Myers-Briggs Personality Type Prediction

Prakhar Kumar MT21066 **Amit Kumar Singh** MT21008 **Maansi Vyas** MT21045

Abstract

Personality refers to the qualities of an individual that separates him from other individuals. Personality Prediction has always been a challenging task as the traditional methods for personality prediction do not always work. This is because conventional methods require input from the individual. This is the main problem as the answers provided by the individual may not map to his actual ideology or thinking.
To solve the above issues in this project, we use textual data from statements made by the user online to predict their way of thinking better. Using online statements, we can better predict personalities as individuals know that they can remain hidden behind the Internet's virtual wall. Online they can remain anonymous; hence they will, without hesitation, write about what they actually think.

1 Why we chose this Project

Personality prediction is an essential task in today's world as this can help individuals better understand the person they are dealing with and make more informed decisions.
Companies can use the personality data to improve their marketing strategy. They can target a specific personality type with customized advertisements. Human Resource Managers can use it to better assign new hires to a team with matching personality traits, and they can also use this data to improve the company's culture so that it fits everyone's needs.
Personality information can also be used by doctors to better treat patients with mental health issues; knowing the patient's personality would make it possible to create a better environment for individuals suffering from depression and other mental problems.

2 Existing Literature

This section summarises some of the relevant work that has been done in the personality prediction task.
In paper [1], the data is pre-processed, removing the stopwords (a, the, is etc), and the text is then lemmatized before converting them to TF-IDF features. The words used to TF-IDF are those which appear in 10% to 70% of the posts. Then, since an MBTI type consists of 4 binary types(MBTI indicators), an MBTI type is broken down into four sub-binary types. Four XgBoost classifiers from sklearn are trained to classify between the four binary classes. All four classifiers are then used to predict the MBTI type for a new sentence.
In paper [2],Preprocessing is similar to paper [1] .Tokenization of text is done next, and the token strings are padded to be used with RNN. The paper describes using of RNN because of the sequential nature of textual data. The authors chose confusion matrix and accuracy to measure the performance of the model trained.
In paper [3], The word2vec technique is used to get the numerical representation of sentences. Four Machine Learning models, Random Forest, Logistic Regression, KNN and Support Vector Machine (SVM), are built, and as an addition to this, more intense work is done by training all Machine Learning models with MBTI personality types(Extroversion–Introversion (E–I) column, Sensation–Intuition (S–N) column, Thinking– Feeling (T–F) column, and Judgment–Perception (J–P) column).
In Paper [4], the pre-processing step is in three phases: data deduplication, named entity identification, and lexicon normalization. The data

deduplication phase involves removing duplicate and redundant text from the dataset by employing a simhash algorithm. Then named entity identification phase recognizes and deletes unreliable useless entities and keywords from the dataset. In the last phase, word normalization methods are applied to convert phrases into their stem/lemma form.
After pre-processing, they used named entity recognition and sentiment analysis to add additional features to the pre-processed dataset. The TF-IDF vectorizer is then used to convert the derived important features into actual values and the word2vec embedding technique to transform the extracted feature into a document term matrix representation. Three ensemble learning methods have been used for the classification part: Boosting, Bagging, and Stacking models.
In paper [5], three machine learning models are designed: primary model, supporting model, and a clinical test model, which is based on MBTI. The clinical test model is designed to give an MBTI type as an output for the input provided from the supporting model. For data pre-processing in the clinical test model, the TF-IDF weighting scheme is used. Further, since the dataset is imbalanced so for this under sampling is done. A machine learning model for MBTI is built, and it is trained on multiple posts per user.
In paper [6], the MBTI classification problem involves two common approaches in supervised machine learning. First, one a multiclass classification into 16 classes. The second divides the problems into four binary classification problems for all four personality indicators.For preprocessing embedding vectors are obtained.Two models are trained Bi-LSTM and CNN architectures. After training model is evaluated using F1-score.

3 Dataset information

The dataset for MBTI personality types is freely available on Kaggle - Dataset Link.
Dataset has 2 columns and 8675 rows.
First column is for the MBTI type, for example - INTJ. An MBTI type is made up of 4 MBTI indicators. In the example shown, I, N, T, J are indicators. We have two choices for each indicator; I or E for first, N or S for second, T or F for third, and J or P for fourth. So in total, there can be 16 MBTI types. More info on what these indicators means is available in this link: More Info.
Second column consists of sentences by an individual with a given MBTI type. Each row contains multiple sentences separated by three pipe characters (|||).

