Myers Briggs Type Indicator(MBTI) Prediction Using Machine Learning

Dhananjay Singh, Mehul Gupta, Naini Narama Lnu, Nivedita Madhekar

Abstract - Personality is the combination of behaviour, emotion, and thought patterns that define an individual. The study on individual personality types began with Ancient Greek Physician Hippocrates theory of humorism, which argued that personality traits are based on four separate temperaments associated with four fluids (humours) of the body. More recently, Myers Briggs Type Indicator(MBTI) is considered one of the most popular and reliable methods to predict anyone's personality. The primary purpose of our project is to employ Natural Language Processing techniques and build Machine Learning models to predict personality traits based on users' social media posts. The main motivation behind this project is to show the potential of a classifier to be accurate enough when predicting people's personality types. We worked on the MBTI-labelled dataset and used supervised learning algorithms like Logistic Regression, Random Forest, Xgb, Support Vector Machines, Naive Bayes, and Decision Trees to make predictions. Ensemble methods like the Extreme Gradient Boosted tree model gave the best overall average accuracy of 76.9%.

Keywords - Supervised, Machine Learning, Classifiers, Extreme Gradient Boosted, Support Vector Machine, Myers Briggs Type Indicator(MBTI), Ensemble, Logistic Regression

1. Introduction

Derived from the Latin word Persona, personality means describing the behaviour or character of an individual. However, researchers and scientists have various definitions for personality; according to Hall and Lindzey [1], personality is “the dynamic organisation within the individual of those psychological systems that determine his characteristic behaviour and thought.” This system determines the unique way in which an individual adapts to an environment.

Various types of personality models are used to characterise personalities, such as the “Big Five Personality Traits Model”, “VIA Classification of Character Strengths”, “Myers-Briggs Type Indicator model”, and “Jung’s Theory of Personality type models”.

We will use a crucial test known as Myers-Briggs Type Indicator (MBTI) for our research. It is an introspective self-report questionnaire examining how people perceive and make decisions. The test has received considerable attention and use in various settings because it has broader applications in various disciplines. MBTI divides everyone into 16 distinct personality types across 4 axes -

1. Introversion(I) or Extraversion(E)
2. Sensing(S) or Intuition(N)
3. Thinking(T) or Feeling(F)
4. Judging(J) or Perceiving(P)

Predicting personality traits using MBTI is used in many businesses and research. Companies use it to analyse job applicants, managers use it to determine which employees might get along with one another, and friends might use it to tell the world what kind of person they are.

Each person is said to have one preferred quality from each category, producing 16 kinds. A person who prefers introversion, intuition, thinking, and perceiving would be labelled as INTP in MBTI. Many personality-based components would model or describe the person's preferences or behaviour.

However, a question comes to light, what is the best way to obtain the data for predictions. The understated answer is that Social media platforms provide a space for everyone to share posts, images, videos, etc. These posts represent the emotions, feelings, and opinions of users. Several times, precedent research on analysing these posts and status to predict personality traits has been done. This motivates us to take our dataset from social media platforms. Furthermore, we are taking this idea to manipulate our research to produce better results.

Predictive analysis using Machine learning is standard in personality topology. These algorithms have benefited businesses and recruiters in choosing the best candidates. We attempt to produce a machine learning algorithm that can determine a person's personality type via social media posts. We have utilised Natural Language Processing(NLP) to work with textual data and perform Data Preprocessing/Cleaning. Then we have performed Exploratory Data Analysis along with Feature Engineering to create a TF-IDF representation of the dataset. We have built classification models to predict multiclass as well as binary class labels, and evaluated performance based on accuracy, confusion matrix, classification reports, and AUC-ROC curves.

2. Related Work

Personality prediction using machine learning has been trendy amongst researchers. Over the past 40 years, various studies have proven the MBTI indicator to be a reliable and valid instrument for predicting personality traits. Carl G. Jung first introduced the theory in the 1920s, also known as Jungian Typology, based on cognitive functions. It grew out to systematise archetypal personality traits used in clinical practice. However, the MBTI instrument was developed in the 1940s by Isabel Briggs Myers. Much work has been done on the MBTI instrument using various parameters. We have looked at some of the research for our motivation.

There are many fields where the MBTI personality test is used, and much research has been conducted. Studies were done by Tricia Varvel, Stephanie G. Adams, and Shelby J Pridie on "The Effect of the Myers-Briggs type indicator on Team Effectiveness". They sought to find a way to make teams more effective by considering and utilising each team member's psychological type information[2]. Another study scrutinises the General Weighted Average(GWA) for Nursing Students based on the MBTI indicator; the research concludes that GWA was higher in Introversion than Extroversion, Sensing than Intuition, Judging compared to Perceiving. Also, it mentioned the top five personality types that have high GWA[3]. Furthermore, studies conducted by William L. Gardner and Mark J. Martinko to study Managers based on MBTI prove the indicator's validity and reliability to identify these relationships(managerial personalities, cognitions, behaviours, effectiveness and relevant variables)[4].

Conversely, the personality system widespread in Psychometrics is the Big Five Personality classification system. The five categories are Extraversion, Agreeableness, Openness, Conscientiousness, and Neuroticism. Moreover, the Big five test considers a few features in an individual's life like income, education, and marital status to be stable for a lifetime which is not the case in the MBTI personality test. Hernandez, Rayne and Knight, Ian Scott discuss using deep feed-forward neural networks to predict MBTI personality types for relatively small textual datasets[5].

There is a prevailing interest in predicting personality traits using social media posts among researchers in the Natural Language Processing field. There is a bigger picture to predicting these posts; organisations use it to their advantage and help them make more intuitive dating applications or websites, or they can use the information for different marketing strategies. Researchers(Amirhosseini and Kazemian) use Neuro-Linguistic Programming (NLP) to predict personality traits, and their studies involve the MBTI instrument based on Gradient Boost Algorithm in Machine Learning. Packages like NLTK (used for natural language processing) and XGBoost (optimised distributed gradient boosting library used to implement machine learning algorithms under the gradient boosting framework) are two packages amongst the many used to conduct this study. In conclusion, the dataset was run on two algorithms, the better of the two being the XGBoost algorithm, predicting with more accuracy[6].

An interesting question arises when discussing the instruments to predict personality traits, i.e., which instrument gives out the more accurate result? Or which test is more reliable?

