Classifying Personality Types with Neural Network Based on Social Media (Twitter) Posts

Stephan Schuller

CS 457 / CS 495 Research Project

Department of Computer Science

Central Washington University

Ellensburg, WA

6/6/2023

Table of Contents

Introduction

The ability to communicate in the internet era means that information is spread quicker and in a greater quantity than ever before. Unfortunately, there is also more suspicion about the validity of information than ever before. Social media platforms have become a central part of the American culture and our ability to tell truth from fiction has become increasingly challenging. The American freedom of speech has encouraged the creation of technology that allow speakers to reach larger audiences and the need for equally matched technology is necessary for consumers to make decisions about the ubiquitous material of social media. Most popular social media platforms have their own algorithms for labeling content as potentially misleading, but as the social media environment evolves and the owners of these media companies change, there will always be a desire for additional third-party watchdogs.

The contribution covered in this research paper is a project, that follows primarily in the footsteps of other papers on the well-trodden ground of categorizing with machine learning. The summary is: if there were method to learn something about the personality or motive of a poster on social media the risk of being captured by controversial rhetoric can be reduced. The subject of this research is the categorization and quantification of personality traits. Utilizing a well-known and verifiable personality trait model the way humans interact with each other can be measured, and when combined with data from popular social media platforms can create tools that allow users to interpret social media content with added context.

Methods

The “classification of personality types using machine learning and social media”, this is not a new topic. A brief exploration of this subject will reveal a robust collection of work exploring this very topic with more intelligence than could be created in this research project. Therefore, this research retraces the proven methods of other researchers in this and adjacent topics. Some methods used in this research were emulated from Vijay Hima and Sebastian Neenu from the School of Engineering & Technology Vidya Nagar, Karukutty, Ernakulam, in their work, “Personality Prediction using Machine Learning, particularly the use of the OCEAN personality trait model for categorization as well as the Open Psychometrics datasets [1]. Additionally, the research of Andrew Dunn, Central Washington University, Ellensburg, Washington, United States was used as well, especially the use of NLTK for preprocessing data to create the Bag-of-Words.[9]

The OCEAN Model

The OCEAN Model was selected as the format and measurement for personality traits for 2 reasons. The first reason is that other Neural Network papers referenced in this research also uses the OCEAN model, meaning that it was possible to make one-to-one comparisons for accuracy, double checking the validity of methods, and to generally operate in a known arena where other research could back up the claims that were being put forward. The second reason is that out of other popular methods for measuring personality traits, the OCEAN model had the most support as a objective, widely accepted and scientific standard for quantifying personality.

Twitter

Twitter was selected as the social media platform for 3 reasons. The first reason is that Twitter was the venue of controversial expression of free speech that had been at the center of discussion in recent American history. The second reason is that Twitter was the focus of other research making it easier to validate methods for effectiveness and accuracy. The third reason is that the largest and most simple dataset that was available that also satisfied all the requirements for training an artificial neural network were available for Twitter.

Python

Python 3.11 programming language was chosen for this research project for several reasons including access to various powerful libraries that make creating an Artificial Neural Network model more accessible. This includes libraries used for preprocessing word data, NLTK. Libraries specifically for creating and training the Neural Network model, SciKit Learn. As well as libraries for accessing and manipulating data frames, Pandas. And libraries for making plots, MatPlotLib.

Datasets

For this research, several different datasets were used for different purposes.

For Model Training

The dataset used for training the MLP model was from the My Personality dataset, created by graduate students at Stanford University. This dataset is no longer supported. However, it is the largest dataset available which has related information categorized by Twitter authors, who had been evaluated by the OCEAN standard psychological trait test. Giving them a rank from 1 to 5 in the categories, “Openness “, “Conscientiousness”, “Extraversion”, “Agreeableness” and “Neurosis”. The summary of this dataset’s usefulness for training is essentially that it provides the means to make a dictionary of words for every author, and that dictionary can be directly correlated to a linear ranking of personality traits. When the dictionaries are combined, they create a scalable library of words (Bag-of-Words) that can be referenced in a binary modality to assign identity and value to words and to test a model on new data based on the weighted presence of those words.

