MBTI Personality Prediction into Judging vs. Perceiving using Machine Learning Approach

Xinyi Rong Yang Li

1 Introduction

The Myers-Briggs Type Indicator or MBTI (Briggs-Myers & Briggs, 1985) is a popular introspective self-report questionnaire [2]. It is a useful tool to quantify non-psychopathological personality types based on the theory of psychological type by Carl G. Jung [3]. It explains the individual's decision-making process, perception of the world, and the mechanisms of interaction with the external environment.

The current North American English version of the MBTI Step 1 includes 93 forced-choice questions (88 are in the European English version). "Forces-choice" means a person should choose only one of two possible answers to each question. Each person's personality type is described by a four-letter code with a short descriptive explanation at the end of the report form. The results are presented through four dichotomies: Extraversion vs Introversion, Sensing vs Intuition, Thinking vs Feeling and Judging vs Perceiving. There are 16 possible four-letter codes. For example, someone who prefers Extraversion, Sensing, Feeling and Perceiving would be labeled an ESFP in the MBTI system.

Recent studies [6, 7, 8] show that people's personality traits influence their choices such as career, job and so on using the 4 axes. According to the results of their works, we want to discover whether one axis can classify users very well. This paper aims to discover a person's personality traits according to Judging vs Perceiving. The data is collected from a professor containing 8,675 rows of the data. Each row is a person's type (4 letters) and posts. It can be divided into two main parts. The first is data preprocessing, which processes the data so that it can be fed into the next algorithm. The second is machine learning algorithm testing consisting of supervised machine learning and unsupervised machine learning.

We implemented three clustering models [11] (K-Means Clustering, Spectral Clustering, and Agglomerative Clustering) and then evaluated by 5 metrics (Adjusted Rand Index, Homogeneity, Completeness, V-Measure and Fowlkes-Mallows Index). The experiment results show that no model can show outstanding results. Those models cannot cluster dataset into several groups. Based on those three models, K-Means performs better. For classification [5], we proposed 10 models and evaluated them by accuracy, precision, recall, and F-1 score. Multi-layer Perceptron performs best. Furthermore, we also tested the performance of different combinations of feature groups, namely metadata,

sentiment scores and language features, on these models. Experimental results show that for each model, language features are the most important. Metadata and sentiment scores may even have a negative impact on the performance of the classification.

2 Data Description & Preprocessing

2.1 Data Description

The data is from the professor and we can also find this data online through the PersonalityCafe forum. So, there is no missing data, duplicate data. This data consists of 8,675 rows. Each row is a person's type (The person's 4 letter MBTI type) and posts (Each entry separated by "|||" (3 pipe characters)). For each person's posts, they consist of non-ASCII characters, ASCII characters, URLs, numbers.

2.2 Preprocessing

1. Separate each entry. For each user, the posts column consists of several posts. So, we separated it using "|||".

2. URLs are replaced by placeholders '__url__'.

3. #hashtag. Convert #words to '__hashtag__'. Because a hashtag may contain multiple words, perhaps the first letter is capitalized, maybe not, which makes it difficult to extract. So we replace them directly with placeholders.

4. @username are replaced by placeholders '__at__'.

5. Emoji words like ":proud:" are processed by removing the colons directly at both ends.

6. Numbers are replaced by placeholders '__number__'.

7. All non-ASCII characters are discarded.

8. Convert all characters into lowercase.

9. Remove all punctuation and English stop words.

10. Stemming. Stemming algorithms are used to search the "root" or base word of a given word. We have used Snowball Stemming Algorithm.

3 Preliminary Analysis

3.1 Basic Statistic

After data preprocessing, we did some preliminary statistics. In terms of the total number of users. there are 8,675 users including 3,434 of Judging and 5,241 of Perceiving. In terms of the total number of posts, there are 422,845 posts including 167,110 of Judging and 255,735 of Perceiving. There are 98,061 different words in our data.

