Personality Prediction

Mohammad Zaid Shamshad

B20EE091

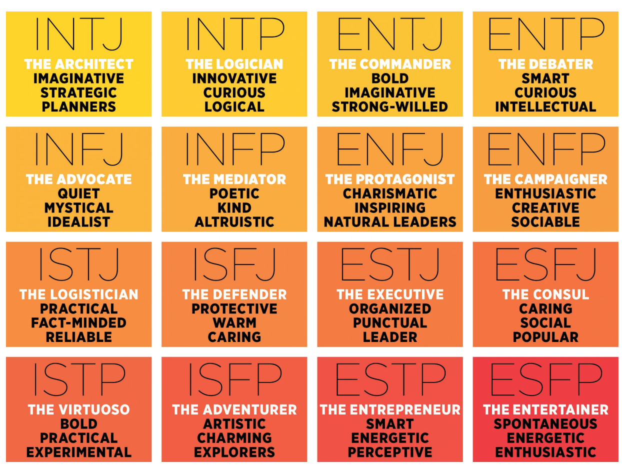
Github Repository: [Project]

Abstract – *My focus for this project is using machine learning to build a classifier capable of sorting people into their Myers-Briggs Type Index (MBTI) personality type based on text samples from their social media posts. The motivations for building such a classifier are twofold. First, the pervasiveness of social media means that such a classifier would have ample data on which to run personality assessments, allowing more people to gain access to their MBTI personality type, and perhaps far more reliably and more quickly. There is significant interest in this area within the academic realm of psychology as well as the private sector. For example, many employers wish to know more about the personality of potential hires, so as to better manage the culture of their firm. My second motivation centres on the potential for my classifier to be more accurate than currently available tests as evinced by the fact that retest error rates for personality tests administered by trained psychologists currently hover around 0.5. That is, there is a probability half that taking the test twice in two different contexts will yield different classifications. Thus, my classifier could serve as a verification system for these initial tests as a means of allowing people to have more confidence in their results. Indeed, a text-based classifier would be able to operate on a far larger amount of data than that given in a single personality test.*

I. Introduction

Personality is a way a person responds to a particular situation. It is a combination of characteristics that make an individual unique. Assessment of personality over the past two decades in various researches has revealed that personality can be defined by five dimensions known as Big Five personality traits. In general, the study of personality is considered as psychology research based on a survey or questionnaire.

In the scientific field of psychology, the concept of personality is considered a powerful but imprecisely defined construct. Psychologists would therefore stand to gain much from the development of more concrete, empirical measures of extant models of personality. My project seeks to improve the understanding of one such model: the Myers-Briggs Type Indicator (MBTI). I intend to use machine learning to build a classifier that will take in text (e.g. social media posts) as input and produce as output a prediction of the MBTI personality type of the author of said text. Successful implementation of such a classifier would demonstrate a strong linguistic basis for MBTI and potentially personality in general. Furthermore, the ability to produce an accurate text-based classifier has significant potential implications for the field of psychology itself, since the connection between natural language and personality type is non-trivial



Experiments

My main data set is a publicly available Kaggle data set containing 8600 rows of data. Each row consists of two columns: (1) the MBTI personality type (e.g. INTJ, ESFP) of a given person, and fifty of that person's social media posts. Since there are fifty posts included for every user, the number of data points is 430,000. This data comes from the users of personalitycafe.com, an online forum where users first take a questionnaire that sorts them into their MBTI type and then allows them to chat publicly with other users.

Classification Task

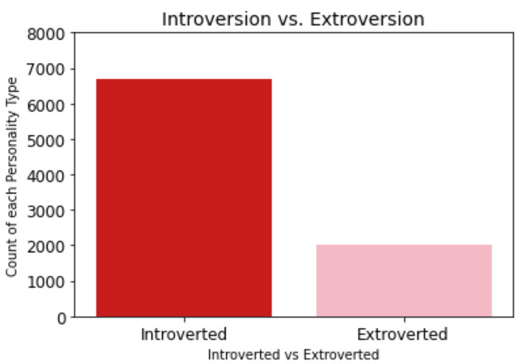
Due to the nature of the Myers-Briggs Type Indicator, we can break down the classification task with 16 classes into four smaller binary classification tasks. This is because an MBTI type is composed of four binary classes, where each binary class represents a dimension of personality as theorized by the inventors of the MBTI personality model. Therefore, instead of training a multi-class classifier, we instead train four different binary classifiers, such that each specializes in one of the dimensions of personality.

