Predicting Myer-Briggs Personality from Tweets

Davide Lupis,Federico Pozzoli

2022-05-31

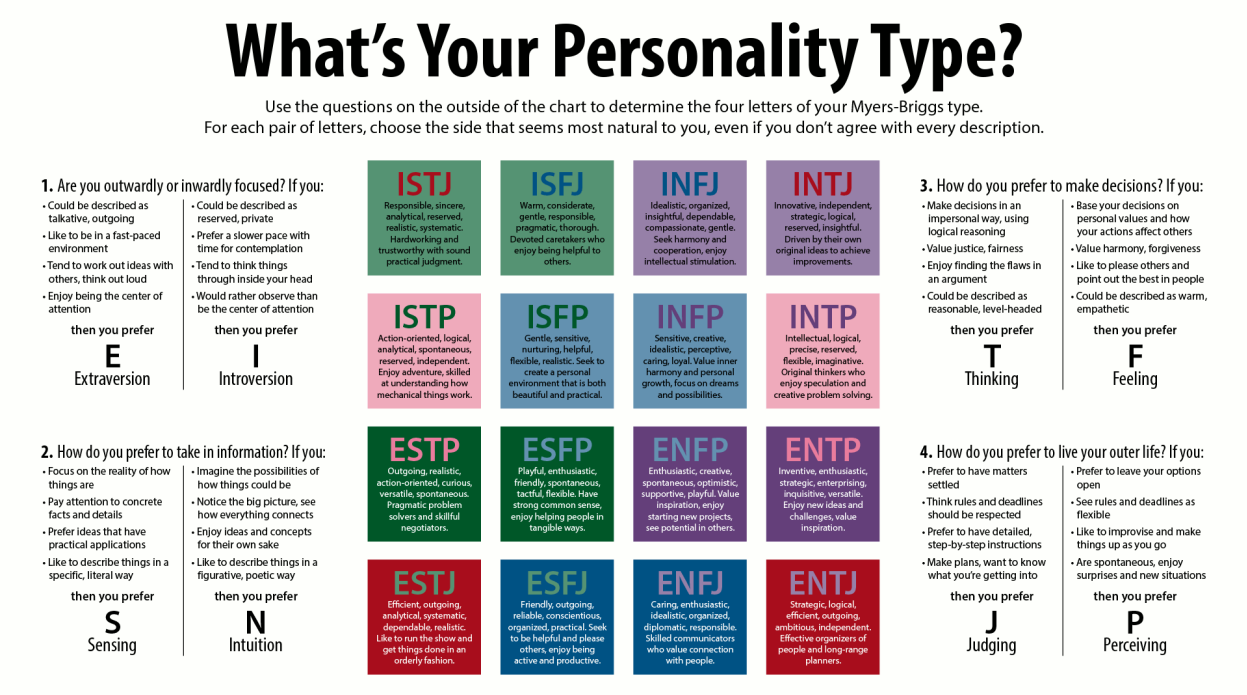

Figure 1: Description of Myer Briggs Indicators

Business Understanding

The Myers Briggs Type Indicator (or MBTI for short) is a "personality type system" that divides everyone into 16 distinct personality types across 4 axis:

- Introversion (I) – Extroversion (E)
- Intuition (N) – Sensing (S)
- Thinking (T) – Feeling (F)
- Judging (J) – Perceiving (P)

Even if our labels are not a solid ground truth we try to create a more feasible system in order to asses the type at large scale.

The use of a model is justified since:

- The test is quite long and a very small percantage of people have it done.
- It is not feasible to make humans check every potential user.
- Data Visualization techniques are not meaningful tools to make a separation between the classes.

The Final Goal is to determine if a model is capable of predicting the Myers Briggs Type Indicator based on this text input.

Data Understanding

The pre-processing steps are already given by the author of the dataset and include:

- removing hyperlinks/special.
- characters/stopwords.
- converting emojis to text.
- lemmatization, and stemming.

The total amount of data is 106067. Each row is a fixed block of words and a label.

There are not missing values and is possible to use a supervised learning approach with classification.