Although there is not enough evidence or studies in identifying to determine the best indicator, Celli and Lepri compare two crucial indicators; the Big Five and MBTI. Using the SVM algorithm on both the annotated dataset (MBTI and Big Five), researchers conclude that SVMs have higher performance in predicting MBTI classes to Big Five. Thus, algorithms trained on MBTI could have better accuracy than the Big Five, albeit the Big Five is much more informative and has tremendous variability in performance[7].

Another intriguing approach is to use Deep Learning Architecture like MLP, LSTM, GRU, and CNN for training the dataset and implementation. Tandera et al. (2017) inculcated this idea in their study that applying deep learning architecture can ameliorate the accuracy of some of the algorithms[8].

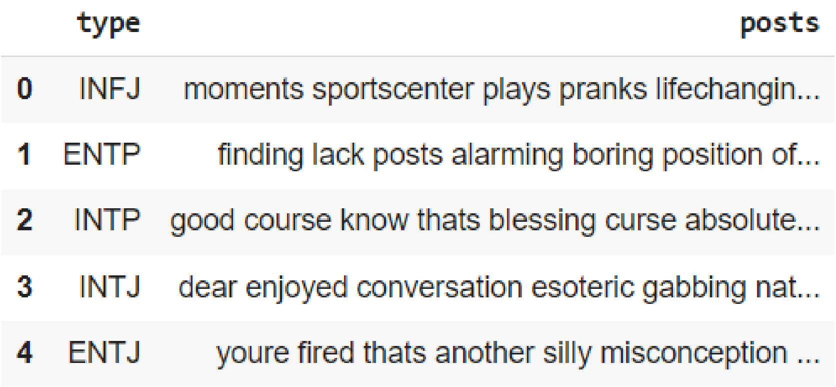
3. Methods

Data Preprocessing/Data Cleaning -

We have followed the data preprocessing/data cleaning steps necessary to clean textual data. Posts contain much information that needs to be removed from the dataset for better training. The steps followed to perform data cleaning on our dataset are as follows -

- Removing stopwords
- Removing hyperlinks/URLs
- Removing punctuation marks
- Removing non-English words/characters
- Removing very long/short words
- Removing MBTI personality codes
- Converting posts to lowercase

A snippet of the cleaned dataset is shown below-



	type	posts
0	INFJ	moments sportscenter plays pranks lifechangin...
1	ENTP	finding lack posts alarming boring position of...
2	INTP	good course know thats blessing curse absolute...
3	INTJ	dear enjoyed conversation esoteric gabbing nat...
4	ENTJ	youre fired thats another silly misconception ...

Feature Engineering -

The steps followed to perform feature engineering are as follows -

- We used Countvectorizer to create a sparse matrix consisting of token counts and build a vocabulary of known words. This returned us a document-term matrix.
- Learned the idf vector using fit and then transformed this count matrix into a tf-idf matrix using TfidfTransformer.
- We utilised the Label Encoder of the Sklearn library to convert our categorical class labels to a value ranging from 0 to (number of classes - 1). So the labels created in our case range from 0 to 15 for 16 unique MBTI codes.

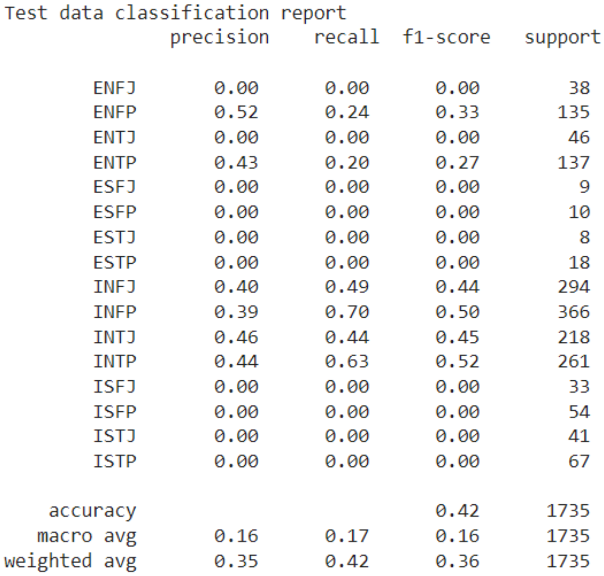
We split the dataset into training and testing datasets and then utilised machine learning models provided in the Sklearn library. We have implemented two baseline models - Decision Trees and Random forest, and two models that we learned in class - Logistic Regression and Multinomial Naive Bayes.

3.1 Models - Multiclass Classification

1. Logistic Regression

We used a Logistic Regression classifier to perform multiclass classification. We performed the classification using multi_class argument as ‘multinomial’ and solver as ‘lbfgs’. The multinomial logistic regression model will be fit using cross-entropy loss and predict the integer value for each integer encoded class label. We used the L2 regularisation penalty parameter, along with the C value of 0.5 to provide moderate regularisation. Another reason why we chose this model is that multinomial logistic regression can predict calibrated probabilities across all known class labels in the dataset.

Results from our classification report gave us precision, recall, F1-score, and accuracy for our model as displayed below. Precision tells us what percent of our predictions were correct, recall tells us what percent of positive cases we catch, and F1 score tells us what percent of positive predictions were correct.



```
Test data classification report
              precision    recall  f1-score   support

        ENFJ       0.00      0.00      0.00        38
        ENFP       0.52      0.24      0.33       135
        ENTJ       0.00      0.00      0.00        46
        ENTP       0.43      0.20      0.27       137
        ESFJ       0.00      0.00      0.00         9
        ESFP       0.00      0.00      0.00        10
        ESTJ       0.00      0.00      0.00         8
        ESTP       0.00      0.00      0.00        18
        INFJ       0.40      0.49      0.44       294
        INFP       0.39      0.70      0.50       366
        INTJ       0.46      0.44      0.45       218
        INTP       0.44      0.63      0.52       261
        ISFJ       0.00      0.00      0.00        33
        ISFP       0.00      0.00      0.00        54
        ISTJ       0.00      0.00      0.00        41
        ISTP       0.00      0.00      0.00        67

    accuracy                           0.42      1735
   macro avg       0.16      0.17      0.16      1735
weighted avg       0.35      0.42      0.36      1735
```

Interpreting these values on our test dataset, we found out that our model could not predict many of the class labels such as ENFJ, ESFJ, ESFP, ESTJ, ESTP, ISFJ, ISFP, ISTJ, ISTP. We got precision, recall, and F1-score for all these classes as 0, which led to the accuracy of our model to 41%. Results from this classifier are shown below -
Training Accuracy = 58%
Testing Accuracy = 41%