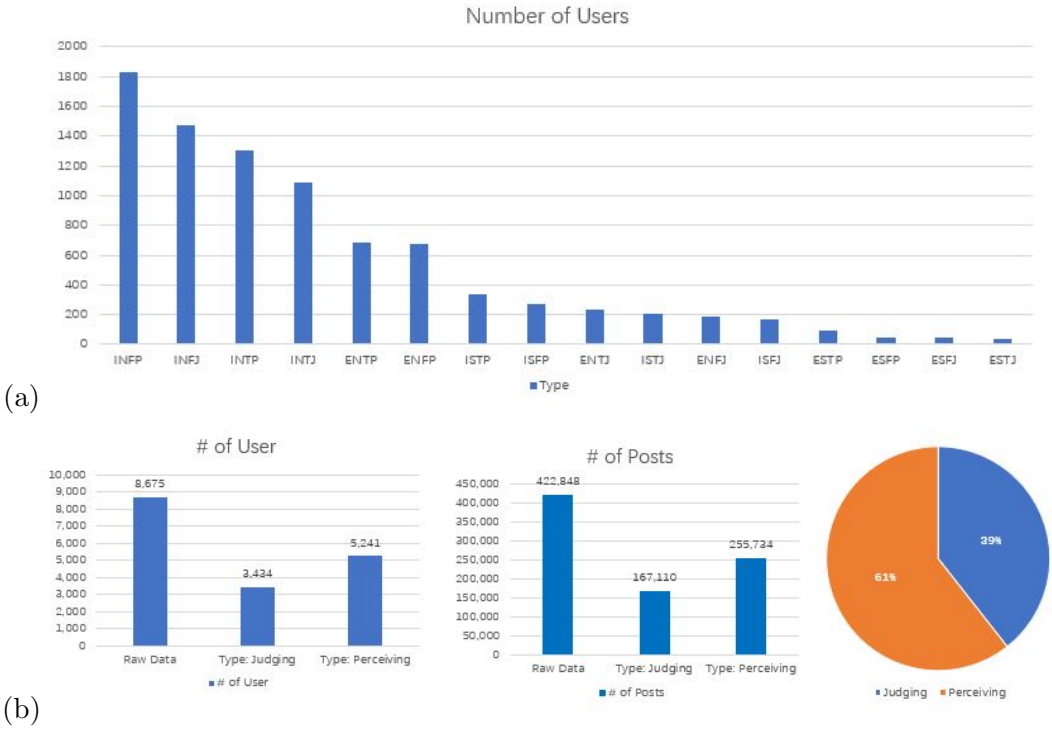


Figure 1: (a) Number Of User per Type (b) Statistics of Two Types

First, we counted the number of users per type (16). According to Figure 1, the number of types (INFP) has the largest number of people. There are more than eight times as many people of type INFP as of type ESTJ. It shows that the dataset is **unbalanced**. However, because we only consider 2 types: Judging vs Perceiving, we assumed that there is no situation as Figure 1(a). As shown in Figure 1(b), the left one is the number of users per type(Judging vs Perceiving), the middle one is the number of posts per type(Judging vs Perceiving), the right one is the percentage of two types (Judging vs Perceiving). All those figures show that the dataset has the same problem even though we just consider two types(Judging vs Perceiving). However, having an unbalanced dataset is not good for business in general as the machine will tend to favor the prediction of the majority. They ignore our minority class almost totally. No matter how small the representation, we still want the machine to be able to predict the minority type. Resampling is an approach to overcome this problem.

Figure 2 shows the word frequency of different words. The x-axis is frequency. The y-axis is the log value of the total number of words with such frequency. According to this figure, we can determine how many words should be filtered out.

3.2 Word Cloud

To better understand our data, different types (Judging vs Perceiving) of word frequency are

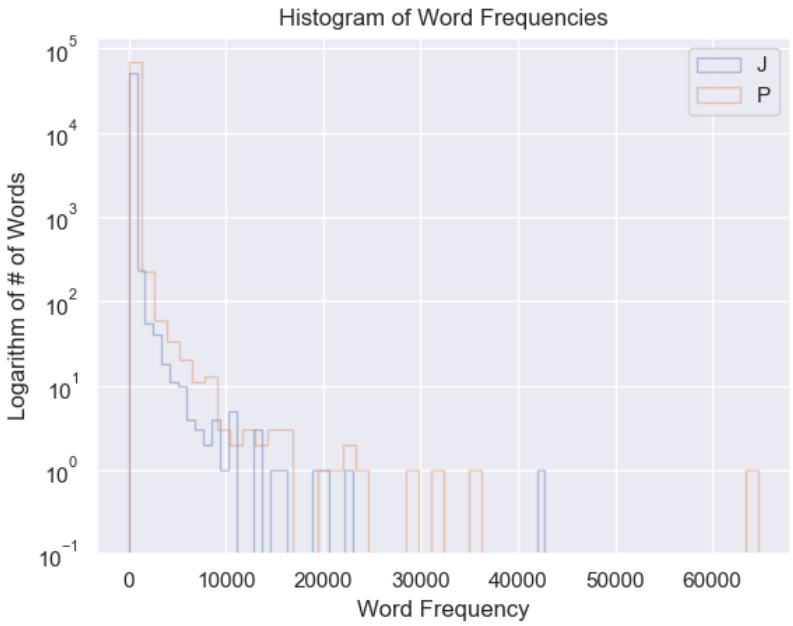


Figure 2: Word Frequency

counted and the first 15 words are shown in Table 1 and Table 2. From those two tables, we can get that word frequency in type "Perceiving" is higher than that in type "Judging". We highlighted the same words in these two types. In our opinion, we cannot get any meaningful information. Those words in Tables 1 and 2 are some normal words instead of words characterized by different types. We have already got all the word frequency. For the sake of demonstration, we only wrote the top 15 words. To prove our opinion that is directly using raw data is not a good idea, we also show the word cloud.