The data is unbalanced and the minority class have just 181 samples. It is assumed that the final label is composed by independent pseudo label and that the classification problem can be seen as :

- Multiclass Classification Problem with 16 labels
- Recursive Binary Classification problem with 4 pseudo binary label.

Note: More details about the Recursive Binary Classification in the Modeling part.

Table 1: Frequency of Myer Briggs Type for All the Dataset

ENFJ	ENFP	ENTJ	ENTP	ESFJ	ESFP	ESTJ	ESTP	INFJ	INFP	INTJ	INTP	ISFJ	ISFP	ISTJ	ISTP
1534	6167	2955	11725	181	360	482	1986	14963	12134	22427	24961	650	875	1243	3424

Data Preparation

According to a supervised learning approach we use as target the type column and as feature the TFIDF vector from the text. We explain the following decision and how it will affect the model.

A TFIDF vectorization is the process of transitioning from raw text into a set of numbers. It's a BAG OF WORDS approach that in its simplest form just create a feature for each word, defined by now as vocabulary, and it flags with a binary encoding whether or not the term is present in the document. Is a naive approach that can be expanded using weights, counting the number of times a term appears in the specific document and multiplying by a scaling factor that ensure high importance to words that are numerous inside the documents and less common among all documents. This final form is defined as Term Frequency Inverse Document Frequency and is the form used in this project.

Pro:

- It is fairly simple to understand and quite powerful
- Is can give a base for model interpretation if needed

Cons:

- It's a Bag of words strategy, therefore the model will not catch the "big picture" of the sentence and struggle in sentences with negative edge cases
- The size of the vector is a free parameter and it will affect the result, Hyperparameter tuning is required but skipped for computational reasons.

Recursive Approach

Create pseudo label

We separate from the very beginning 50 balanced observation per class, for a total of 800 rows that will be

used as text. As for the first basic approach we collect a balanced training sample of 100 observation per class, for a total of 1600.

Table 2: Frequency of Data for Training

ENFJ	ENFP	ENTJ	ENTP	ESFJ	ESFP	ESTJ	ESTP	INFJ	INFP	INTJ	INTP	ISFJ	ISFP	ISTJ	ISTP
100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Table 3: Frequency of Data for Testing

ENFJ	ENFP	ENTJ	ENTP	ESFJ	ESFP	ESTJ	ESTP	INFJ	INFP	INTJ	INTP	ISFJ	ISFP	ISTJ	ISTP
50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50

TF IDF Vectorization

An introduction of the method: A TFIDF vectorization is the process of transitioning from raw text into a set of numbers. It's a BAG OF WORDS approach that in its simplest form just create a feature for each word, defined by now as vocabulary, and it flags with a binary encoding whether or not the term is present in the document. It's a naive approach that can be expanded using weights, counting the number of times a term appears in the specific document and multiplying it by a scaling factor that ensure high importance to words that are numerous inside the documents and less importance to words that often present among all documents. Example : Hate is an important word because is not used in all the context, therefore it will have an high score. "The" is a very common word, and will instead get a low score. This final form is defined as Term Frequency Inverse Document Frequency and is the form used in this project.

Implications:

- Variability Factor : Length of Vector / Vocabulary. It is assumed that an higher dimension will bring asymptotically more information to the model.
- Instability Factor: If the length of the vector exceed the number of rows the model will be highly unstable.

Note that we will fit the TFIDF model just on train data to avoid any data leakage and keep it for transforming the test set. After a few trials we decide that the best trade off is around 500.

Modeling

Possible Goals and Metrics

As for the success metric we assume that there is not any type more likely than the others, therefore accuracy in a balanced test will be the optimizing metric.

Given that the best way to asses the MBI type is the test itself, even if we could measure it's predictive power we certainly will be below that number. We expect to perform better than random or an untrained human in terms of precision and way faster than an expert in terms of time to prediction. The real benefit of a model will mostly be that we can apply a first grasp of prediction at a very large scale.

Going at Random

Table 4: Random Model Performance on Test Data

Accuracy	Kappa	AccuracyLower	AccuracyUpper	AccuracyNull	AccuracyPValue	McnemarPValue
0.05625	-0.0066667	0.0413214	0.074546	0.21125	1	NaN

Given the random model baseline we discuss why we have decided to use *Decision Trees* as model.