For Quantifying Output

The dataset used as a benchmark for comparing the “normalness” of distributions for personality traits was the IPIP dataset from Open Psychometrics, which includes questions and responses to 50 OCEAN style questions. These self-administered tests allow for a baseline model of the distribution of normal personality traits across the 5 categories. These normal distributions were then available to be compared both to the Twitter base distribution and the model prediction output to determine that all numbers were within a realistic range.

For Confirming Ideas

The 100,000 words dataset was the weighted output of a similar model created by someone else which shows the correlation between various words and their most strong correlated personality trait, the summary of this data was used exclusively for comparing outputs from the model to ensure that more experienced researchers were not being contradicted by my naïve assumptions.

Model

Making the Training Set

The first step was to create the training set. Every training case comprised of all statuses (also called tweets) from a single author. This could be between 20 and several hundred tweets, and every tweet was between 3 and 280 characters. Every author had also been evaluated by an OCEAN personality trait model. With one OCEAN test per author. [1]

The levels of preprocessing of status content are as follows: The first step that was taken was to combine all tweets from a single user and split those sentences into a list of words. Using NLTK to complete the following 3 steps: 1) that list of words had all unknown and unusual characters removed as well as punctuation. 2) The list of words had all stop words removed, which are common words that search engines ignore such as conjunctions personal pronouns, and repeated words including weak forms. 3) the list was sorted by occurrence. This was a way to glean some level of importance from the list, assuming that more frequently or commonly used words of a certain scope and variety would be adequate for categorizing traits.

Each of these test dictionaries were associated with the OCEAN scores for that author. Which is a list of 5 numbers with up to 2 decimal places between 1 and 5. The OCEAN scores are the expected output of the model. These lists were put into an array with each element representing a user's OCEAN trait levels. This would serve as the Y set.

The X set and Y set were split using built in functions of SciKit Learn to create an 80/20 split for training and testing. The training set would be referred to as X_train and Y_train. And the testing set would be referred to as X_test and Y_test.

Making the Model

The basis of our neural network model is a Multi-Layer Perceptron (MLP) regression model. The model was created by using the My Personality dataset.

Testing the Model

The X_test is offered to the model using the fit function, where the model is free. Using weights developed in the training phase, to make predictions on the X_test data. The model creates an array of prediction values called Y_pred which is used for measuring the trained efficiency of the model.

How the Model Works

Starting from the training set of word dictionaries the Bag-of-Words can now be created. There are 5 steps that follow how the training set is converted into a binary word dictionary. 1) The word list will no longer be separated by author. 2.) The resulting set was created from the remaining words meaning all duplicate words are finally removed after being occurrence sorted. This means that the most repeated words are located at the beginning of the list and progresses to least repeated in increasing order, ending the set with words that were only used once in the set. 3) Finally, the dictionary is trimmed to the final dictionary size, choosing to keep "common" words before the trim number and remove "rare" words after the trim number. This list will serve as a word "lookup" dictionary. 4.) A binary array of the same size as the lookup is created and populated with all 0's (zero).

The actual data that is fed into the model for training and predictions will now be a binary list that is compared to the in-author input and cross referenced against the lookup table.

Any word from the author input that matches a word in the lookup table has its same index swapped to a 1 (one) in the binary array. This array is then fed into the model and paired with the 5 OCEAN related scores.

How the lookup table is created and trimmed is shown in Figure 1.

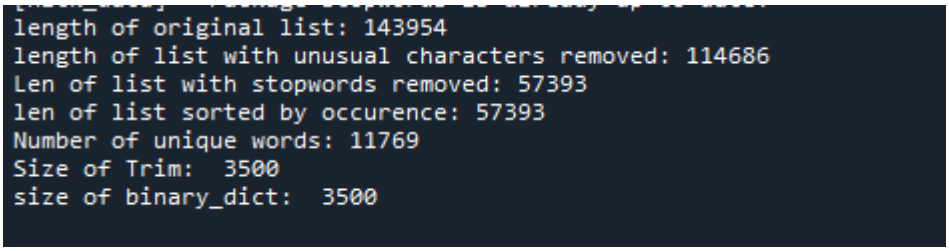

Figure 1: Lookup Table is Created

Measuring Accuracy

This is where the final outcomes are measured. This case a SciKit Learn function to measure Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) by comparing the Y_pred set with the Y_test set. The process of training the model involves the adjustment of several parameters, the four main parameters being the number of hidden layers and the number of neurons in those hidden layers. The size of the word dictionary and the number of iterations that would be performed on the training data set.