4 Data Preprocessing

Here we first remove the URLs in the textual data. Then we removed punctuation in the text and replaced it with ". Then we converted the text into lower case. Some posts in the data also included the MBTI tag itself so we removed those words as well. Then we lemmatized each word in the posts and removed words which were stopwords or had length $<= 2$. Textual data is converted to numerical features using TFIDF and choosing only those words which occur more than 50 times and do not occur more than 80% of the time in whole corpus.

5 Data Analysis

We first see that the data is imbalanced . Most number of users for whom posts data is given belong to INFP, INFJ, INTJ, INTP personality types while ESFJ, ESFP, ESTJ personality types have least number of users.
We also wanted to see what kind of words are prevalent for different class labels, so we created a word cloud for each class label. For some classes, the word cloud is shown below. In the word clouds, it was observed that words like know, think, one, people, really, etc., are very common words for every class.

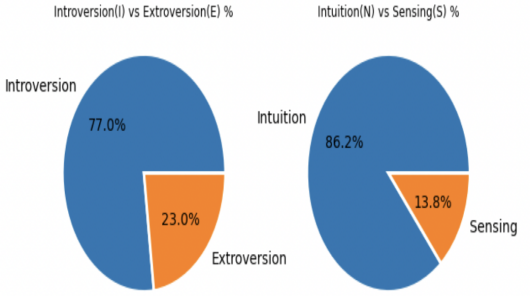
We also plotted distribution of individual indicators of MBTI type.

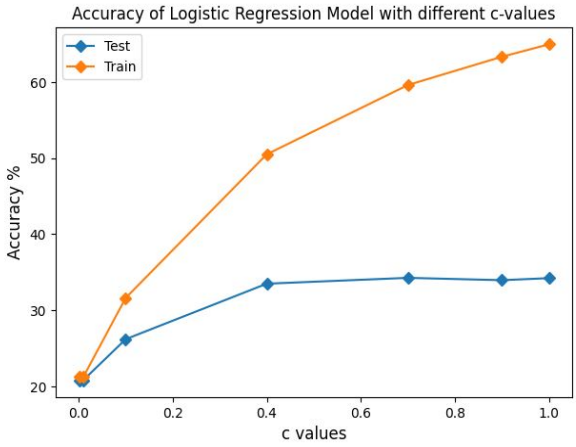
There a very few posts which have S as an indicator in MBTI type. A lot of posts have I as an indicator in MBTI type.

6 Baseline ML Models

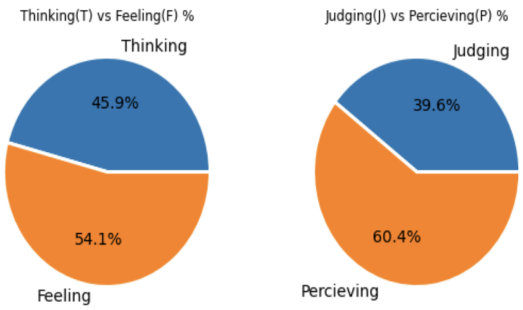
6.1 Logistic Regression

Logistic Regression is a Linear Model which tries to find a line(in 2D, a hyperplane in n-D) to classify between the two classes. For multiple classes, the one vs. rest strategy is used. We hyperparameter tuned the C value, and the maximum accuracy that we got on test data was 34.22%. The chart below shows train and test accuracy vs. C value (inverse of regularization strength).