The major issue is that we do not directly work with the text, instead we try to create a feature space which partially represent the amount of total information inside the text. We cannot rely on models with many assumptions or too sensible to high dimensions.

Decision Trees are good for multiclassification problems and often provide similar performance to other models. Trees tends to have low bias but high variance, therefore using a single Decision Tree for the prediction will lead to overfitting or in general high variance.

Random Forest is so the final choice for our model according to the following reasoning.

- Features sampling will better deal with the problem of high dimension
- Bagging will decrease the variance of predictions
- Trees are more explainable and the idea of voting will ensure more stability in the predictions

We will use all default parameters for the algorithm since tuning the params will be computationally expensive. Generally we expect:

- The variance of prediction to decrease with the increasing of estimators
- The Bias to decrease with the increase of trees depth

Multiclassification Approach

Table 5: Confusion Matrix for Multiclass

Accuracy	Kappa	AccuracyLower	AccuracyUpper	AccuracyNull	AccuracyPValue	McnemarPValue
0.70375	0.684	0.6707747	0.7352155	0.0775	0	NaN

	ENFJ	ENFP	ENTJ	ENTP	ESFJ	ESFP	ESTJ	ESTP	INFJ	INFP	INTJ	INTP	ISFJ	ISFP	ISTJ	ISTP
ENFJ	29	0	3	1	0	2	1	0	6	2	1	0	2	3	0	0
ENFP	1	37	1	0	0	1	2	1	1	2	3	0	1	0	0	0
ENTJ	1	1	40	2	0	0	1	0	1	1	1	2	0	0	0	0
ENTP	0	5	2	29	0	0	0	1	1	1	7	2	1	1	0	0
ESFJ	0	1	0	0	44	1	0	0	1	0	0	0	1	2	0	0
ESFP	1	2	1	0	0	34	2	1	1	1	0	1	0	5	0	1
ESTJ	1	0	0	1	0	0	43	2	2	0	1	0	0	0	0	0
ESTP	0	0	2	0	0	1	0	41	1	3	2	0	0	0	0	0
INFJ	2	2	2	0	1	1	0	3	32	2	2	0	2	0	0	1
INFP	0	5	0	0	0	1	1	1	1	29	3	2	2	0	4	1
INTJ	0	4	3	1	0	1	0	1	2	1	29	2	1	2	2	1
INTP	0	0	1	3	0	0	0	0	0	0	5	39	0	1	0	1
ISFJ	0	2	0	0	2	2	0	0	2	0	0	0	36	2	4	0
ISFP	0	0	0	0	1	5	0	0	1	0	3	0	3	37	0	0
ISTJ	0	1	1	0	0	1	0	0	0	2	1	3	0	1	40	0
ISTP	2	0	3	2	0	0	0	7	0	1	4	1	1	0	5	24

PCA Approach

Given the high dimensionality and the high chance of breaking many assumptions for several models, an option could be to use a dimensionality reduction technique. We have not by any means a hope for model improvement, since pca is an unsupervised technique and does not optimize for a better association between response and predictors. As rule of thumb we aim to capture 95% of the total variability and hope for an acceptable performance reduction.