RMSE

When dealing with small numbers RMSE can be a misleading metric to measure error as the RMSE can change in an unbound way depending on the actual numbers being compared.

For the creation of the model, RMSE is used for tuning parameters however as a final word an RMSE of one (1) must be taken in context of a range of numbers from 1 to 5. This could mean at the most inaccurate the accuracy would be 50% and the most accurate could be 80% this is quite a large range because the numbers being dealt with are small.

MAPES

Taken from create_model.py. shows the SciKit Learn Methods being used, and the calculations used to get report results. Resulting in output as shown in Figure 2.

```
251
252     # Calculate and print the mean squared error
253     mse = mean_squared_error(y_test, y_pred)
254     rmse = np.sqrt(mse)
255
256     neg_accur = mean_absolute_percentage_error(y_test, y_pred)
257     accur = (1 - neg_accur) * 100
258
```

Figure 2: Two ways of Measuring Accuracy

Issues of Measuring Accuracy

Depending on the type of variance in the prediction. The RMSE can vary significantly and the MAPE will not change very much. As shown in Figure 3.

This adjustment of parameters was done until the RMSE was minimized and the accuracy was maximized. Unfortunately, as described above these measurements are not strictly related. So, adjustments to parameters had to be considered optimizing both of these.

Accuracy measured by MAPES is also unfortunately not a good measure, because at the test the size is quite small (30 cases).

```
RMSE:  0.8966533332341993
Accuracy:  76.55366618441184

RMSE:  0.9763036412919907
Accuracy:  74.89118071995384

RMSE:  0.9851405991024835
Accuracy:  74.45293912135463
```

Figure 3: Comparing RMSE and MAPES

The SciKit Learn documentation confirms the use of the measure will give properly calculated results however those results may have skewed accuracy for small sets, which states:

> "The output can be arbitrarily high when y_test is small (which is specific to the metric) or when abs(y_test - y_pred) is large (which is common for most regression metrics)." [8]

There was an attempt to create the most accurate model using the tactic of adjusting parameters, and after some iteration and comparison while limiting the size of the input of the model by 2 metrics: The first metric to be optimized would the size of the word dictionary. And the second metric to be optimized would be to reduce the number of iterations needed to create an accurate model. as shown in Figure 4.

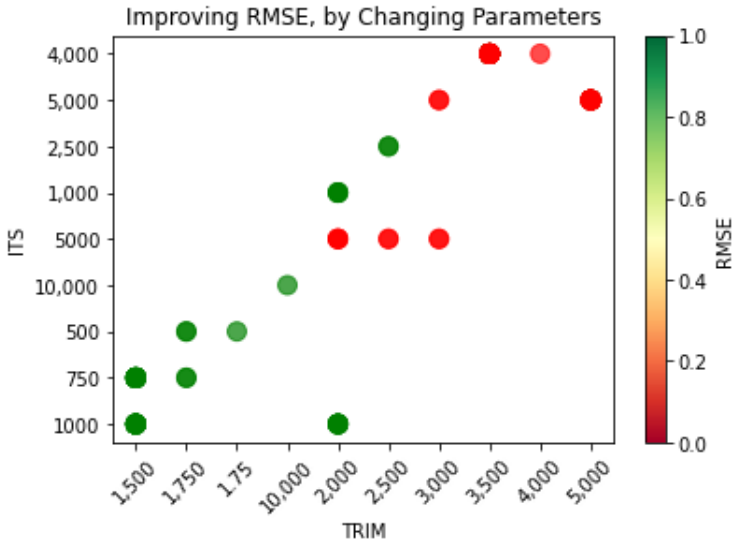

Figure 4: Reducing RMSE by Adjusting Parameters

The Final Parameters of the Model

The model is a Multi-Layer Perceptron model with multi-layer linear output. The model takes a list of words represented in binary when cross referenced to the word dictionary and outputs 5 linear values between 1 and 5 which correspond to OCEAN traits.