Figures 3(a) and 3(b) illustrate the most frequent word in a different area. Figures 3(a) shows the high frequency of the words "people", "like", "think" and "get". Figure Figures 3(b) shows the same result. However, the weight of each word is different. Those words are not meaning or unrepresentative. They are some normal words we use in daily life.

We know that the first way of analyzing word frequency is not effective based on words cloud and word frequency shown in Figures 3(a) and 3(b) and Table 1 and 2. Furthermore, we came out with two different ways which can better analyze our data.

1) The first one is normalizing each frequency based on each type's word number. By normalizing, each word frequency's range is scaled from 0 to 1. Furthermore, subtracting the part of the word frequency common to both types. As shown in Figure 4(a) and 4(b), words cloud have different words in different types. "INTJ" and "INFJ" appears more in type "Judging". "INFP" and "INTP" appears more in type "Perceiving". Let's go back to Figure 1(a). Of the number of people in type with "Judging", the number of "INTJ" and "INFJ" are the most. Same thing for type "Perceiving".

Type: Judging	
Word	Word Frequency
__number__	30,011
like	29,231
think	25,884
people	19,289
get	19,131
know	17,312
would	15,813
one	15,803
__url__	15,772
say	15,435
feel	14,915
make	13,792
thing	13,572
time	13,512
go	13,409

Table 1: "Judging" type Word Frequency

Type: Perceiving	
Word	Word Frequency
like	46,626
__number__	45,658
think	40,247
get	31,025
people	28,691
know	25,512
__url__	25,250
one	23,922
would	23,408
say	22,248
feel	22,146
realli	22,139
go	20,892
thing	20,846
time	13,512

Table 2: "Perceiving" type Word Frequency

That's why we got those results. The underlying cause is an unbalanced dataset.

2) The second one is fetching the top 10 words of each type. Setting the intersection of each type's top 10 words as "stop-words". Removing these "stop-words" from both word frequencies. As shown in Figures 5(a) and 5(b), each type has a different word frequency. Because Judging types tend to have a structured way or theory to approach the world. Perceiving types tend to be unstructured and keep options open. They use different words to express themselves.

3.3 Sentiment Analysis

We also computed sentiment scores using VADER (Valence Aware Dictionary and sEntiment Reasoner) library. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media, and works well on texts from other domains. Note that to compute the sentiment score, we will have to work with the original posts, not the pre-processed ones. VADER will returns 4 scores: positive, negative and neutral. As shown in Figure 6, the left figure is sentiment "positive" score, the middle one is the sentiment "negative" score, the right one is the sentiment "neutral" score.

Using compound score will be better to understand the sentiment of each type. Compound is the overall score of a post. If the compound score is between -0.05 and 0.05, this post is considered as neutral, otherwise this post will be considered as positive or negative. We analyzed the sentiment

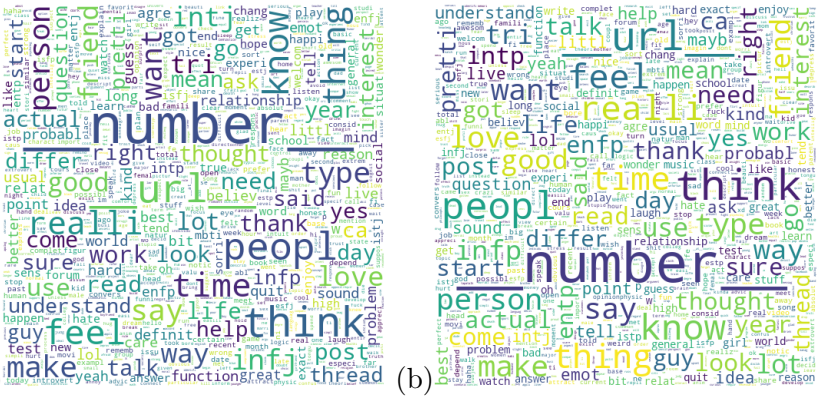


Figure 3: (a) Word Cloud of Type "Judging" (b) Word Cloud of Type "Perceiving"

