MBTI Personality Prediction Using Machine Learning

Abstract: The Myers-Briggs Type Indicator (MBTI) is a widely used personality assessment tool. It categorizes individuals into different personality types by assessing their preferences in four dichotomies: Introversion (I) - Extroversion (E), Intuition (N) - Sensing (S), Feeling (F) - Thinking (T), and Judging (J) - Perceiving (P). In this study, we apply NLP approaches to classify individuals' MBTI based on their social media posts. The MBTI classifiers predict four personality traits. Our aim is to provide accessible and accurate personality tests to a wider audience, reducing the error rates of current tests, as well as evaluate the validity of the current MBTI test.

I. INTRODUCTION

Within the field of psychology, personality is considered a powerful yet somewhat ambiguous construct. Thus, there is a need for more concrete and empirical assessments of existing personality models, providing valuable insights to psychologists. The Myers-Briggs Type Indicator (MBTI), is a widely recognized model (Boyle, 1995), evaluating people's characters in four dichotomies: Introversion (I) - Extroversion (E), Intuition (N) - Sensing (S), Feeling (F) - Thinking (T), and Judging (J) - Perceiving (P) (Briggs-Myers & Briggs, 1985). The most common method of obtaining our MBTI type is by finishing a MBTI test. A test taker can obtain an assessment of four dimensions by completing a series of questionnaires or multiple-choice questions. By combining the results from these four dimensions, a specific MBTI type is determined. For example, Jack gets a set of results in the four dichotomies, which are Introversion, Intuition, Feeling, Judging, respectively. Then his MBTI type is defined as "INFJ". Since MBTI can indicate how a person make reactions and thinks, it has lots of application in the field of Human Resources and Sales. By implementing MBTI test, employer can do a screening on the candidates and identifying the most job-matching candidate. However, it usually takes a long time to finish a complete version of MBTI test and candidates have incentive to lie on their preference on the test to get a favored result. Thus, a more convenient and objective methods is needed to solve these problems.

In this paper, we tried different model in machine leaning to develop a classifier capable of predicting an individual's MBTI personality type based on text inputs, such as social media posts. The successful implementation of such a classifier would demonstrate a strong linguistic foundation for the MBTI and potentially shed light on the broader field of personality research. Given the significant correlation between personality type and natural language, the development of an accurate text-based classifier holds immense potential for advancing the study of psychology itself.

II. DATASET DESCRIPTION

The MBTI dataset can be downloaded publicly on the website of Kaggle[1]. It's originally scrapped form a website called "Personality Cafe". The dataset contains 8675 inputs form different users. Each input consists of 50 recent comments from a user and each user is labeled the corresponding MBTI type.

	type	posts
8665	ENTP	'This test wasn't even close on my gender, age...
8666	INTJ	'Highly recommend this to those who wants to t...
8667	ENTP	'I think generally people experience post trau...
8668	INTJ	'Here's a planned stress relieving activity th...
8669	INFJ	'I'm not sure about a method for picking out l...
8670	ISFP	'https://www.youtube.com/watch?v=t8edHB_h908||...
8671	ENFP	'So...if this thread already exists someplace ...
8672	INTP	'So many questions when i do these things. I ...
8673	INFP	'I am very conflicted right now when it comes ...
8674	INFP	'It has been too long since I have been on per...

Form the first post we can find that each comment is separated by "|||", and some comments contains URL links and repeated punctuation marks, which we considered

[1] https://www.kaggle.com/datasets/datasnaek/mbti-type/data

meaningless.