The model has 2 hidden layers. The first hidden layer has 250 neurons, and the second hidden layer has 10 neurons. The training of the model goes through 4000 iterations, before testing to acquire accuracy.

The default weights of the model are within (+/-) 0.02 of 3 for each of the 5 output neurons. This represents that given a binary array with no 1s (ones) the model will assume that the author has roughly average representation of all traits.

Results

There are 4 main observable results that can be taken from this research project and a further 4 new considerations for improving this project.

The first result of this model is counter intuitive to the amateur instinct of this research. Which is the assumption that the larger the word dictionary, the better the accuracy would become. The idea that a more granular assignment of words to OCEAN scores proved to create a model that predicted OCEAN scores that did not represent reality as show in Figure 5.

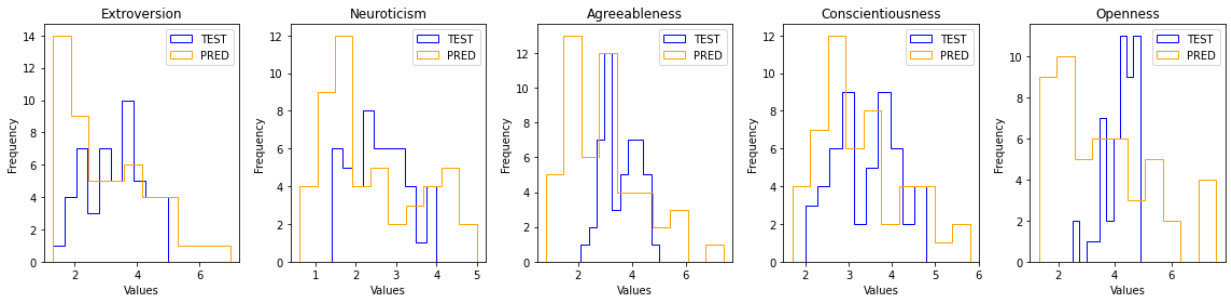

Figure 5: Large Word Dictionary (n = 10,000)

The issue with this is that generous trends were recognized where and values predicted even exceeded the total score range possible for the OCEAN model (1-5).

The second wrong assumption in creating the model was to reduce the number of iterations too low. This probed to create very accurate distributions for scores of some traits but showing very inaccurate distributions for other traits. This may mean that the traits measured by words inside the word dictionary, are more represented than others. This is a logical assumption since the word dictionary was not curated to represent the five traits equally. This is shown in Figure 6.

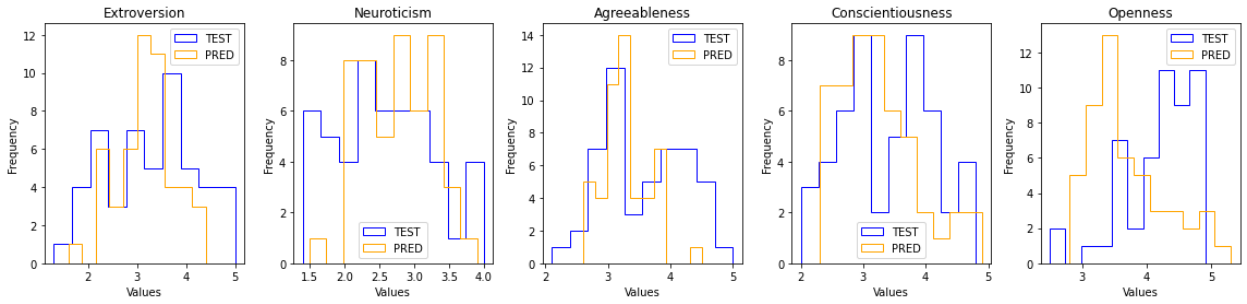

Figure 6: Skewed Representation of Trait Words

These first 2 realizations lead to an attempt to offset these two factors by balancing other parameters. The third realization was changing the solver type from “adam” to “lbfgs” to allow for much smaller training sized resulted in more accuracy but a distribution which favored the “hedging” of scores by largely concentrating predictions towards the average. The distribution shown in Figure 7 has a tell-tale “spike” of distribution towards the

center, which does not match with the distribution of real data, which can be described as “flatter”.