2. **Decision Tree Classifier**

Decision Tree is a non-parametric supervised learning method used for classification. They are built by a process known as binary recursive partitioning. This is an iterative process of splitting the data into partitions, and then splitting it up further on each of the branches. First we select the root node which is the best attribute to split the records. This attribute will become a decision node and splitting of the tree will take place based on the splitting rule. The tree starts to build

recursively in this way until a leaf node is reached which contains the output. Attribute selection measure used in decision tree classifier is entropy, gini index, and information gain ratio.
Entropy is the measure of impurity in the dataset. Information is the opposite of entropy. It measures the difference between entropy before and after the split.

$$\text{Info(D)= - } \sum_{i=1}^{m} pi \log_2 pi$$

In the formula pi the probability of any instance in the dataset belonging to class i.

$$\text{Info}_A\text{(D)=}\sum_{j=1}^{V} \frac{|Dj|}{|D|} \text{ X Info(D}_j\text{)}$$

$$\text{Gain(A)=Info(D)- Info}_A\text{(D)}$$

Gini index uses squared proportions of classes to decide the splitting criteria.

The algorithm works as *"1 — (P(class1)^2 + P(class2)^2 + … + P(classN)^2)"*

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

We created a Decision Tree classifier and experimented with its parameters to find the best model. We found out that the model performed best with max_depth = 10. Increasing other parameters like min_samples_leaf and min_samples_split were leading to underfitting. We used Gini index criteria for the splitting of the nodes. Decision trees are a compelling classification model, and they can be extended to multiclass classification by giving a unique integer to each of the class labels. According to the criteria specified, the tree splits, and the leaf nodes lead to different class labels. The classification report was similar to the rest of the models. Results from this classifier are shown below -
Training Accuracy = 52%
Testing Accuracy = 35%

3. Random Forest Classifier

This classifier fits several decision trees on various sub-samples of our dataset and uses averages to improve the predictive accuracy and control over-fitting. We performed hyperparameter tuning for this classifier as well. Random forest is chosen because they are robust classifiers that can generalise well. The Random Forest model performed better than Multinomial Naive Bayes and Decision Tree classifiers. The classification report suggested that the results were pretty much the same as the Logit model. Results from this classifier are shown below -
Training Accuracy = 66%

Testing Accuracy = 40%

4. Multinomial Naive Bayes

Given the predictor variables x1, x2, x3, … , xn, and class variable y, the Bayes theorem states that -

$$P(y|X) = \frac{P(X|y) * P(y)}{P(X)}$$

Using the chain rule, the likelihood P(X|y) can be decomposed into -

$$P(X|y) = P(x_1, x_2, \dots, x_n|y)$$

$$= P(x_1|x_2, \dots, x_n, y) * P(x_2|x_3, \dots, x_n, y) \dots P(x_n|y)$$

Naive Bayes algorithm works under the assumption that the conditional probabilities are independent of each other. So, by conditional independence we have the following equation -

$$P(y|X) = \frac{P(x_1|y)^* P(x_2|y) \dots\dots P(x_n|y) * P(y)}{P(x_1) * P(x_2) \dots P(x_n)}$$

Naive Bayes model take the posterior probabilities and uses the decision rule maximum a posteriori to reach the final equation -

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

The Multinomial Naive Bayes classifier uses discrete features (e.g., word counts for text classification). However, fractional counts such as tf-idf also work for this classifier, so we chose this model to classify the MBTI types.

When interpreting the results of the classification report, It could not predict more label types than the rest of the classifiers, resulting in precision, recall, and F1 score to be 0. These values were even worse for the class labels for which it made predictions, leading to the worst accuracy among all four classifiers implemented.
Results from this classifier are shown below -
Training Accuracy = 42%
Testing Accuracy = 32%

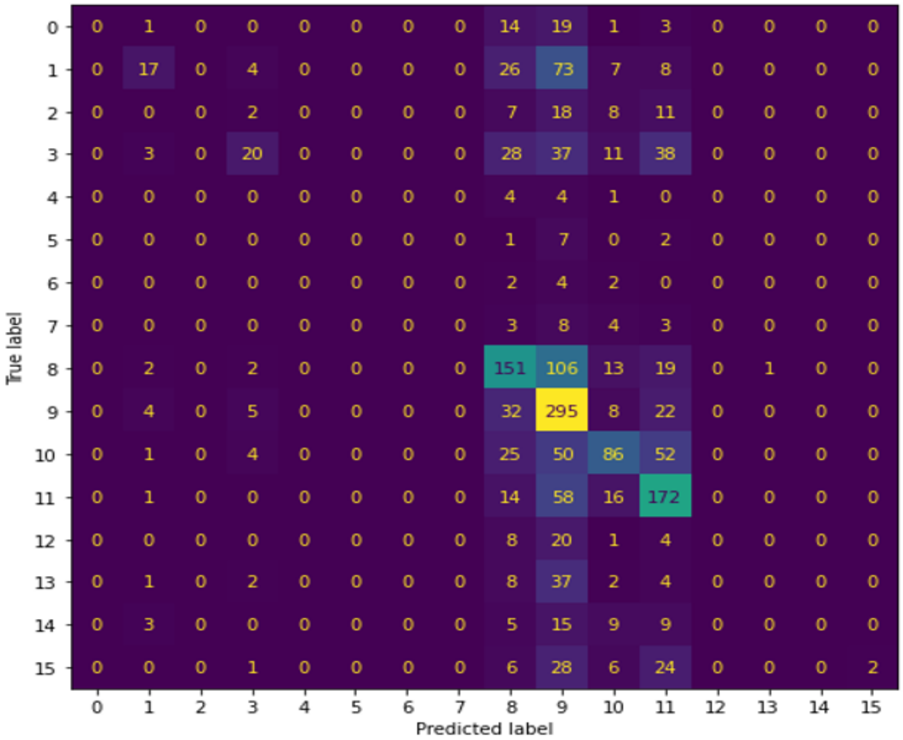
Hyperparameter Tuning for SVM and Logistic Regression -
We performed hyperparameter tuning on the SVM and Logit model to improve accuracy. The parameter grid, along with the best combination of parameters chosen for both the models is displayed below.