III. DATA PRE-PROCESSING

A. Data Cleaning

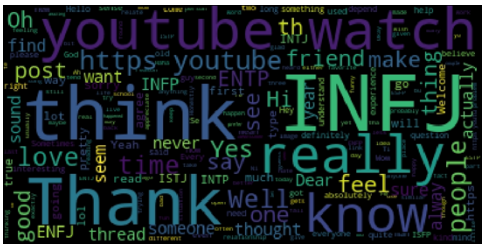
Firstly, since the data was obtained from online platform comments, it is obviously that users often incorporate symbols to form text emoticons or random combinations of numbers and symbols, such as ":)". These expressions are challenging for machines to comprehend, so we opt to remove them from the data. Secondly, users sometimes include image or video links within their comments, which represent as hyperlinks in the data set. Consequently, we replace these hyperlinks with spaces. Thirdly, in English text, certain words are extremely common but typically lack substantial meaning. These words, such as stopwords (e.g., "a," "the," "or"), can be disregarded for analysis purposes. To accomplish this, we utilize the stopwords corpus by NLTK to eliminate them from the data. Lastly, there are instances where words may possess different spellings while conveying identical or similar meanings. For instance, "apple" and "apples," "is" and "was." To facilitate model training, we employ the WordNetLemmatizer from NLTK to lemmatize words to their base or dictionary form (lemmatization) within the text.

There are some questions that deserve discussion:

1.Whether to retain punctuation marks:

There is a valid reason to consider the impact of punctuation marks on determining MBTI types. Individuals with a "Judging" personality type are likely to ask more questions compared to those with a "Feeling" personality type. Taking this into consideration, in the final model, I will attempt to replace various punctuation marks with specific English words, such as replacing "?" with "questmark", to examine whether it improves the overall accuracy.

2.Whether to remove MBTI vocabulary:



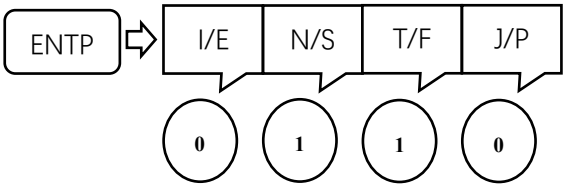
Word cloud of posts from INFJ

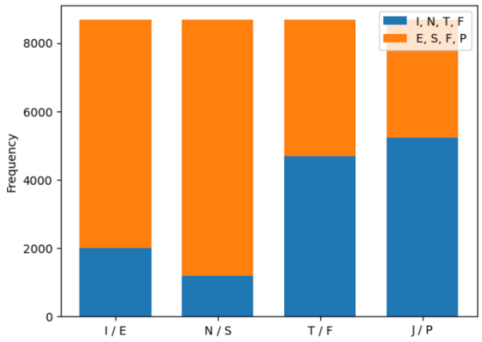
Based on word clouds, it is evident that users of different MBTI types tend to mention their own personalities in their discussions. This means that these individuals have already implied their MBTI in the text. Intuitively, retaining MBTI vocabulary could potentially improve the prediction accuracy. However, one concern is that if the model assigns high weights to these specific MBTI-related words, what happens when there are instances where certain samples mention MBTI terms that do not align with their actual types? Could this potentially mislead the final prediction results? To investigate this issue, I will use the final model to evaluate different approaches to data cleaning. This will help determine the optimal method for handling MBTI vocabulary and its impact on the accuracy of predictions.

B. Feature Engineering

In this study, our target variable is the MBTI type. Since we need to predict each of the four dimensions separately, I have added four additional variables to the original data, each corresponding to one of the four dimensions.

	posts	IE	NS	TF	JP
0	intj moment sportscenter top ten play prank l...	1	1	0	1
1	finding lack post alarming sex boring positio...	0	1	1	0
2	good one course say know blessing curse absol...	1	1	1	0
3	dear intp enjoyed conversation day esoteric g...	1	1	1	1
4	fired another silly misconception approaching...	0	1	1	1





In terms of input feature extraction, we use TF-IDF (Term Frequency-Inverse Document Frequency) to represent the importance of a term in a corpus (Ramos, 2003). TF-IDF combines two components: term frequency (TF) and inverse document frequency (IDF).

Term Frequency (TF) measures the frequency of a term within a document. It indicates how often a word appears in a specific document relative to the total number of words in that document.