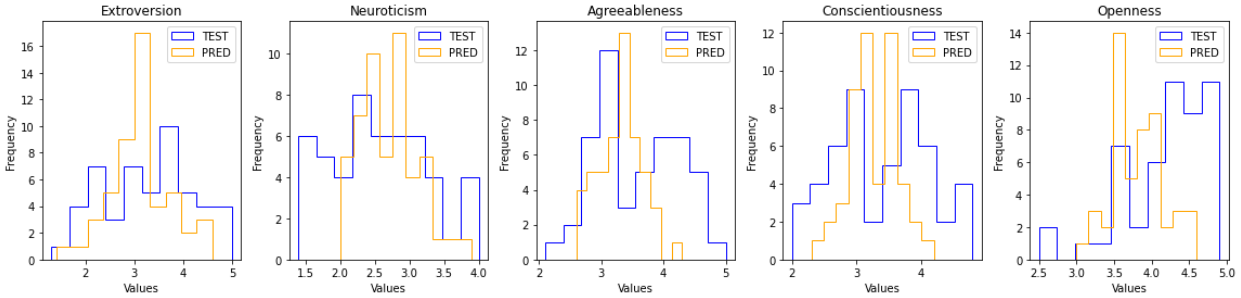

Figure 7: Spike Towards Average

The fourth and final realization of the model was that by adjusting the size of the first hidden layer of neurons the distribution could be shifted to represent the distribution of the true Twitter data more accurately. This can be seen in Figure 8.

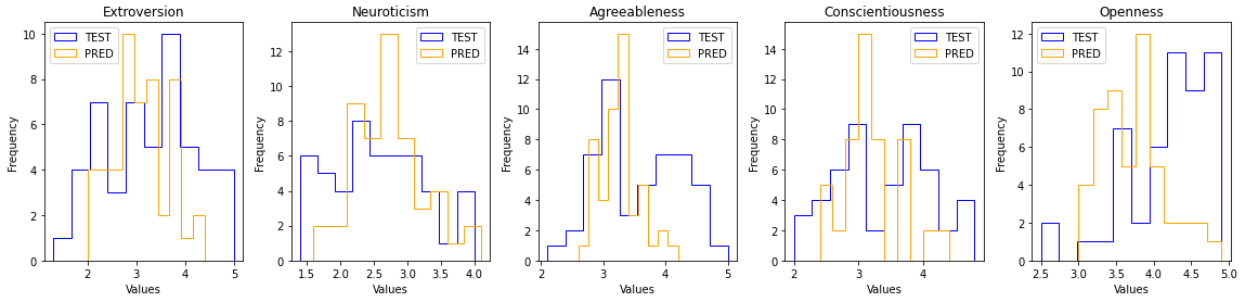

Figure 8: Adjust Hidden Layer Size

With these 4 realizations in mind the tuning of the model was pursued until no further noticeable gains could be made in accuracy and distribution mimicking. The outcome still represents a larger spike in the distribution than is desired, however skews of the distributions are mostly represented as shown in Figure 9.