Models	Parameter Grid	Best Parameter
Support Vector Machine	`param_grid = {'C': [0.01,0.1, 1, 10, 100, 1000], 'gamma': [1, 0.1, 0.01, 0.001, 0.0001], 'kernel': ["linear","poly","rbf", "sigmoid"]}`	`{'C': 1, 'gamma': 1, 'kernel': 'rbf'}`
Logistic Regression	`param_grid = {'C': [0.001,10, 100,1.0,0.1,0.01],'solver':['newton-cg','lbfgs','liblinear','saga'],'penalty':['l2','l1','elasticnet']}`	`{C=1.0, penalty='l1', solver='saga'}`

Hyperparameter tuning did not improve accuracy. We came to the conclusion that this is because of class imbalance in the dataset. We now move on to converting our problem to a binary class classification to handle class imbalance and hence improve accuracy.

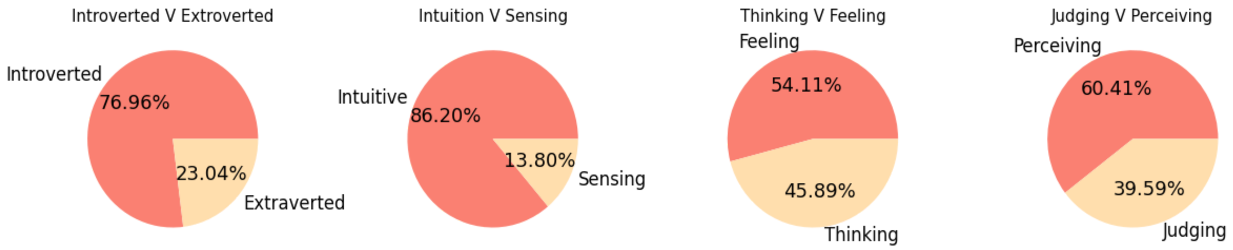
3.2 Models - Binary Classification

Motivation for binary classification - Multiclass Classification models resulted in very low accuracy scores. The models over-classified the majority class labels due to their higher prior probabilities. Imbalanced class distribution in the dataset contributed to this factor. As displayed in the confusion matrix, the minority class labels have a higher misclassification rate.



The 16 MBTI classes also had a lot of overlap between them and there was not any clear way to distinguish between them. For example, INFJ and INFP are treated as distinct classes even though they overlap in many aspects and only have a minor difference. These classes aren't actually independent, which thwarts a classifier that seeks to find complete separation. F1-score from the classification report for the minority class labels is 0. This score tells us what percent of positive predictions were correct. Models are unable to make any predictions on the minority class labels. This explains why we got such low accuracy for all classification models.

To improve the accuracy of models and also handle the class imbalance problem we have built classifiers across the 4 axis - Introversion (I)/Extroversion (E), Intuition (N)/Sensing (S), Thinking (T)/Feeling (F), Judging (J)/Perceiving (P) of MBTI. The label distribution is a lot more balanced as displayed below.



To achieve this classification problem, we binarized the MBTI label such that INFJ would correspond to [0 0 0 0]. This would lead to 4 binary class labels across the 4 axes of MBTI. We performed the same steps for feature engineering as before to create a tf-idf matrix consisting of

595 features., before implementing the machine learning models. The results of the binary classification models are explained in the Experimental Results section. We built SVM, and XGB models besides Logit and Random Forest to predict the binary class labels.

Logistic Regression - It is a statistical model used to predict a binary dependent variable with the help of a logistic function called the sigmoid function. The sigmoid function is described below -

$$F(x) = \frac{1}{1 + e^{-x}} = \frac{e^{x}}{e^{x} + 1}$$

This function helps the logistic regression model to squeeze the values of the unbounded linear equation. Consider a linear function p(x). We will assume log p(x) to be linear as described below -

$$\log \frac{p(x)}{1 - p(x)} = \alpha_0 + \alpha . x$$

Solving this for x will give the following -

$$p(x) = \frac{e^{\alpha_0 + \alpha}}{e^{\alpha_0 + \alpha} + 1}$$

We can assume that the probability of the predicted class is p(x) when y=1 or it is 1-p(x) when y=0. We can take the log likelihood function of this probability and substitute it with value of p(x) and then maximise it to get the following equation -

$$\frac{\partial l}{\partial \alpha_j} = \sum_{i=0}^{n} \big(y_i - p(x_i; \alpha_0, \alpha)\big) x_{ij}$$

We take the maximum of log likelihood because in case of logistic regression, gradient ascent is implemented.

The penalty parameter of logistic regression models, elastic net is a regularised regression method that linearly combines the L1 and L2 penalties of the lasso and ridge methods.

Random Forest - This is an ensemble learning algorithm that constructs multiple decision trees at the training time. Each decision tree gives out an output or class prediction and the class with the most number of votes becomes the model’s prediction. Random forest uses bagging and feature randomness at the time of building individual trees. In bagging, a random sample of data in a training set is selected with replacement—meaning that the individual data points can be chosen more than once.

Support Vector Machine - Support Vector Machines are used in classification problems when the classes are linearly or not linearly separable. The feature space is enlarged using kernels in order to accommodate a non-linear boundary between classes. When the degree of a kernel is greater than one, the decision boundary becomes more flexible.

The objective of SVM is to find a hyperplane with the maximum margin in an N-dimensional feature space that clearly classifies the data points. The greater the margin, more the confidence in classifying new data points in the future. Support vectors are those data points that are closest to the hyperplane and influence the position of the hyperplane. When the support vectors are removed, the margin also changes. In order to maximise the margin, the loss function for SVM (hinge loss) has to be minimised. Hinge loss function is given by -

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases}$$

If the predicted class and the actual class are of the same sign, then the loss function is 0. If they aren't the same, the loss value has to be calculated. A regularisation parameter is added to balance out the margin maximisation and the loss. The cost function with the regularisation parameter is -

$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+$$

In order to find the gradients, we have to find the partial derivatives with respect to the weights. These gradients can be used to update the weights -

$$\frac{\delta}{\delta w_k} \lambda \parallel w \parallel^2 \quad = 2\lambda w_k$$

$$\frac{\delta}{\delta w_k} \big(1 - y_i \langle x_i, w \rangle\big)_+ \quad = \begin{cases} 0, & \text{if } y_i \langle x_i, w \rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases}$$

<u>No misclassification:</u>

Only the gradient from the regularisation parameter has to be updated -

$$w = w - \alpha \cdot (2\lambda w)$$

<u>When there is misclassification:</u>

In order to find the new gradients, we use the loss with the regularisation parameter -

$$w = w + \alpha \cdot (y_i \cdot x_i - 2\lambda w)$$

Extreme Gradient Boosting - This is a decision-tree-based ensemble machine learning algorithm that uses a gradient boosting framework. The motivation to use the XGBoost model was to utilise the model's speed and performance. XGB uses decision trees as its base learners and trees are built using residuals instead of class labels. The pseudo code of XGBoost classifier is as follows -

1. Use residuals from initial prediction as target values to build the first tree.
2. Use residuals after combining the initial prediction with the first tree scaled by the learning rate as target values for the second tree.
3. Multiply the tree results with the learning rate and add to the previous prediction.
4. Repeat until the specified number of trees is reached or no further improvement is possible.