TF = (Number of occurrences of term 't' in document 'd') / (Total number of terms in document 'd')

$$\mathbf{TF} = \frac{\mathbf{n}}{\mathbf{N}}$$

Inverse Document Frequency (IDF) measures the importance of a term in a corpus. It quantifies how rare or common a term is across all documents in the corpus.

IDF = log ((Total number of documents) / (Number of documents containing term 't'))

$$\mathbf{IDF} = \mathbf{log}\frac{\mathbf{N}}{\mathbf{n+1}}$$

The TF-IDF score of a term in a document is obtained by multiplying the term's TF by its IDF. The higher the TF-IDF score of a term in a document, the more relevant or important the term is to that document.

$$\mathbf{TF\text{-}IDF} = \mathbf{TF} * \mathbf{IDF} = \frac{\mathbf{n}}{\mathbf{N}} * \mathbf{log}\frac{\mathbf{N}}{\mathbf{n+1}}$$

C. Train-test Spilt

This research uses 80% of the data for training and 20% for test.

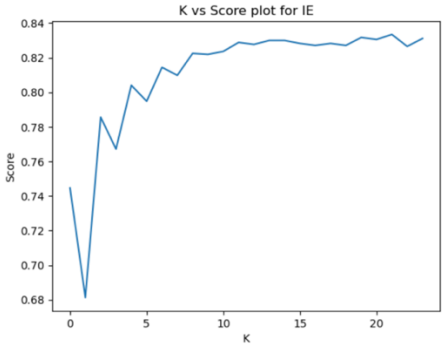
IV. METHODOLOGY

A. Model Specification

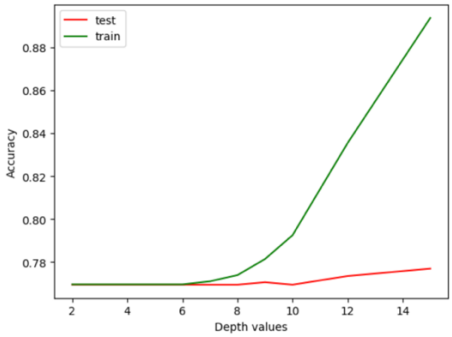
This paper investigates five machine learning methods for classification: Logistic regression (Logit), Gaussian Native Bayes (GNB), K-Nearest Neighbors (KNN). Random Forest (RF), Support Vector Machines Classification (SVC)

B. Cross Validation

Cross-validation is used to evaluate the performance of a machine learning model. It involves dividing the training data set into subsets, training the model on some subsets, and testing it on others. This process is repeated multiple times, with each subset serving as both training and testing data (Browne, 2000). By averaging the results, cross-validation provides an estimate of how well the model will perform on unseen data, selects the best model and tuning hyperparameters. This method helps to deal with overfitting and the issue of small sample size. In our research, we use 5-fold cross validation. That is, each model was trained on subsets for 5 times. “RandomizedSearchCV” is used to find the best hyperparameters for models with hyperparameters.



A visualization of how to find best k in KNN.



A visualization of how to find best depth in DT.

Below is the specification imposed on the 4 model with hyperparameters in this paper.

Logistic Regression (Logit)

(a). C: [0.01, 0.1, 1, 10, 100]

(b). solver: [lbfgs, liblinear, sag, saga]

(c). class_weight: [balanced, None]

K Nearest Neighbor (KNN)

(a). Numbers of neighbor: [3, 4, 5 … 26]

(b). Weights: [Uniform, Distance]

(c) Metric: [Euclidean, Manhattan]

Random Forest (RF)

(a). Numbers of estimators: [100, 200, 300]

(b). Maximum depth: [2, 3, 4 … 15]

Support Vector Machines (SVC)

(a). C: [0.1, 1, 10]

(b). gamma: [0.1, 0.01, 0.001]

(c). kernel: [rbf, linear, poly]

Model	Best Parameters for I/E
Logit	solver: liblinear class_weight: None C: 10
KNN	weights: distance p: 2 n_neighbors: 23
RF	n_estimators: 100 Maximum depth: 14
SVC	kernel: rbf gamma: 0.1 C: 10

C. Performance Evaluation

For each model, we calculated its accuracy, precision, recall, and F1 score in predicting the four MBTI dimensions (Joshi, 2016). Our primary focus was on accuracy, which measures the overall correctness of predictions. Additionally, we considered the F1 score, which provides a balanced measure by considering both precision and recall. The F1 score is particularly useful when dealing with imbalanced datasets. By examining these performance metrics, we gained insights into the model's predictive capabilities across the different dimensions of the MBTI.