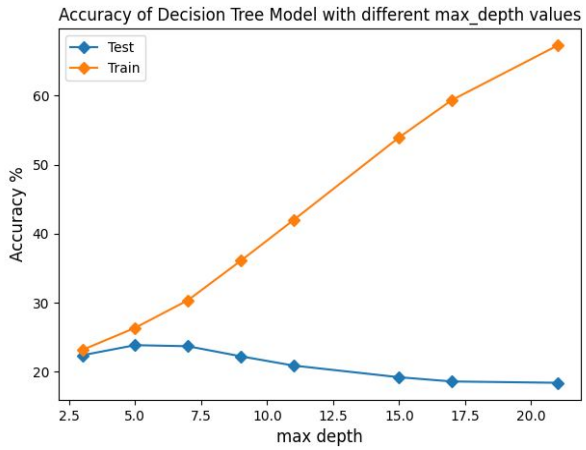




6.2 Decision Tree



Decision Tree is another model that uses a greedy heuristic to create a tree where every node judges the input sample on an attribute. Based on the comparison result, we move to another node in the tree to again compare on another attribute so on. We hyperparameter tuned the max depth parameter of the decision tree, and the maximum accuracy we got on test data was 23.81%. The chart below shows train and test accuracy vs. max depth parameter.



7 Improving Baselines

In our baseline models we were directly trying to classify between the 16 classes. This problem is very hard as the dataset is highly imbalanced. Now instead of using 16 classes directly we now break down the MBTI type into the indicators and train models on indicators. i.e IvsE, NvsS, TvsF, JvsP. So to predict an MBTI type we now have four models each predicting an MBTI indicator.

8 Final Models

8.1 Logistic Regression

Hyperparameter tuning was performed and for TFIDF best results were obtained with C=10. For word2Vec also C=10 gave best results. Same for Glove vectorization. F1 Scores are shown in table for all indicator models.

	TFIDF	Word2Vec	Glove
IvsE	0.85	0.87	0.87
NvsS	0.91	0.93	0.93
TvsF	0.71	0.72	0.73
JvsP	0.46	0.38	0.43

8.2 Decision Tree

Performed hyperparameter tuning on max_depth parameter, for TFIDF best results were obtained with max_depth = 3. For word2Vec and Glove vectorization also max_depth = 3 gave best results.Obtained F1 Scores are shown in table for all indicator models.

	TFIDF	Word2Vec	Glove
IvsE	0.86	0.87	0.87
NvsS	0.92	0.92	0.92
TvsF	0.62	0.63	0.61
JvsP	0.24	0.41	0.34

8.3 Random Forest

Performed hyperparameter tuning on n_estimators parameter, for TFIDF best results were obtained with n_estimators = 250. For word2Vec also n_estimators = 250 gave best results, for Glove vectorization n_estimators = 200 was optimal value.Obtained F1 Scores are shown in table for all indicator models.

	TFIDF	Word2Vec	Glove
IvsE	0.87	0.87	0.87
NvsS	0.93	0.92	0.92
TvsF	0.66	0.74	0.70
JvsP	0.10	0.34	0.31

8.4 SVC

Performed hyperparameter tuning on C parameter, for TFIDF best results were obtained with C=1.Similarly, for word2Vec and Glove vectorization C = 1 gave best results.Obtained F1 Scores are shown in table for all indicator models.

	TFIDF	Word2Vec	Glove
IvsE	0.87	0.87	0.87
NvsS	0.93	0.93	0.93
TvsF	0.72	0.72	0.72
JvsP	0.39	0.01	0.01

8.5 XGBoost

Performed hyperparameter tuning on n_estimators parameter, for TFIDF best results were obtained with n_estimators = 200. For word2Vec n_estimators = 150 gave best results, for Glove vectorization n_estimators = 250 was optimal value .Obtained F1 Scores are shown in table for all indicator models.

	TFIDF	Word2Vec	Glove
IvsE	0.86	0.86	0.86
NvsS	0.92	0.92	0.92
TvsF	0.69	0.72	0.71
JvsP	0.41	0.47	0.46

9 Conclusion

It is a very challenging problem as the dataset is highly imbalanced. Classifying between all 16 classes never really gave good results in our approach. So instead of trying to get all correct we changed the problem to 4 classification problems (predicting each indicator separately). We are able to get 2 to 3 indicators correct most of the time which is quite good given how imbalanced the data is and how similar the indicators seem.

10 References

1. https://www.mdpi.com/2414-4088/4/1/9/pdf
2. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/reports/6839354.pdf
3. https://thesai.org/Downloads/Volume11No11/Paper_25-Improving_Intelligent_Personality_Prediction.pdf
4. https://ijcrt.org/papers/IJCRT2106344.pdf
5. https://ieeexplore.ieee.org/document/9498486
6. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9578983