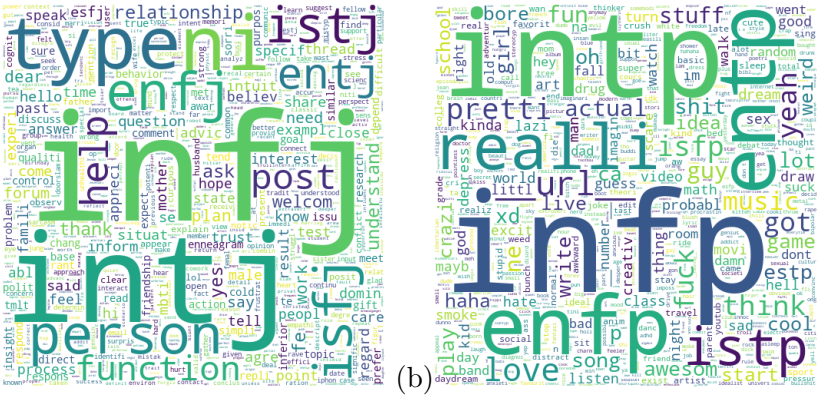


Figure 4: (a) Word Cloud of Type "Judging" (b) Word Cloud of Type "Perceiving"

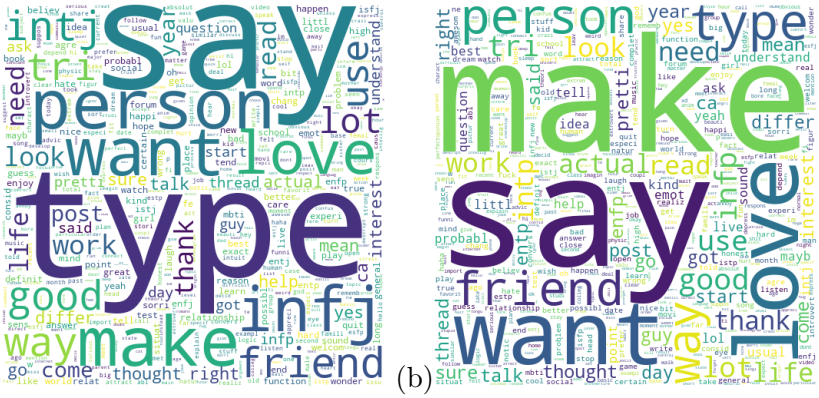


Figure 5: (a) Word Cloud of Type "Judging" (b) Word Cloud of Type "Perceiving"

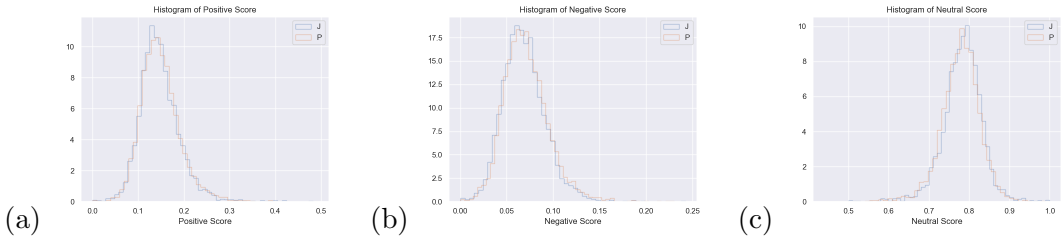


Figure 6: Distribution of Positive, Negative and Neutral Sentiment Scores

score of type "Judging" and type "Perceiving". Based on Figure 7, most posts are positive, but their compound scores are not too high.

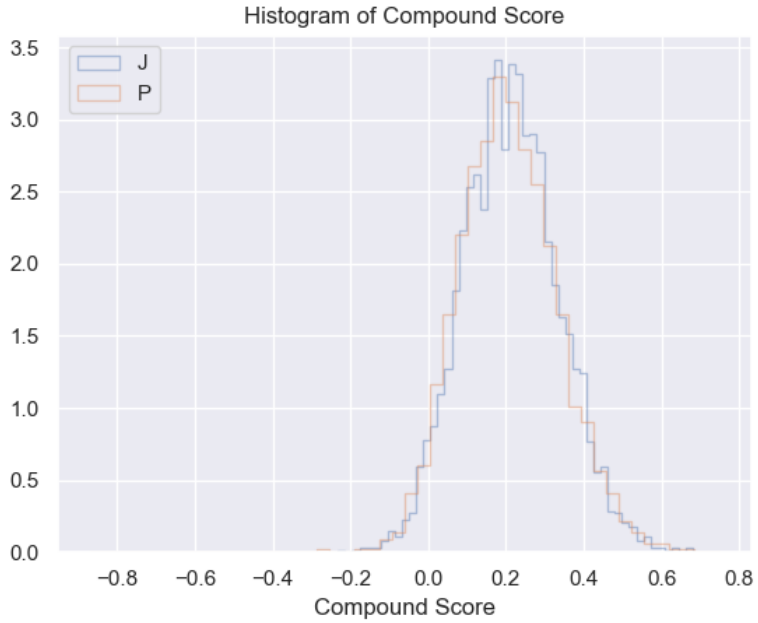
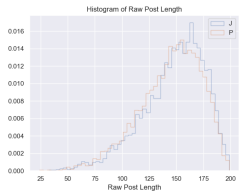


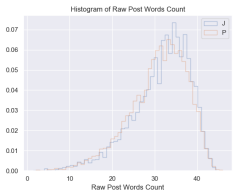
Figure 7: Distribution of Compound Sentiment Score

4 Feature Extraction

In this section, we discuss the features extracted. During preprocessing, we convert some tags into placeholders such as converting URL into '__url__'. The number of placeholders indicates the trend in the content of posts. So, we also consider them as part of features.