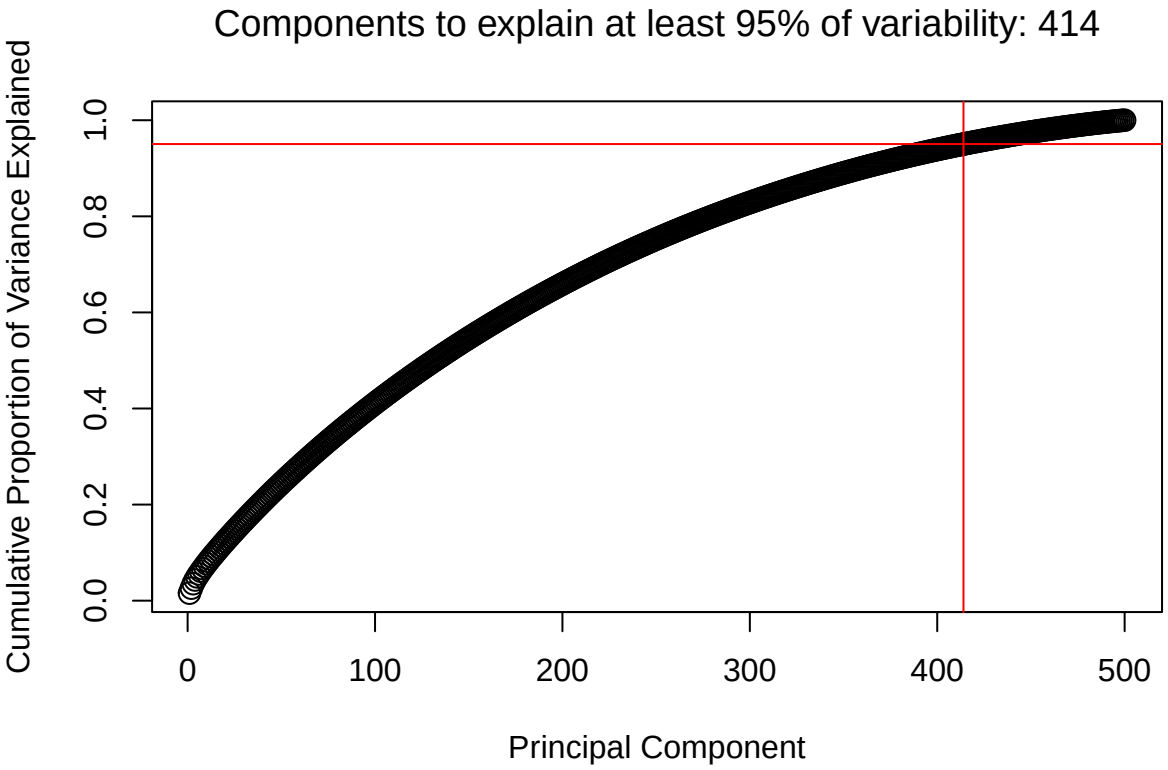

Table 7: Confusion Matrix for Multiclass

Accuracy	Kappa	AccuracyLower	AccuracyUpper	AccuracyNull	AccuracyPValue	McnemarPValue
0.4175	0.3786667	0.3830605	0.4525497	0.1	0	NaN

	ENFJ	ENFP	ENTJ	ENTP	ESFJ	ESFP	ESTJ	ESTP	INFJ	INFP	INTJ	INTP	ISFJ	ISFP	ISTJ	ISTP
ENFJ	16	5	1	0	7	2	1	1	2	6	1	1	6	1	0	0
ENFP	5	17	2	1	4	1	2	1	1	4	3	1	3	1	3	1
ENTJ	2	0	21	0	1	2	1	2	2	0	10	6	0	1	1	1
ENTP	0	4	2	9	5	5	1	3	1	1	8	4	1	1	2	3
ESFJ	0	2	0	0	42	1	0	0	1	1	0	0	3	0	0	0
ESFP	1	1	0	0	1	39	2	2	0	0	0	0	0	3	0	1
ESTJ	1	0	1	0	2	0	36	2	2	1	1	0	0	1	2	1
ESTP	0	0	2	0	0	0	0	41	0	0	2	1	0	0	1	3
INFJ	9	7	1	1	1	2	0	0	8	9	3	1	2	2	3	1
INFP	7	7	0	0	0	2	3	0	1	15	7	5	0	1	1	1
INTJ	0	1	11	3	2	0	1	4	1	0	16	8	0	0	1	2
INTP	2	4	6	2	0	1	0	3	2	2	11	13	0	0	1	3
ISFJ	0	2	2	3	12	4	1	0	1	2	0	1	15	6	1	0
ISFP	1	2	1	0	2	10	4	0	2	6	0	1	2	19	0	0

	ENFJ	ENFP	ENTJ	ENTP	ESFJ	ESFP	ESTJ	ESTP	INFJ	INFP	INTJ	INTP	ISFJ	ISFP	ISTJ	ISTP
ISTJ	0	0	5	1	0	2	1	1	4	6	2	2	3	1	21	1
ISTP	0	1	3	2	1	2	1	9	3	2	4	8	3	1	4	6