XGBoost uses similarity score and gain to build the individual trees by determining the best node splits.

$$Similarity\ Score = \frac{(\sum_{i=1}^{n} Residual_i)^2}{\sum_{i=1}^{n} [Previous\ Probability_i * (1 - Previous\ Probability_i)] + \lambda}$$

Residual is actual - predicted value.

Previous probability is the probability of an event calculated at the previous step. Initial probability is assumed to be 0.5.

Lambda is the regularisation parameter.

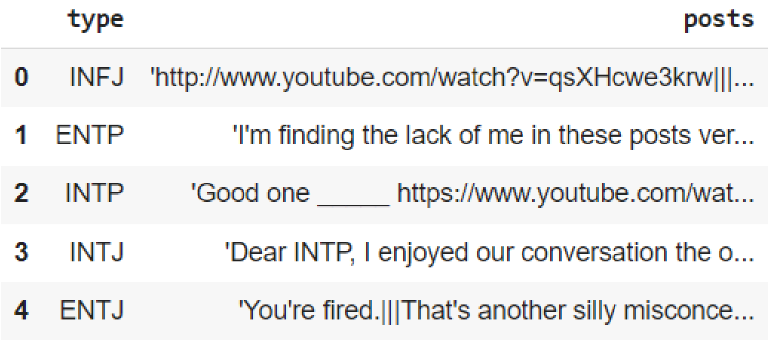
$$Gain = Left\ leaf_{similarity} + Right\ leaf_{similarity} - Root_{similarity}$$

Splitting criteria - The node split with the highest Gain is then chosen as the best split for the tree.

4. Experimental Results

Exploratory Data Analysis -

The dataset(url) consists of 8675 rows and 2 columns. The columns are "type", which is the four letter MBTI personality code, and "posts", which consists of posts that each user has posted. A snippet of the dataset prior to data preprocessing and cleaning -

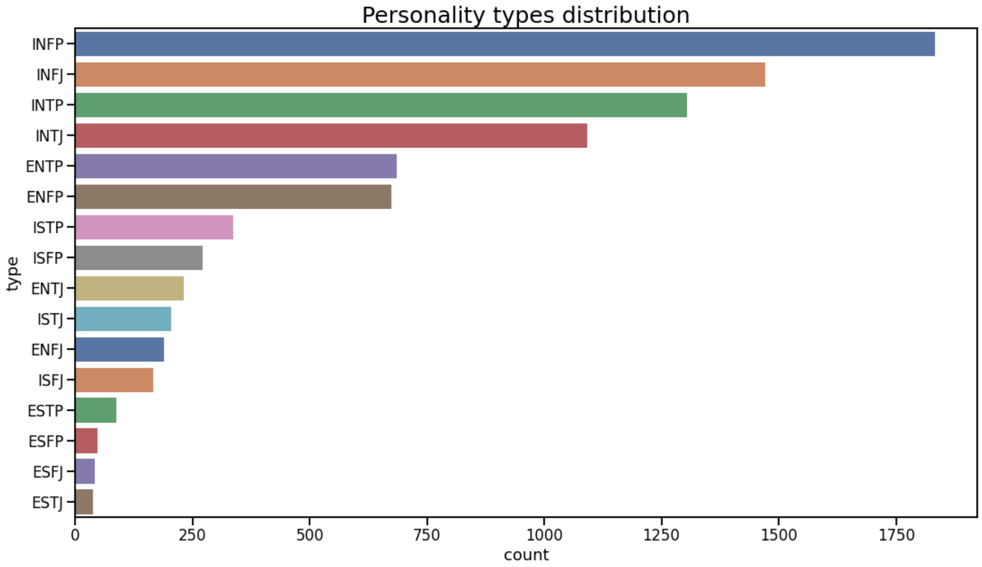


	type	posts
0	INFJ	'http://www.youtube.com/watch?v=qsXHcwe3krw|||...
1	ENTP	'I'm finding the lack of me in these posts ver...
2	INTP	'Good one _____ https://www.youtube.com/wat...
3	INTJ	'Dear INTP, I enjoyed our conversation the o...
4	ENTJ	'You're fired.|||That's another silly misconce...

Dataset Statistics are as follows -

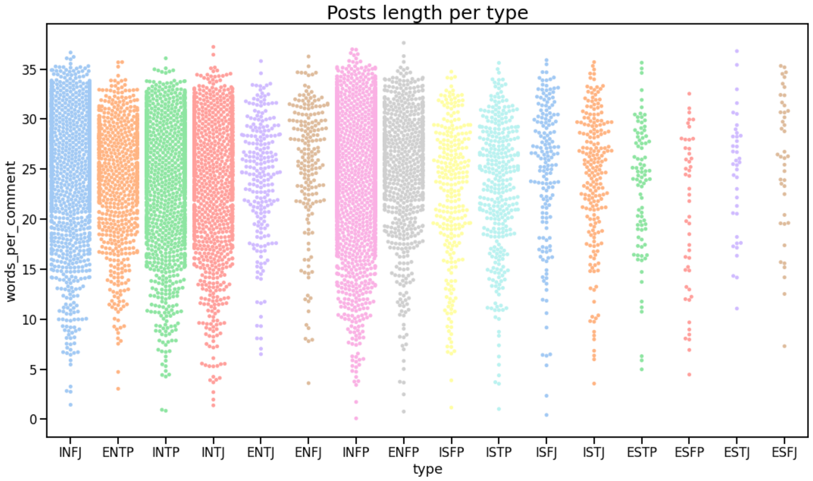
- ➔ There are no missing values in the dataset
- ➔ There are no duplicate values in the dataset
- ➔ The number of unique values in the dataset are - 16 types, 8675 posts.

