb	anx	0.1017
-0.0774	hear	friend	0.0801
-0.0763	ingest	bio	0.0673
-0.0693	swear	ipron	0.0624
-0.0658	percept	article	0.0615
-0.064	we	you	0.0511
-0.0626	money	present	0.0508
-0.062	insight	achieve	0.05

Empath Feature Detection

Score	Non-Neurotic Feature	Neurotic Feature	Score
-0.1586	positive_emotion	negative_emotion	0.1509
-0.1315	celebration	dance	0.0995
-0.1298	childish	swearing_terms	0.0863
-0.1156	giving	night	0.0821
-0.1021	business	social_media	0.0805
-0.1003	friends	communication	0.0801
-0.0862	cheerfulness	real_estate	0.0779
-0.0857	cooking	speaking	0.0715
-0.0856	restaurant	internet	0.0709
-0.0796	love	movement	0.0679
-0.0796	work	domestic_work	0.0632
-0.0756	beach	music	0.0494
-0.0704	eating	sleep	0.0478
-0.0684	sports	violence	0.0437
-0.0682	affection	strength	0.0413
-0.0619	ocean	injury	0.0403
-0.0617	divine	home	0.0378
-0.0614	children	emotional	0.0378
-0.0613	wedding	technology	0.0374
-0.0603	optimism	government	0.0359

Table 1: Top 20 Most Informative Features: For each feature detection method, the most informative features were extracted via the scikit logistic regression classifier (using a .1/.9 test/train split. The further from 0 the score is, the more biased the feature is in favor of that associated classification label (neurotic, or not neurotic).

5.3 Optimizing the LIWC Category Range

The first attempts we made at using LIWC for feature extraction were unsuccessful. However, we didn't want to rule it out completely, given how informative all of the features could be if optimized. We decided to test several different ranges for the features to determine whether the classification accuracy could be improved. In this case, what we mean by range is that only features that had a number of occurrences between a lower and upper bound would be included in the feature vector for that document. Through several successive tests, we found that the range between 400 and 600 occurrences rendered the classification accuracy as high as TF-IDF/Empath for a logistic regression classifier, and as high as TF-IDF/BOW for a SVM linear classifier, as shown by Figures 10 and 11 below.

One initial takeaway might be that a range somewhere in the middle of the LIWC category features might avoid the noise on either end of the range. The lower end of the range might include those features that are so unique to particular documents that they can't reliably classify other documents. And the higher end of the range might include those categories that are so ubiquitous to both document classes that they can't be used for classification. The middle range could be accounting for those that are unique enough, and specific enough to a particular class, that they're optimal for classification.

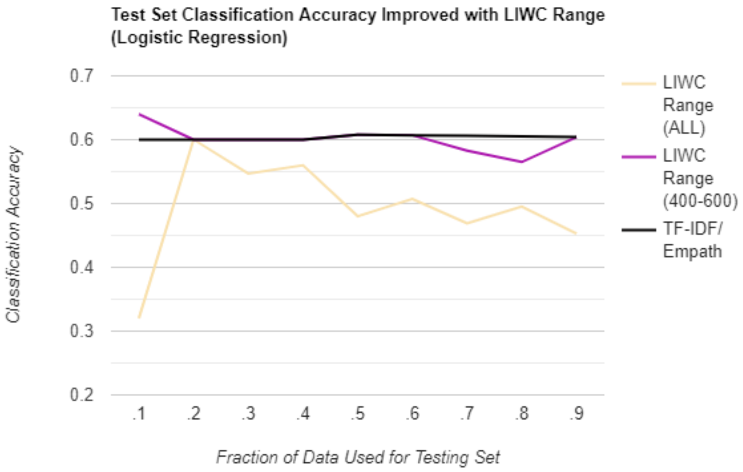

Figure 10: For the logistic regression classifier, comparing the best initial feature detection method/s to an optimized LIWC range trial