Prediction as Recursive Problem

This last approach is justified from the intuition that having a relative small size of training data, due to the minority class constraints, is affecting the potential performance. The question is:

- *Can we improve the model performance if we assemble a predictions using sub models trained with much more data?*

In order to use more data inside the model we can approach the prediction problem as a recursive problem. The idea is the following:

- Create a Pseudo Label
- For each :
  - Fit the model using the tfidf vector as features and the pseudo label as target
  - Make a prediction for the pseudo label
- Concatenate the subprediction to get a global prediction.

Note: The pseudo Label is built using one of the complementary letter for each component.

- Pseudo Label Energy = 1 in case the sample, regardless of the type, contains an E (extrovert component)
- Pseudo Label Energy = 0 in case the sample, regardless of the type, contains an I (introvert component)

Table 9: Frequency of Data for each Binary Classification Training

0	1
2000	2000

Table 10: Confusion Matrix for Recursive Model

Accuracy	Kappa	AccuracyLower	AccuracyUpper	AccuracyNull	AccuracyPValue	McnemarPValue
0.3675	0.3253333	0.3340071	0.4019734	0.11875	0	NaN

	ENFJ	ENFP	ENTJ	ENTP	ESFJ	ESFP	ESTJ	ESTP	INFJ	INFP	INTJ	INTP	ISFJ	ISFP	ISTJ	ISTP
ENFJ	10	8	1	1	3	3	1	2	11	4	1	2	0	1	0	2
ENFP	15	16	2	1	1	7	0	1	4	0	0	2	1	0	0	0
ENTJ	1	3	17	6	2	0	3	6	2	0	6	3	0	0	1	0
ENTP	3	7	7	18	0	1	1	9	0	0	1	3	0	0	0	0
ESFJ	2	2	0	2	9	4	0	8	6	3	2	2	5	3	2	0
ESFP	2	2	0	0	9	15	1	8	0	1	1	0	3	3	2	3
ESTJ	0	1	0	0	7	4	18	12	4	0	0	0	0	0	2	2
ESTP	0	0	0	0	5	0	4	30	2	1	1	0	0	0	4	3
INFJ	4	3	1	0	2	2	0	0	29	1	3	1	2	1	1	0
INFP	1	3	0	1	1	0	0	0	9	22	2	2	1	6	1	1
INTJ	0	0	2	1	2	0	4	0	2	1	36	0	0	1	1	0
INTP	0	1	0	5	0	0	0	1	1	0	6	34	0	0	0	2
ISFJ	3	2	0	0	11	4	1	2	5	3	5	0	5	2	1	6
ISFP	1	1	1	1	3	18	1	4	1	2	0	2	5	6	3	1

	ENFJ	ENFP	ENTJ	ENTP	ESFJ	ESFP	ESTJ	ESTP	INFJ	INFP	INTJ	INTP	ISFJ	ISFP	ISTJ	ISTP
ISTJ	1	0	0	1	2	3	3	3	3	1	2	1	6	2	13	9
ISTP	0	0	0	0	0	3	1	9	0	0	3	2	3	5	8	16

Conclusions

	Random	Multiclass	Multiclass.PCA	Recursive.Binary	Energy	Information	Decision	Organize
Accuracy	0.05625	0.70375	0.4175	0.3675	0.735	0.81375	0.79	0.68125

According to the results the best model is the **Multiclass model**. It must be mentioned that the recursive model was significantly more powerful in resolving the sub-task but the combining process was quite challenging and led to many errors. The PCA Multiclass model did not gave any advantage in terms of performance, but it is assumed that for higher dimension will help model speed. Acquiring more data did not therefore help the process, but it is assumed that a deeper data quality work using lexical richness filters and more elaborated embedding techniques might help the model to improve. In conclusion we are aligned with the majority of performance related to this topic and we confirm that in terms of business result is generally much more helpful to use a model rather that humans or data visualization techniques to resolve the problem of customer segmentation.