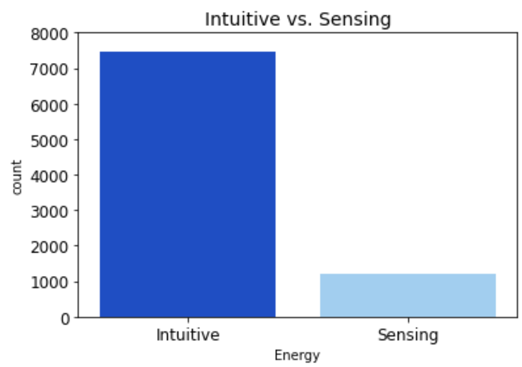
Background/Related Work

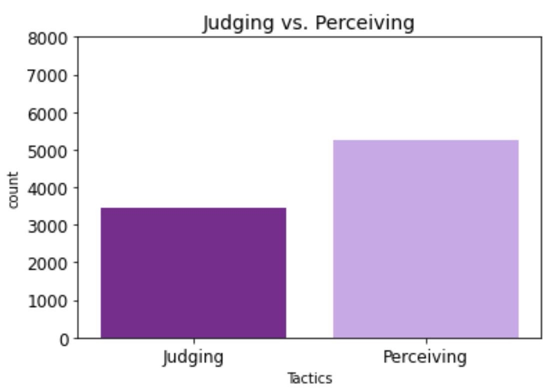
The MBTI personality classification system grew out of Jungian psychoanalytic psychology as a systematization of archetypal personality types used in clinical practice. The system is divided along four binary orthogonal personality dimensions, altogether comprising a total of 16 distinct personality types. The dimensions are the following. Extraversion (E) vs Introversion (1) is a measure of how much an individual prefers their outer or inner world. Sensing (S) vs Intuition (N): a measure of how much an individual processes information through the five senses versus impressions through patterns. Thinking (T) vs Feeling (F): a measure of preference for objective principles and facts versus weighing the emotional perspectives of others. Lastly, Judging (1) vs Perceiving (P): is a measure of how much an individual prefers a planned and ordered life versus a flexible and spontaneous life. There is a current debate over the predictive validity of MBTI regarding persistent pers

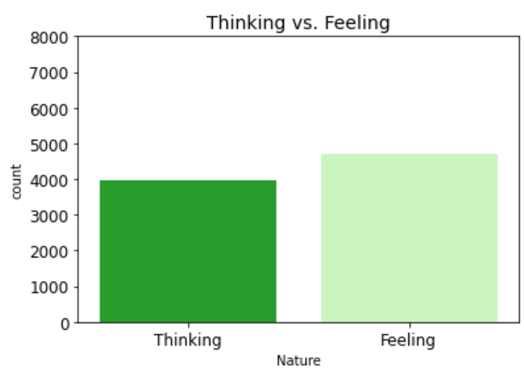
Exploring the dataset and pre-processing

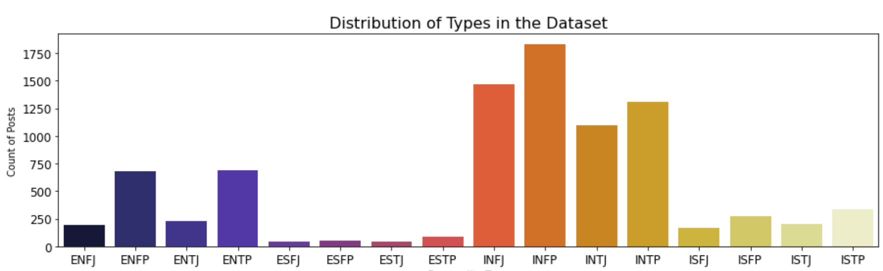
On counting the number of NULL values in the train dataset , it was found that there are no NULL values present.