Below are confusion matrices for I/E Prediction

I/E Prediction	Predicted I	Predicted E
Actual I	True Positives (TP)	False Negatives (FN)
Actual E	False Positives (FP)	True Negatives (TN)

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

V. RESULT & EVALUATION

	Accuracy (out sample test)				
	I/E	N/S	T/F	J/P	Overall
Logit	0.848	0.904	0.863	0.797	0.527
GNB	0.711	0.788	0.782	0.681	0.298
KNN	0.832	0.886	0.714	0.725	0.382
RF	0.778	0.862	0.828	0.656	0.364
SVC	0.862	0.904	0.866	0.786	0.534
	F1 score				Training
	I/E	N/S	T/F	J/P	time
Logit	0.904	0.946	0.851	0.733	6 min
GNB	0.797	0.869	0.763	0.607	2 s
KNN	0.900	0.937	0.595	0.537	102 s
RF	0.874	0.926	0.803	0.258	5 min
SVC	0.907	0.946	0.856	0.702	63 min

1. Comparing the accuracy and F1 scores of all models, it is not surprising that the Naive Bayes model performs poorly in all four dimensions. This is because the Naive Bayes model is based on the assumption of feature

independence, while in actual data, there is some correlation among the features, especially in the case of language vocabulary. The overall accuracy is 29.8%.

2. The SVC performs well in most dimensions. Since our post-processed data have very high dimensions, SVC's superior performance can be attributed to its capability to construct intricate decision boundaries in high-dimensional space (Ghaddar & Naoum-Sawaya, 2018). By leveraging kernel functions, SVC can effectively map data onto higher-dimensional feature spaces, where it can identify non-linear patterns and achieve better classification accuracy.
3. All the models do not perform well in the F/T dimension and J/P dimension; However, data in these two dimensions are more balanced. The problem might be that data imbalance can lead to upward bias: When one class has a significantly larger number of samples than the other classes in the dataset, the classifier may tend to classify samples into the majority class. In this case, the classifier's accuracy in I/E and N/S may appear higher because the majority of class samples are correctly classified most of the time.

Overall, SVC performs the best, and the Logit regression model shows similar effectiveness to SVC. Considering computational costs, we will use the Logistic regression model to validate the two issues mentioned earlier: whether to retain punctuation marks and whether to remove MBTI vocabulary.

Use I/E classification as an example, we find that the transform of punctuation marks will lead to 0.012 reduction in the I/E classification accuracy. The removing of MBTI vocabulary will lead to 0.113 reduction in the I/E classification accuracy.

VI. APPLICATION & CONCLUSION

This research validates the effectiveness of machine learning-based MBTI prediction models in accurately discerning MBTI types. Despite an overall accuracy of 53%, which may seem modest, it significantly outperforms the random guessing accuracy of 19% based on sample distribution. Moreover, considering our training dataset's limitations of a maximum of 50 posts per user, it can have great improvement in real-world applications, where we often have access to a larger volume of data for individual users. Hence, now we will try to apply this model to predict Elon Musk's MBTI using his Twitter (now "X") tweets data[2]. The data contains 3218 posts. After dropping the retweet posts, more than 2500 posts remain, which is far more than 50.