For each user, we first compute the following features for each his posts. Then take the average value among his posts as the user's features.

(a) 

(b) 

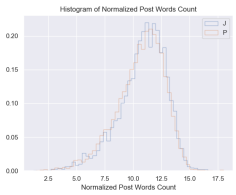
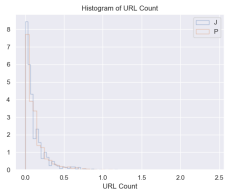
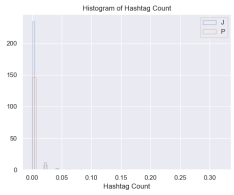
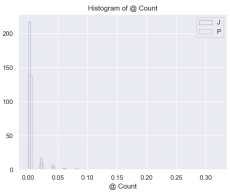
(c) 

Figure 8: (a) Raw Post Length (b) Post Words Count (c) Normalized Post Words Count

(a) 

(b) 

(c) 

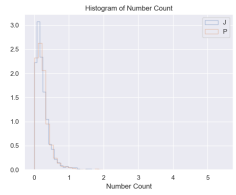
(d) 

Figure 9: (a) URLs (b) Hashtags (c) @mentions (d) numbers

4.1 Metadata Extraction

According to the analysis of different metadata below, there is no obvious difference in the number distribution of each placeholder of "Judging" and "Perceiving" based on the histograms.

1. Average length of raw post (in characters)

Computing the average length of each user's posts. Figure 8(a) demonstrates most users' post is about 150 to 160 characters.

2. Average count of raw post words

Using the tokenization results, we counted the number of words per post. Most users' posts have around 30 to 40 words according to Figure 8(b).

3. Average count of normalized post words

In normalization, we removed some characters that are not words, which will also eliminated some words. Figure 8(c) is the number of words per post after normalization. In general, the average word count is 10 to 13 per user.

4. Average count of the number of URLs

Compute average number of URLs placeholders('__url__').

5. Average count of the number of Hashtags

Compute the average number of Hashtags placeholders('__hashtag__').

6. Average count of the number of @mentions

Compute the average number of @mentions placeholders('__at__').

7. Average count of the number of numbers

Compute the average number of number placeholders ('__number__').

4.2 Sentiment Feature Extraction

As we discussed above, each post has positive, negative and neutral sentiment with different percentage. The sentiment information can be part of the features. So, we used 4 sentiment features: 1) Average score of post positive score, 2) Average score of post neutral score, 3) Average score of post negative score, 4) Average score of post compound score.

4.3 Language Feature Extraction

In our project, we decided to extract features using unigram+bigram and TF-IDF.

The TF-IDF value increases the number of words in the document but is usually offset by the frequency of the words in the world, which helps to adjust the fact that some words appear more frequently in general situations.

TF-IDF uses two statistical methods, the first is the term frequency and the other is the inverse document frequency. The term frequency refers to the total number of times a given term t appears in the document doc, and how much information is provided in the total number of words in the document and the inverse document frequency measure. It measures the weight of a given word in the entire document. IDF shows that given words are common or rare in documents.

To make the features more powerful, we use unigram+bigram to generate "words" for TF-IDF.

5 t-SNE

Before clustering, we implemented t-SNE(t-distributed Stochastic Neighbor Embedding) which is a tool to visualize high-dimensional data. Because the number of language features we got is too large to be fed into t-SNE, we used feature selection [4] method to extract 500 language features. Figure 10 shows the result of t-SNE on each feature groups.

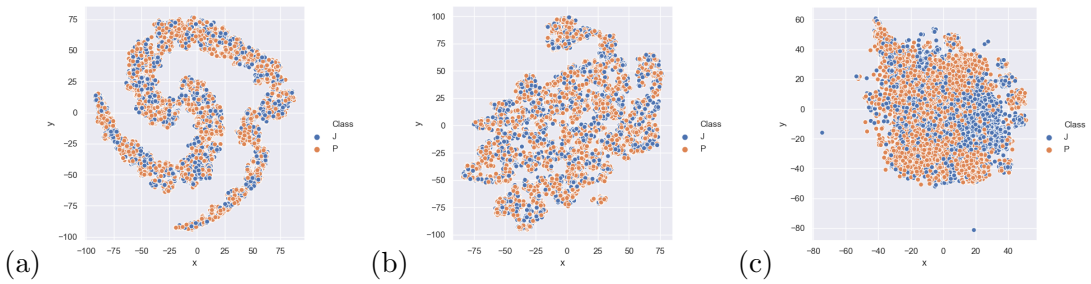


Figure 10: (a) t-SNE on Metadata (b) t-SNE on Sentiment Scores (c) t-SNE on TF-IDF Features