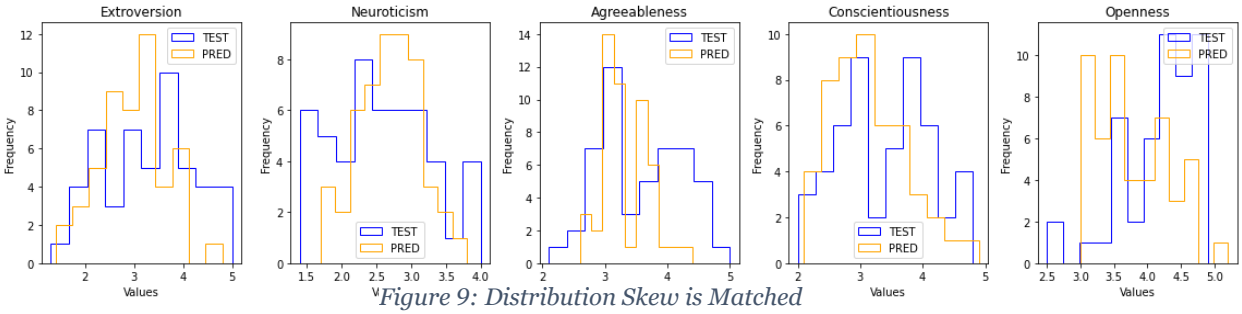

Figure 9: Distribution Skew is Matched

The final round of parameter tuning was optimized to match distribution as closely as possible while sacrificing some accuracy measured in RMSE and MAPE. The goal here was to try and see if the model could predict a very real distribution which ignored accuracy. The result of this attempt is shown in Figure 10.

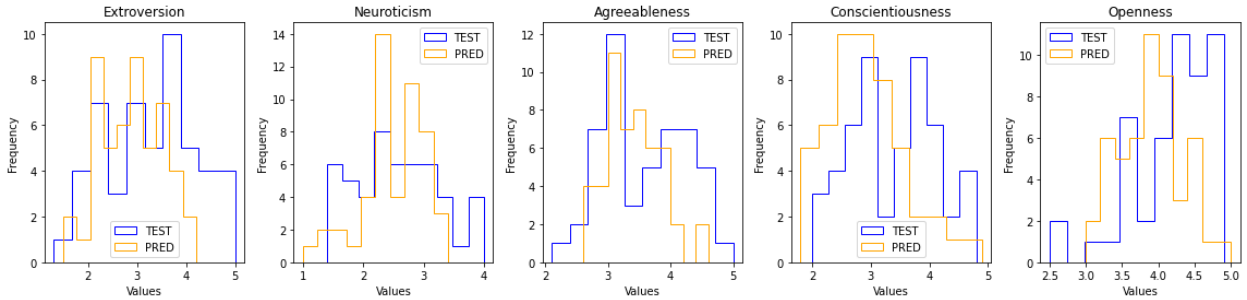

Figure 10: Sacrificing Accuracy to Optimize Matching Distribution

The final accuracy of the model when tuned to optimize accuracy measured by RMSE 0.73 (not promising for such a small range of linear output) When measuring MAPE the peak accuracy was 77%. A far departure from the accuracy of models created in other research papers (83%) [4]. There are other considerations as well. When attempting to balance maximizing accuracy and having a distribution of traits that reflects the population the accuracy was much lower by taking the average (69%). This is a troubling revelation, that this research was conducted without understanding the underlying reasons for why accuracy changed.

Conclusion

The model can be trained and predict values for OCEAN traits as desired, however even during the performance of the research project the shortcomings of the chosen methodologies became apparent. There were several obstacles in the way of achieving the desired accuracy of the model and functionality of the project.

The Bag of Words is Not Optimized

As previously stated, this is a valuable method for adding context to social media posts, as it is lightweight, accurate, simple, and requires no “Black Box” for reaching its conclusions, this is merely a logical “learning” of behavior through language expression and has been proven by others to be more than novel idea.[4] It is verifiable that the predictions achieved by this type of model can be very accurate assessments of personality traits. And when properly evaluated, even political ideology can be attributed to personality traits measured in a conforming way to the OCEAN model.[3] However. the honestly arbitrary selection of which words to trim from the dictionary resulted in counter intuitive predictions. Other research on the subject seems to suggest that not commonly occurring words but more specialized words were better for correlating to personality traits in both a positive and negative way. [5] Therefore for the model to progress some extra effort must be given to curate the bag-of-words. The author of this research, being nothing more than an amateur admirer of Machine Learning, did not consider this fact during the selection and trimming of the word dictionary that made the “Bag-of-Words”.