The target label in our model is “types”. The label distribution is shown below -

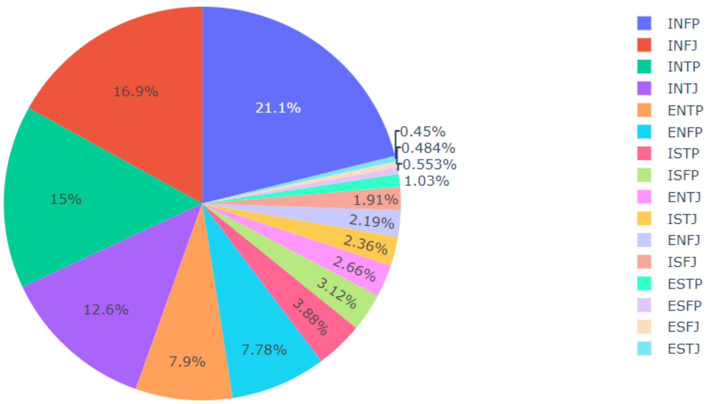


The data set is unbalanced throughout the different labels. The most common personality type in the dataset is INFP. This is probably because introverted, perceived users post most frequently on social media.

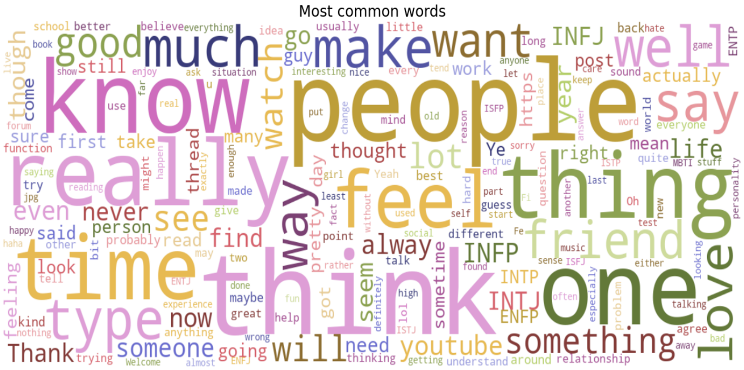
We plotted a graph to visualise the length of posts for each personality type and the words per post. This graph shows that the INFP type is the most cluttered, again reiterating the imbalance in the dataset.



A pie chart representing the proportion of each MBTI type is also visualised -



Next, we found the most common words in all the posts in the dataset, which were represented by a word cloud -

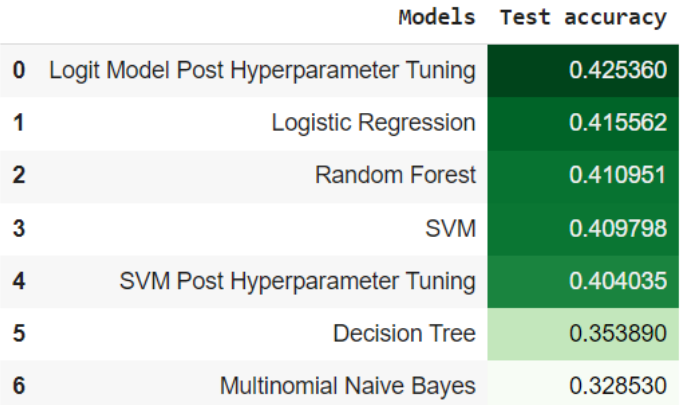


The most common words in the posts include - I, me, if, and MBTI types like INFP, ENFP, INTJ, etc. Common words like I, me, or other punctuations would not contribute to model training. While removing MBTI types is crucial to get better model accuracy on test data. These words would be removed as part of our data preprocessing and cleaning.

We would perform classification on our dataset to predict the class label “type”, using both multiclass as well as binary classification.

Results From Multiclass Classification Models -

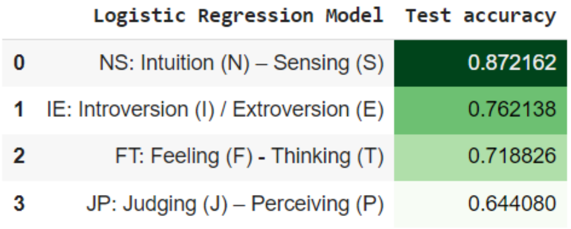
The accuracy achieved on test data from all machine learning models for multiclass classification are summarised in the below table -



	Models	Test accuracy
0	Logit Model Post Hyperparameter Tuning	0.425360
1	Logistic Regression	0.415562
2	Random Forest	0.410951
3	SVM	0.409798
4	SVM Post Hyperparameter Tuning	0.404035
5	Decision Tree	0.353890
6	Multinomial Naive Bayes	0.328530

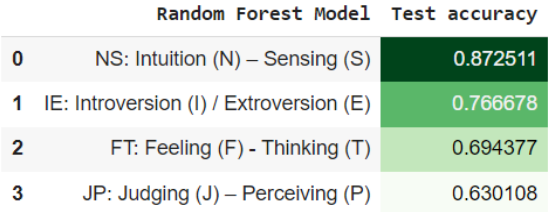
Results From Binary Class Classification Models -

Logistic Regression -



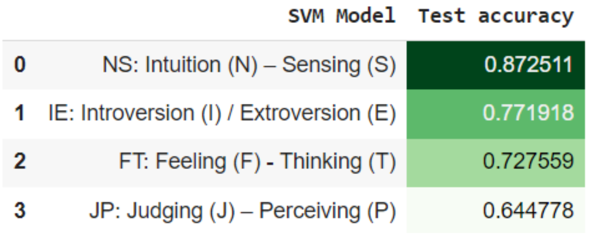
	Logistic Regression Model	Test accuracy
0	NS: Intuition (N) – Sensing (S)	0.872162
1	IE: Introversion (I) / Extroversion (E)	0.762138
2	FT: Feeling (F) - Thinking (T)	0.718826
3	JP: Judging (J) – Perceiving (P)	0.644080

Random Forest -



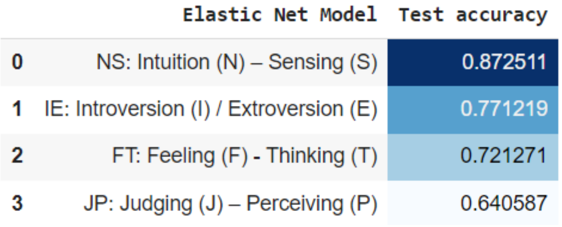
	Random Forest Model	Test accuracy
0	NS: Intuition (N) – Sensing (S)	0.872511
1	IE: Introversion (I) / Extroversion (E)	0.766678
2	FT: Feeling (F) - Thinking (T)	0.694377
3	JP: Judging (J) – Perceiving (P)	0.630108