The metric we used to selected features is Chi-square test with p-value. Chi-square test is a statistic test of independence to determine the dependency of two variables. The higher the score, the more relevant, and the smaller the p-value, the more reliable. Table 3 shows some words with highest score.

word	Chi-square score	p-Value
infj	332.914658	0.000000
intj	199.409089	0.000000
infjs	142.462627	0.000000
infp	118.744865	0.000000
ni	95.762359	0.000000
intp	89.298632	0.000000

Table 3: Chi-square with p-Value

6 Unsupervised Machine Learning

We implemented 3 clustering models (K-Means Clustering, Spectral Clustering and Agglomerative Clustering) with 5 evaluation metrics (Adjusted Rand Index, Homogeneity, Completeness, V-Measure and Fowlkes-Mallows Index) using bigram, tf-idf and metadata features.

6.1 Clustering Methods

1. K-means clustering is one of the simplest and most popular unsupervised machine learning algorithms. The k-means algorithm identifies k centrosomes and then assigns each data point to the nearest cluster while keeping the centrosomes as small as possible. We set $k = 2$ based on our mission.

2. Spectral Clustering is a technique with roots in graph theory, where the approach is used to identify communities of nodes in a graph based on the edges connecting them. The method is flexible and allows us to cluster non-graph data as well. Spectral clustering uses information from the eigenvalues (spectrum) of special matrices built from the graph or the data set. We'll learn how to construct these matrices, interpret their spectrum, and use the eigenvectors to assign our data to clusters.

3. Hierarchical clustering is a general family of clustering algorithms that build nested clusters by merging or splitting them successively. This hierarchy of clusters is represented as a tree (or dendrogram). The root of the tree is the unique cluster that gathers all the samples, the leaves being the clusters with only one sample. The AgglomerativeClustering object performs a hierarchical clustering using a bottom-up approach: each observation starts in its own cluster, and clusters are successively merged together.

6.2 Evaluation Metrics

Only using one way to evaluate the cluster models is not persuasiveness. So we used 5 metrics to evaluate the cluster models. All metrics are from different aspects of judging the model. By considering all metrics, we can figure out which model is the best one. Table 4 shows the results.

Model	Adjusted Rand Index	Homogeneity	Completeness	V-Measure	Fowlkes-Mallows Index
K-Means Clustering	0.0290	0.0222	0.0215	0.0219	0.5249
Spectral Clustering	0.0256	0.0190	0.0184	0.0187	0.5233
Agglomerative Clustering	0.0242	0.0175	0.0169	0.0172	0.5228

Table 4: Evaluation Results

1. Adjusted Rand Index computes a similarity measure between two clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusterings.

2. Homogeneity metric. A clustering result satisfies homogeneity if all of its clusters contain only data points that are members of a single class. This metric is independent of the absolute values of the labels: a permutation of the class or cluster label values won't change the score value in any way.

3. Completeness. A clustering result satisfies completeness if all the data points that are members of a given class are elements of the same cluster. This metric is independent of the absolute values of the labels: a permutation of the class or cluster label values won't change the score value in any way.

4. V-Measure. The V-measure is the harmonic mean between homogeneity and completeness: $v = (1+\beta) \times \frac{homogeneity \times completeness}{\beta \times homogeneity + completeness}$

5. Fowlkes-Mallows Index. The Fowlkes-Mallows index (FMI) is defined as the geometric mean between of the precision and recall: $FMI = \frac{TP}{\sqrt{((TP+FP) \times (TP+FN))}}$ Where TP is the number of True Positive, FP is the number of False Positive and FN is the number of False Negatives. The score ranges from 0 to 1. A high value indicates a good similarity between two clusters.

6.3 Discussion

From Table 4, we can see no model performs well. As we discussed, the score ranges from 0 to 1. A high value indicates a good similarity between two clusters. However, no result is above 0.5. We analyzed the results. We think that is because the number of features is too small. We only used 500 selected features which are not enough to describe the dataset. So, the models cannot gather enough information to cluster the data. So, if we cluster dataset using all features, the results will be better. Due to time constraints, we were unable to retrain the three models. So, we only rerun the K-Means clustering model using all features. The result is better. All evaluation metrics' results are larger than 0.5.