Variables Were Left Unconsidered

In essence it was ignored that this research was addressing a problem of not just multiple linear output prediction but also a multi-label classification problem. There was no attempt to classify or label traits garnered from the word dictionary. Two in-line models would be required to properly predict both elements of trait and linear ranking. The first model would focus on label prediction for the OCEAN traits and the multiple linear prediction would be made independently by a second model.[10] The truth is, that it was an unconsidered side effect that OCEAN traits are not mutually exclusive based on word. This means that word samples may have multiple trait labels as well as multiple linear assignments per word. Altogether the model should have separated the evaluation of traits with words and the linear assignment of those traits.

Considering the splitting of the model in two, there would be necessary provisions made for heuristic processing of sentences as well. This would be done to deal with elements of sentiment, polarity, and sarcasm to offset linear predictions without considering context. Without the previously mentioned concessions the model described in this research reached an average accuracy of 69% with maximum accuracy of 76%. Other research has reached accuracy levels of 83% [4] using roughly the same methods as described in this research: a neural network multi label output.

What is Being Measured?

With all of the preceding conclusions considered, and additionally mentioning that the Accuracy and Distribution could be optimized independently; this created a new wholistic revelation about the model. If the model measures distribution and accuracy separately, was the model predicting what it was intended to predict? The answer is, “Yes”, the model was predicting what it was designed to predict, albeit in a naïve way. Furthermore, the question arose, could the model be tinkered with to improve accuracy without re-creating the entire methodology of the model? As far as this research shows. The answer is, “No”.

How to Improve the Model

If the model were to be redesigned from the ground up different methods would be employed. The first method being to explore alternatives to Twitter. The fact that author status is limited to 280 characters is strictly limiting to the purpose of the model. Without the ability to capture larger sets of words, the ability of the model to predict personality traits is limited.

The model should be broken in two. The first section of the model should be changed to a multi label classifier. Which means first taking words in chunks to classify those chunks as correlating to an OCEAN trait. This can be easily achieved using a K-Means clustering model if words are considered additive to create multi-dimensional representations. The second part of the model would utilize Natural Language Processing, to measure elements of polarity, sentiment, and sarcasm to be added as inputs to the MLP for creating the linear prediction of the trait.

Finally, the bag of words would have to be curated to include the correct words, not just the commonly occurring words. The fact that a word occurs more often is detrimental to the classification of traits because there is a likelihood that multiple traits are being measured with every word. It is better to include fewer more specialized words to create the Bag-of-Words.

References

[1] Personality Prediction using Machine Learning, 2022 International Conference on Computing, Communication, Security and Intelligent Systems, Vijay Hima, & Sebastian Neenu.

[2] “A Neural Network Approach to Personality Prediction based on the Big-Five Model”, Mayuri Pundlik Kalghatgi, Manjula Ramannavar , Dr. Nandini S. Sidnal.

[3] “Personality Traits and Political Ideology: A First Global Assessment” Political Psychology Volume38, Issue5 October 2017 Pages 881-899, Matthias Fatke

[4] Lima A. C. E. S., de Castro L. N. (2014). A multi-label, semi-supervised classification approach applied to personality prediction in social media. Neural Networks, 58, 122–130

[5] Personality in 100,000 Words: A large-scale analysis of personality and word use among bloggers, J Res Pers. 2010 Jun 1; 44(3): 363–373., Tal Yarkoni

[6] https://openpsychometrics.org/_rawdata/

[7] https://mypersonality.org/ Stillwell, D. J., & Kosinski, M. (2015)

[8] https://scikit-learn.org/stable/user_guide.html SciKit Learn Supervised Model Documentation

[9] Sentiment Analysis and Sarcasm Detection of Donald Trump’s Tweets Using Machine Learning, 2020, Central Washington University, Andrew Dunn

[10] Başaran, S., & Ejimogu, O. H. (2021). A Neural Network Approach for Predicting Personality from Facebook Data. SAGE Open, 11(3).