SVM -



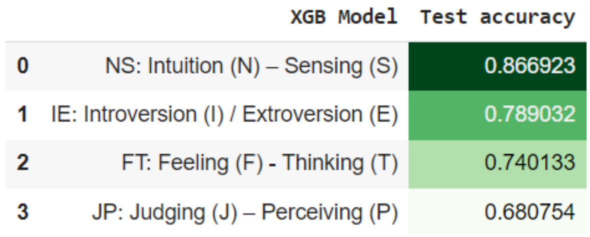
	SVM Model	Test accuracy
0	NS: Intuition (N) – Sensing (S)	0.872511
1	IE: Introversion (I) / Extroversion (E)	0.771918
2	FT: Feeling (F) - Thinking (T)	0.727559
3	JP: Judging (J) – Perceiving (P)	0.644778

Logit(Elastic Net Penalty) -



	Elastic Net Model	Test accuracy
0	NS: Intuition (N) – Sensing (S)	0.872511
1	IE: Introversion (I) / Extroversion (E)	0.771219
2	FT: Feeling (F) - Thinking (T)	0.721271
3	JP: Judging (J) – Perceiving (P)	0.640587

XGB -

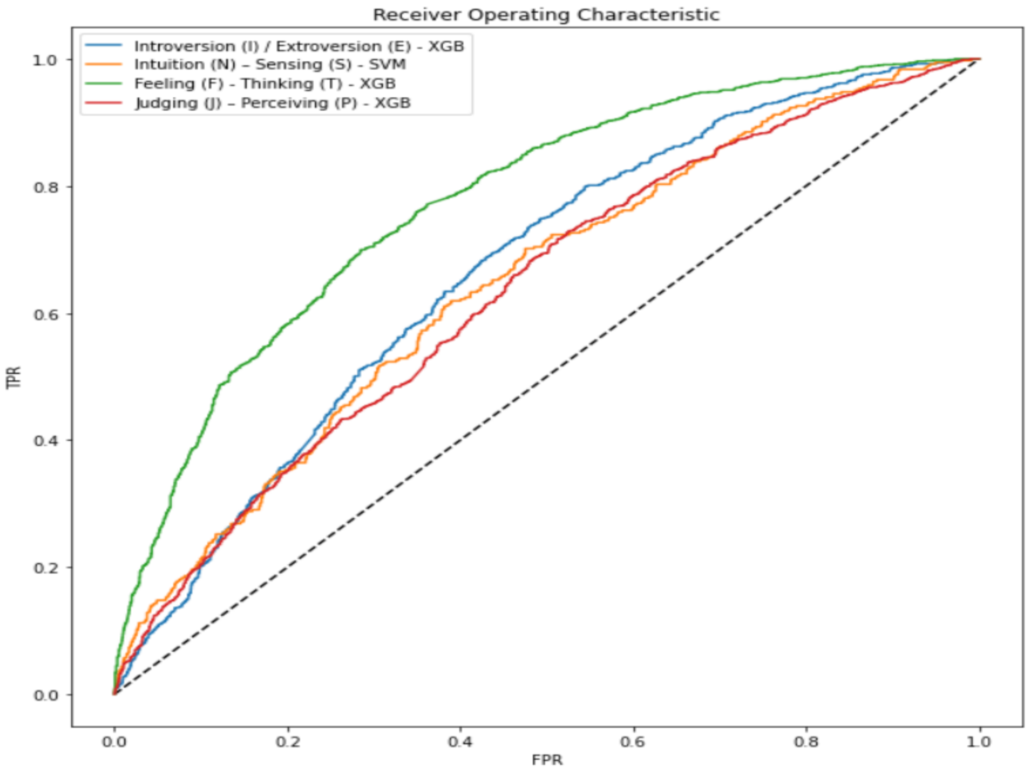


	XGB Model	Test accuracy
0	NS: Intuition (N) – Sensing (S)	0.866923
1	IE: Introversion (I) / Extroversion (E)	0.789032
2	FT: Feeling (F) - Thinking (T)	0.740133
3	JP: Judging (J) – Perceiving (P)	0.680754

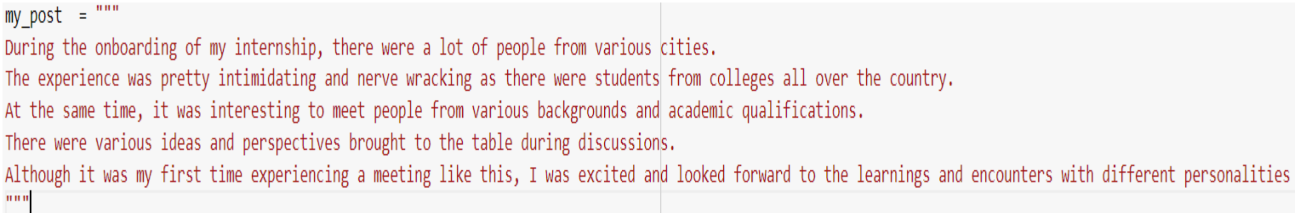
Summary Of The Model's Accuracy - XGB performed the best across 3 axes - Introversion (I)/Extroversion (E), Thinking (T)/Feeling (F), Judging (J)/Perceiving (P), and SVM model performed the best for Intuition (N)/Sensing (S) class label.