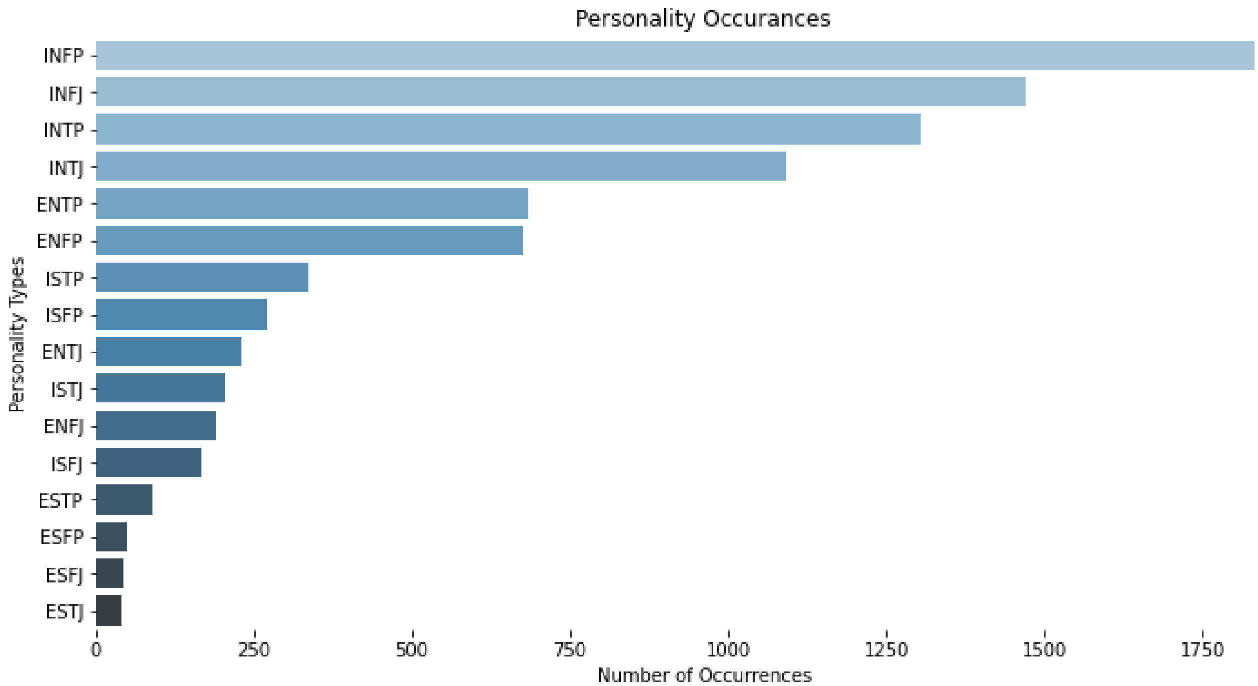






Proportionality

When I examined other studies of MBTI using machine learning, I was surprised to find that researchers rarely made a point of cleaning their data set to accord with the actual proportions of MBTI types in the general population (e.g. ISTJ = 0.11, ISFJ= 0.09, etc.). Since our raw data set is severely disproportional compared to the roughly uniform distribution for the general population,



Selective Word Removal *Since the data set comes from an Internet forum where individuals communicate strictly written text, some word removal was clearly necessary. For example, there were several instances of data points containing links to websites. Since I want my model to generalize to the English language, I removed any data points containing links to websites. Next, since I want every word in the data to be as meaningful as common filler words like "IS",I removed so-called "stop words" from the text (e.g." very" "a", "the", "or", etc.) using python's NLTK. Finally, since the particular data set we are working with comes from a website intended for explicit discussion of personality models, especially MBTI, we removed types themselves (e.g. 'INTJ', 'INFP', etc.), so as to prevent the model from "cheating" by learning to recognize mentions of MBTI by name.*

Removing URLs: *URLs don't add any value when analyzing texts therefore they were removed*

Lemmatization

I used nLtk. WordNetLemmatizer to lemmatize the text, meaning that inflected forms of the same root word were transformed into their dictionary form (e.g. "walking", "walked", "walk" all become "Walk"). This will allow us to make use of the fact that inflected forms of the same word still carry one shared meaning.

Stopwords:

Stopwords tend to have a negative influence over the accuracy of a model , therefore these will need to be removed. This will be done by vectorizer in the model building section, TfidfVectorizer has the ability of removing stopwords.

Tokenization:

Vectorise the words with TfidfVectorizer

Several iterations of parameters were applied for each characteristic.