	row ID	Tweet	Time	Retweet from	User
3191	Row3190	Congrats to @dmetc…	2012–12–09 10:38:48		elonmusk
3192	Row3191	Interesting possible …	2012–12–05 07:20:41		elonmusk
3193	Row3192	RT @Slate: New gov…	2012–12–04 07:34:57	Slate	elonmusk
3194	Row3193	Am happy to report t…	2012–12–04 06:40:56		elonmusk
3195	Row3194	Uranium ore now av…	2012–12–02 23:12:03		elonmusk
3196	Row3195	@shervin Thanks Sh…	2012–11–29 12:00:15		elonmusk
3197	Row3196	RT @TheEconomist: …	2012–11–28 08:14:15	TheEconomist	elonmusk
3198	Row3197	Can't put my finger …	2012–11–28 03:19:37		elonmusk
3199	Row3198	But if humanity wish…	2012–11–27 20:00:03		elonmusk
3200	Row3199	And, yes, I do in fact…	2012–11–27 19:55:32		elonmusk

According to the predictions of the logistic regression model, Elon Musk's MBTI type is INTJ, which aligns perfectly with the MBTI record available on the internet[3].

```
IE_musk=model1_log.fit(X_train_IE,Y_train_IE).best_estimator_.predict(features_2)
NS_musk=model2_log.fit(X_train_NS,Y_train_NS).best_estimator_.predict(features_2)
TF_musk=model3_log.fit(X_train_TF,Y_train_TF).best_estimator_.predict(features_2)
JP_musk=model4_log.fit(X_train_JP,Y_train_JP).best_estimator_.predict(features_2)
```

```
print(IE_musk)
print(NS_musk)
print(TF_musk)
print(JP_musk)
```

```
[1]
[1]
[1]
[1]
```

Overall, the machine learning model constructed in this study has shown significant effectiveness in predicting MBTI types. It provides a novel approach to testing MBTI and can serve as an auxiliary tool to assess the reliability of MBTI test results.

Although our study provides a feasible approach for training the machine learning model, there are still several ways to improve its accuracy. Future research can explore improvement in the feature engineering process. For instance, alternative feature extraction strategies such as Word2Vec (Ma & Zhang, 2015) and BERT (Koroteev, 2021) could be employed. Even with TF-IDF, improvements can be made by considering Word-Level TF-IDF instead of the Character-level TF-IDF used in this study. Training the model to understand the meaning between words by incorporating larger word n-grams could be explored. To

[2] This data set is from econ4130 assignment 2.

[3] https://www.crystalknows.com/personality/elon-musk

deal with the data imbalance problem, methods like SMOTE can be implement. Regarding model selection, there is an opportunity to investigate a wider range of hyperparameters to obtain better-performing models. Furthermore, exploring the use of deep learning models could also potentially lead to improved accuracy. Overall, these avenues for future research hold the potential to further enhance the accuracy of the model.

References:

Boyle, G.J. (1995), Myers-Briggs Type Indicator (MBTI): Some Psychometric Limitations. Australian Psychologist, 30: 71-74. https://doi.org/10.1111/j.1742-9544.1995.tb01750.x

Briggs-Myers, I., & Briggs, K.C. (1985). Myers-Briggs Type Indicator (MBTI). Palo Alto, CA: Consulting Psychologists Press.

Browne, M. W. (2000). Cross-validation methods. Journal of mathematical psychology, 44(1), 108-132.

Ghaddar, B., & Naoum-Sawaya, J. (2018). High dimensional data classification and feature selection using support vector machines. European Journal of Operational Research, 265(3), 993-1004.

L. Ma and Y. Zhang, "Using Word2Vec to process big text data," 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA, 2015, pp. 2895-2897, doi: 10.1109/BigData.2015.7364114.

Joshi, R. (2016). Accuracy, precision, recall & f1 score: Interpretation of performance measures.

Koroteev, M. (2021, March 22). BERT: A Review of Applications in Natural Language Processing and Understanding. arXiv.org. https://arxiv.org/abs/2103.11943

Ramos, J. (2003, December). Using tf-idf to determine word relevance in document queries. In Proceedings of the first instructional conference on machine learning (Vol. 242, No. 1, pp. 29-48).

Thirumuruganathan, S. (2010). A detailed introduction to K-Nearest Neighbor (KNN) algorithm.