Model	Accuracy
XGB - Introversion (I)/Extroversion (E)	78.9%
XGB - Thinking (T)/Feeling (F)	74.01%
XGB - Judging (J)/Perceiving (P)	68.07%
SVM - Intuition (N)/Sensing (S)	87.25%

Combined AUC-ROC Curve For Top 4 CLassifiers -

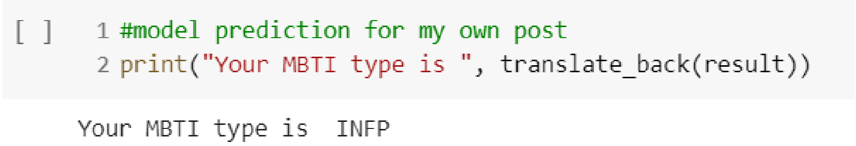


Predictions On User Defined Posts - We wrote 4 different posts with different sentiment connotations. We gave these posts as inputs to the best 4 models that we got when predicting the binary class labels. We translated back the result that our models gave to get the 4 letter MBTI code for the user defined posts. Example of one of the posts we gave is displayed below -



```
my_post  = """
During the onboarding of my internship, there were a lot of people from various cities.
The experience was pretty intimidating and nerve wracking as there were students from colleges all over the country.
At the same time, it was interesting to meet people from various backgrounds and academic qualifications.
There were various ideas and perspectives brought to the table during discussions.
Although it was my first time experiencing a meeting like this, I was excited and looked forward to the learnings and encounters with different personalities
"""
```

The result translated back by the XGB model gave the MBTI type for this post as “INFP”



```
#model prediction for my own post
print("Your MBTI type is ", translate_back(result))
```

```
Your MBTI type is  INFP
```

The MBTI type INFP was expected as well. The post contains introverted(I) terms like intimidating, and nerve racking. The post also shows that the person was feeling(F) excited for the onboarding process. The person is definitely intuitive(N) because they have made this impression about the internship without actually sensing it. There is no suggestion of judgement in the post which is why the model predicts perceiving(P).

Analysis -

1. **Multiclass Classifiers:**

All the classifier models were underfitting the dataset, even when we split the dataset in different ratios of training and test sets such as 70:30 or 80:20. Models were not learning enough from the training data, resulting in low generalisations and very poor predictions.

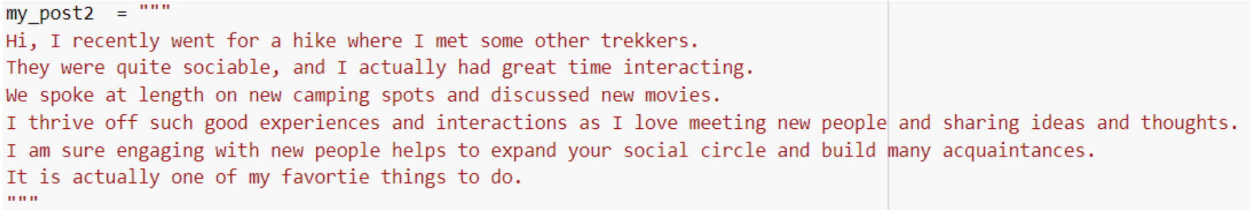
There is a massive imbalance in our dataset throughout the classes, which leads to the imbalance problem for all the classifiers. The prediction always diverges toward the sizable class, and the smaller classes are bypassed, clearly visible from the classification reports for the four classifiers. In our case, the dataset mainly consisted of labels such as INFP, INFJ, INTJ, INTP, ENTP, and ENFP. As a result of which, all the models were able to predict only for these mentioned MBTI codes, while for all other classes, there was no prediction.

The 16 MBTI classes also had much overlap between them, and there was no clear way to distinguish between them. For example, INFJ and INFP are treated as distinct classes even though they overlap in many aspects and only have a minor difference. These classes are not actually independent, which thwarts a classifier that seeks to find complete separation.

2. **Binary Classifiers:**

Creating 4 different classes across the MBTI axes helped achieve class balance. There was a jump in accuracy for all the classification models, with SVM and XGB performing the best. The AUC-ROC curve is plotted for these classifiers. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Based on the AUC-ROC curve the classifiers predicting Introversion (I)/Extroversion (E), Judging (J)/Perceiving (P), and Intuition (N)/Sensing (S) class labels have more type 1 and type 2 errors. The XGB model predicting Thinking (T)/Feeling (F) class label has the best predicting power to separate the two classes. This is due to the fact that out of all the 4 class labels, the Thinking (T)/Feeling (F) label is the most balanced.

3. **Predictions On User Defined Posts:**



```
my_post2  = """
Hi, I recently went for a hike where I met some other trekkers.
They were quite sociable, and I actually had great time interacting.
We spoke at length on new camping spots and discussed new movies.
I thrive off such good experiences and interactions as I love meeting new people and sharing ideas and thoughts.
I am sure engaging with new people helps to expand your social circle and build many acquaintances.
It is actually one of my favortie things to do.
"""
```

One of the posts written by us and passed as input to the model had an extroverted connotation but the model predicted introversion. One of the reasons that the XGB model predicting Introversion(I)/Extroversion(E) is unable to predict Extroversion easily is that this class has the highest class imbalance with Introversion being 80% and Extroversion being 20%. The training dataset might contain very few examples of Extroversion(E) class which is why it is unable to make generalisations for this class label. Model is unable to capture the relationship between input posts and the output class label(E). This XGB model has an accuracy of 78.9% for the (I)/(E) label. Inability of the model to learn well from training data and subsequently not able to predict class (E) labels might be the reason for lower accuracy.

5. CONCLUSION

This project was able to utilise the supervised machine learning algorithms to successfully predict personality using social media posts. Models did not particularly work well in case of multiclass classification due to class imbalance. However we were able to use NLTK library and classifiers like SVM and Extreme Gradient Boosting to improve the accuracy of predicting Introversion (I)/Extroversion (E), Intuition (N)/Sensing (S), Thinking (T)/Feeling (F), Judging (J)/Perceiving (P) class labels of MBTI. The XGBoost model gave the highest average accuracy of 76.9%.

Past researchers have pointed out that the dataset consists only of posts from one social media network called the Personality Cafe Forum. This cannot be generalised to users from another forum since the topic diversity is too narrow. Other social media posts could help improve accuracy and also even out the class imbalance. Future work for this project includes collecting more social media posts for all the MBTI labels but mostly related to extroversion to create a balanced dataset and achieve more generalisations. We also intend to utilise deep learning models to further improve accuracy and build the perfect personality indicator model.

References

[1] Hall, C.; Lindzey, G. Theories of Personality, 2nd ed.; Wiley: New York, NY, USA, 1970.
[2]https://www.researchgate.net/publication/266881677_A_Study_of_the_Effect_of_the_Myers-Briggs_Type_Indicator_on_Team_Effectiveness
[3]https://files.eric.ed.gov/fulltext/ED579286.pdf
[4]https://www.sciencedirect.com/science/article/pii/S0149206396900124
[5]https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/reports/6839354.pdf
[6]https://www.mdpi.com/2414-4088/4/1/9
[7]http://ceur-ws.org/Vol-2253/paper04.pdf
[8]https://www.sciencedirect.com/science/article/pii/S1877050917320537