7 Supervised Machine Learning

Our mission is to classify users into two classes: Judging vs. Perceiving. So, the classification problem is binary. We implemented 10 classifiers (shown in Table 5) with 4 evaluation metrics under

Model	Accuracy	Precision (Weighted)	Recall (Weighted)	F-1 score (Weighted)
Gaussian Naive Bayes	0.73723	0.76774	0.73723	0.70945
Bernoulli Naive Bayes	0.78371	0.78264	0.78371	0.78303
K-Neighbors	0.60123	0.58414	0.60123	0.5846
Ridge	0.8083	0.81112	0.8083	0.80288
Linear w/SGD	0.80407	0.80269	0.80407	0.80284
Linear SVC	0.83365	0.83389	0.83365	0.83107
RBF SVC	0.69535	0.69148	0.69535	0.67949
Decision Tree	0.76681	0.76439	0.76681	0.76414
Random Forest	0.80638	0.80874	0.80638	0.80107
Multi-layer Perceptron	0.84518	0.84501	0.84518	0.84509

Table 5: Classification Results

K-Fold cross validation which can tune the hyper-parameters of the model. The evaluation metrics we used are accuracy, precision, recall and F-1 score. All metrics are computed by confusion metrics. Figure 11 shows the definition of confusion matrix. Accuracy can be computed by $Accuracy = \frac{TP+TN}{Total}$. Precision can be computed by $Precision = \frac{TP}{TP+FP}$. Recall can be computed by $Recall = \frac{TP}{TP+FN}$. F-1 socre can be computed by $F-1 = 2 \times \frac{precision \times recall}{precision+recall}$.

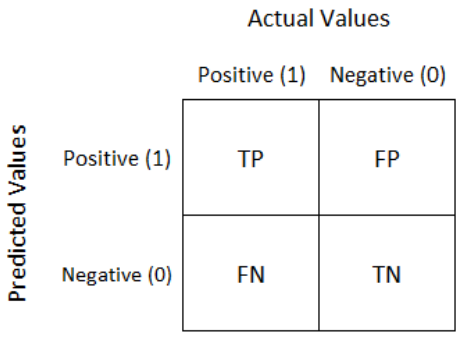


Figure 11: Confusion Matrix Definition

7.1 Classification Results

We randomly divided the dataset into training set (70%) and test set (30%) and then fed them into the models we used, and then trained the models. The most time-consuming part is how to find a set of hyperparameters. To find the best combination, we use 5-fold cross validation on the training set for each candidate combination. Then use the best hyperparameter setting to test the classifier's performance on the test set. Table 5 shows the results. We can see that Multi-layer perceptron performs best.

7.2 Impact of feature groups

Our features can be divided into three groups, metadata, sentiment scores and TF-IDF features. Using the hyperparameter settings we found in the previous step, we evaluated the models with different combinations of feature groups. Figure 12 shows the results.

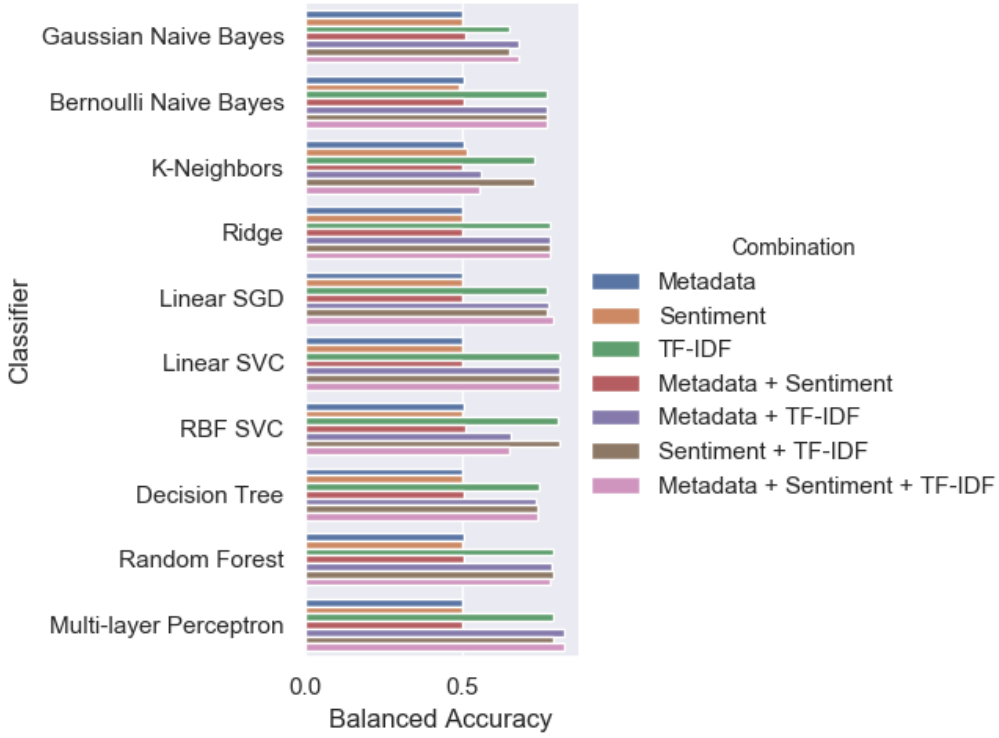


Figure 12: Distribution of Compound Sentiment Score

It is clear that the TF-IDF feature contributes the most. Metadata and sentiment scores have almost no contribution, and the classification results when using them are approximately random output. In some models they can even reduce the accuracy of classification.