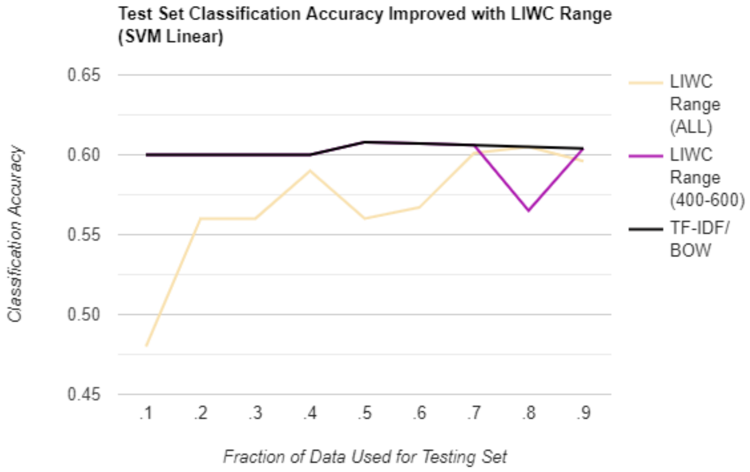

Figure 11: For the SVM liniear classifier, comparing the best initial feature detection method/s to an optimized LIWC range trial

5.4 Neural Networks

Note: We slightly altered the script from the following tutorial link to run the Neural Networks: **Click Here**

Using the Keras package, we implemented both a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN). We tried tuning the parameters of the network architecture, which included its layers, the number of epochs, the filter size/number, and the word embeddings. Despite the tuning, however, the network never surpassed a classification accuracy of more than 60 percent.

5.5 BERT Model

Note: We slightly altered the script from the following tutorial link to run the BERT classifier: **Click Here**

Results for MyPersonality Data:

Metric	**Value**
eval loss	0.685
eval accuracy	0.6
eval runtime	41.217
eval samples per second	0.607
epoch	3

The table above shows that the evaluation (classification) accuracy of the held out validation set was 60 percent, which is consistent with the maximum accuracy of the other neural networks we attempted.

These results occurred when the BERT model was trained on 90 percent of the posts (combined into one long post for each unique user) and tested on the other 10 percent. This model, when tested on the 'Essays' dataset, returned an accuracy of 0.499.

This model, when trained on the combined posts, returns 'Not Neurotic' for almost everything. Ultimately, the accuracy is around the distribution of the testing set.

6 Conclusions

6.1 Classifiers

Based on these experiments, which are simple to understand and perform, one can get an idea of how different classifiers perform for the data. One of the first choices we had was whether to split or combine the user posts. It is worth noting that when the posts are not combined into one post per user, the accuracies of the SVM with a linear kernel, the SVM with an RBF kernel and the logistic regressor are roughly the same, but when they are combined into one post per user, the SVM with an RBF kernel shows a noticeably higher accuracy. With that said, all of the classifiers performed better when the posts were separated, suggesting that the consolidation of their posts might result in posts that are easy to classify being diluted by other posts that are harder to classify. A next step for analysis might be to compare post classifications for users between consolidation and separation, and identify those posts that might be affecting the classification accuracy.

6.2 Feature Detection

The different feature detection methods revealed many word categories and ngrams that are intuitively associated with neuroticism. However, it's clear by our inability to yield a classification accuracy higher than approximately 60 percent (for consolidated posts) that while those features might exist, the post won't be classified correctly if they aren't present. That speaks to the broader issue of social media posts having such wildly different topics, and being so dependent upon current events and subcultures, that personality assessments are difficult to make based off of arbitrary moments in time.

6.3 Limitations

One of the greatest limitations posed by the MyPersonality data was the lack of clear criteria for what kinds of posts warrant the classification of neuroticism. Because of this experiment's exploratory nature, the indicators of neuroticism weren't guaranteed to exist within the data. We had to test to determine whether they exist. So, despite how many combinations of feature vectors and classifiers we paired, we couldn't force an association that didn't exist. This seems to be the explanation for why BERT, which is supposed to be one of the higher end NLP text classification models, could only reach a maximum evaluation accuracy of 60 percent, no better than the other classifiers we used.

One might wonder whether classification would be improved by a larger amount of data for each user, or by each user's data being filtered down to particular kinds of posts that might be better indicators of neuroticism. For example, if posting about something regularly and obsessively is an indicator, you could easily take a large series of posts from an individual user and count how many times, and with what language, they chose to discuss it. Similarly, if emotional instability is an indicator, one could try to track their emotional fluctuations (positive vs. negative emotions) and turn that into a feature metric. The intersection of data analysis and psychology poses a fascinating potential for better identifying self-destructive and life threatening behaviors, independent of the ethical implications (for which there are many!).