For Mind classification, the best log loss result was achieved with the below parameters

- *max_features = 250 (this parameter that was adjusted the most when tuning)*
- *min_df = 4 (had very little effect, when using the max_features the lower frequency words aren't selected)*
- *max_df = 0.5 (consistently produced the better results with a 0.5 setting)*

Model

Implementation of classification algorithm

Initially a multi-class classification approach was taken, in which the model would use the input data to predict each row of input as one of the 16 personality types. The results were not very good and this led to the change in approach where each of the individual personality characteristics 'mind', 'energy', 'nature' & 'tactics' were classified separately as binary classification problems.
classification technique applied to the data: Logistic Regression
vectorizers were used: TfidfVectorizer
I have used 4 logistic models trained on individual columns for the four categories E/I S/N F/T P/J. The predictions are combined and converted to a string, and that string is returned as the final classification.

Evaluation

Accuracy:

Mind: Accuracy 0.853833
Energy: Accuracy 0.898907
Nature: Accuracy 0.860634
Tactics: Accuracy 0.801843

Model conclusions

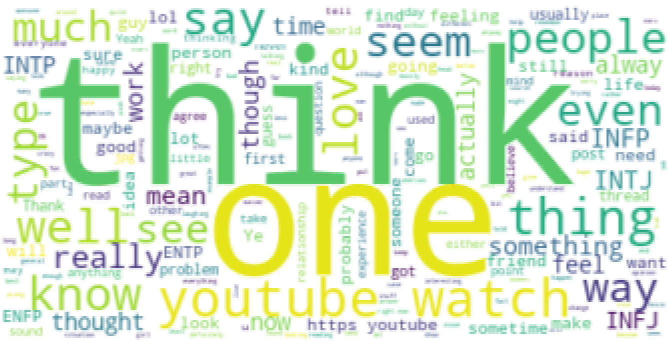
Of the models that were applied the best result was achieved with LogisticRegression with TfidfVectorizer. SVC with CountVectorizer is a very close second. MutlinomialNB & RandomForest performance was poor in comparison to these two.

Model	*Mind Log Loss*	*Energy Log Loss*	*Nature Log Loss*	*Tactics Log Loss*	*Average Log Loss*
Logistic Regression	*4.87*	*3.59*	*4.89*	*6.56*	*4.98*

Word Clouds For data visualization

I produced word clouds for concepts most prevalently used by specific classes of the personality dimensions. These were created by extracting the posts with the most extreme class probability predictions. These word clouds are then produced such that the size of each word is proportional to its appearance frequency in the top posts. I consider these word clouds to be illustrative of some of the unique ways different MBTIs use language. I vs. E (oft vs. right):

Intuitive



Sensing

Feeling

Thinking

Judging

Perceiving

Conclusion:

The overall accuracy of our trained model when classifying users is (0.85 x 0.89 x 0.86 x 0.80). While this seems to indicate a weak overall ability of our model to correctly classify all four MBTI dimensions, it should be noted that this number represents perfect classification and that it does not demonstrate the effectiveness of our model to achieve approximate predictions of overall MBTI types. In fact, other models that focus on multi-class classification of MBTI may achieve higher accuracy of perfect classification, but they do so at the risk of getting their prediction completely wrong. That is, multi-class classification treats all classes as independent of each other, and so they fail to capture to in-built relatedness of some types to other types eg. INFP is much more similar to INTJ than it is to ESTJ). That being said, my model represents a trade-off of these two aspects: I achieve lower rates of perfect classification in exchange for higher rates of approximately correct classification (i.e. "good" classification).

References:

[1] Estp archetype. https://www.16personalities.com/estp-personality.
[2] Mbti kaggle data set. https://www.kaggle.com/dataanaek/nbti-type.
[3] Mbti representation in the general population. http://aiweb.techfak.uni-bielefeld/. de/content/bworld-robot-control-software/
[4] Trump's mbti. https://www.kaggle.com/datasnaek/mbti-type.
[5] Jonathan S. Adelstein, Zarrar Shehzad, Maarten Mennes, Colin G. De Young, Xi-Nian Zuo, Clare Kelly, Daniel S. Margulies, Aaron Bloomfield, Jeremy R. Gray, F. Xavier Castellanos, and Michael P. Milham. Personality is reflected in brain's intrinsic functional architecture. PLOS ONE, 6(11):1-12, 11 2011.
6] Champa H N and Dr. KR Anandakumar. Artificial neural network for human behavior prediction through handwriting analysis. International Journal of Computer Applications, 2010. [10] James W Pennebaker and Laura A King, Linguistic styles: Language use as an individual