8 Environment/Libraries

In this paper, we finished all processes in Python. And in Table 4 we listed all libraries we used during this project.

9 Related Research

Stilwell et al. [10] studied differences in MBTI tests among current medical students. Identifying the differences between men and women on the MBTI and explore the links between specific types and

Libraries Name
Numpy
Pandas
Seaborn
Matplotlib
NLTK
Scikit-learn
vaderSentiment
Wordcloud

Table 6: Libraries

medical career choices. Based on the research, individuals with introverted and emotional personality preferences are more likely to choose primary care majors. Extroverted and reflective individuals are more likely to choose surgery as their major.

Other researches [7, 8, 1, 9] shows different personalities will make different choices based on their MBTI testing report. As the saying goes, there are no two identical people in the world. Different people make different decisions about things. The MBTI test quantifies a person's personality as a variable that can be analyzed for behavior. However, most of the research combined 4 axis personality types identifying a person.

Anthony Ma et al. [6] studied that an individuals writing style is largely coupled with their personality traits and present a deep learning model to predict Myers Briggs Personality Type. They delved into a more complex long-short term memory based recurrent neural network and aim to build a more generalizable system that can incorporate meaning of writing to determine overall personality types. However, their results are not good enough. We think the performance of their approach could be higher.

From those studies we read, they all showed their idea about how to deal with a old dataset with a new idea. The ways of feature extraction they used help us broaden our horizon. In this paper, we came out with some new ideas about feature extraction. And we added a new step: feature selection into our processes. Based on their experiments, our classification accuracy is around 84% which is higher than Neural Network [6].

These studies have shown that human personality influences major, career choices, and writing styles. They all concentrated on 16 different types and implemented some unsupervised or supervised machine learning algorithms. However, we can wondering can we just predict users into two types? According to the dataset, users are partitioned into 16 types. Different posts per user represent their MBTI characterize. We want to know which personality will effect person's post most. In this paper, we assume that type "Judging" and type "Perceiving" have more influence on users' posts. So, we predict MBTI personality into "Judging" and "Perceiving" using unsupervised and supervised

machine learning algorithms.

References

[1] Nicole J. Borges and Mark L. Savickas. Personality and medical specialty choice: A literature review and integration. *Journal of Career Assessment*, 10(3):362–380, 2002.

[2] The Myers & Briggs Foundation. The myers & briggs foundation, 2013. [Online; accessed 5-November-2019].

[3] C. G. Jung. *The collected works of C.G. Jung*. Pantheon Books, New York, 1953.

[4] S. Khalid, T. Khalil, and S. Nasreen. A survey of feature selection and feature extraction techniques in machine learning. In *2014 Science and Information Conference*, pages 372–378, Aug 2014.

[5] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. In *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pages 3–24, Amsterdam, The Netherlands, The Netherlands, 2007. IOS Press.

[6] Anthony Ma. Neural networks in predicting myers brigg personality type from writing style, 2017.

[7] Ronald Markert, Paul Rodenhauser, Mariam El-Baghdadi, Kornelija Juskaite, Alexander Hillel, and Bradley Maron. Personality as a prognostic factor for specialty choice: A prospective study of 4 medical school classes. *Medscape journal of medicine*, 10:49, 02 2008.

[8] Ronald Markert, Paul Rodenhauser, Mariam El-Baghdadi, Kornelija Juskaite, Alexander Hillel, and Bradley Maron. Personality as a prognostic factor for specialty choice: A prospective study of 4 medical school classes. *Medscape journal of medicine*, 10:49, 02 2008.

[9] Syed Imran Mehmood, Muhammad Abid Khan, Kieran M. Walsh, and Jan C.C. Borleffs. Personality types and specialist choices in medical students. *Medical Teacher*, 35(1):63–68, 2013. PMID: 23134199.

[10] Nancy Stilwell, Mollie Wallick, Sara Thal, and Joseph Burleson. Myers-briggs type and medical specialty choice: A new look at an old question. *Teaching and learning in medicine*, 12:14–20, 02 2000.

[11] M. Usama, J. Qadir, A. Raza, H. Arif, K. A. Yau, Y. Elkhatib, A. Hussain, and A. Al-Fuqaha. Unsupervised machine learning for networking: Techniques, applications and research challenges. *IEEE Access*, 7:65579–65615, 2019.