**A close-up of a compass

Description automatically generated with medium confidence**

**Personality Predicition Write-up**

SMU Data Science

12/01/2021

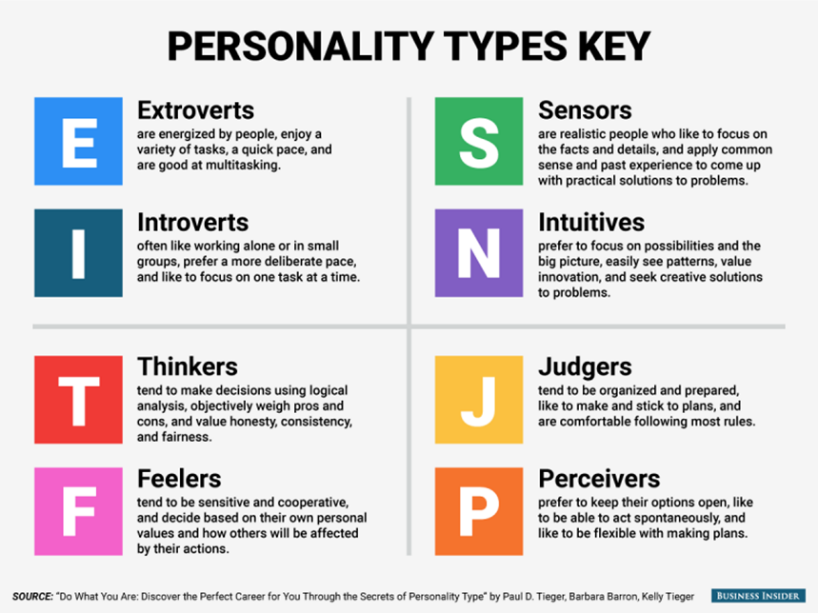
By:

Jon Arnold, Steven Green and Lucas Da Silva

**Inspiration**

Our team was searching for something a little outside of the box that could fun and potentially useful for professionally. It was important for us that the subject selected had available a strong dataset and published papers online, so that we would not need to spend a lot of time cleaning and editing the data.

It is becoming more and more common for companies to perform some type of personality test when considering a candidate for a position or promotion, so what if we could streamline the prediction of a person’s personality type?



**Data Source**

Our group decided to base our project on two datasets that we retrieved from Kaggle.com. Our main data set consisted of 8600 rows and 2 columns, and it was the base for our Natural Language Processing library and the personality types associated with them. The NLP library was produced by analyzing 50 social media posts published by people with each personality type and key words that help identify the personality type.

The second data source included the breakdown of the most popular personality types in more than 150 countries. We used this dataset to identify the most popular personality type in each country and how that was reflected across the globe. Although several countries displayed a very balanced distribution of personality types withing their population, we chose to just look at the most popular personality type for our analysis.

Chart, sunburst chart

Description automatically generated

**Model Cleaning**

The model cleaning portion of this project consisted of cleaning, tokenizing, and lemmatizing the posts that were in the Kaggle database. The first step was to check to see how many posts there were and how many terms would that be. Then we decided to clean the data to eliminate the three vertical bars that separate the posts, commas, apostrophes, and uppercase letters. For this process, we used the NLTK python library as well as the regular expression (re) syntax. Once we cleaned the data, we then tokenized the posts. Tokenized means that we inserted commas in all the posts so that those posts can individually be evaluated. After that, we then filtered the posts using stopwords. Stopwords is a NLTK.corpus dictionary that is further used to clean up our data of common words that would only be noise for the dataset. We used both for loops and list comprehensions to achieve this goal. Once the data was loaded, we found out the data which is in a list format and not a dataframe has 87000 words. This would not be feasible to collect given our computer memory. After this we decided to filter this down to the first entry that would allow us to lemmatize that post at a sacrifice of accuracy, but it would allow the machine learning code to be run successfully in Jupyter Notebooks. This list would be 4343 words long, and after that we used the tf-idf feature generation to vectorize the list. This creates a list that finds which words are only in a certain post. We then created a dataframe with the columns being the feature names and the list being the count. We also created a sentiment column using TextBlob so that each word would have a sentiment from negative 1 to positive 1 for being negative or positive. After that, due to memory concerns, what we did was create a CSV file that had the tf-idf dataframe to then analyze in a different notebook.

After the new notebook was created, the first thing we did in the new notebooks was to create a feature and a target which we did for our train test split. We also had to convert the dataframe back to arrays for this. We then ran the machine learning models Logistic Regression model. We also ran the Decision Tree, Random Forest, Ada Boost, Gradient Boosting, Extreme Gradient Boosting (XGB), K-nearest neighbors (KNN), Light Gradient Boosting, and support vector classification models. Because of the sacrifice in accuracy that was mentioned above, the most accurate model that we found was 21% accurate. Also, all the models had that accuracy, so we just chose the logistic regression model for our predictor. After we chose which model to use for the predictions, we decided to save the model as a python pickle file to be opened and used in the predictor. Our predictor was us using the pickle file to create a prediction-based model that would take any input text and take it through the system to bring back a prediction of a personality type. For that we followed some of the same processes as the NLTK notebook. Cleaning the user inputted text, looping through the text to create an input, creating a sentiment analysis for that input, and making a prediction on the personality type based on that input.

**Model Development**

Our team wanted to develop a model that could predict the personality type of a person based on a few sentences where the individual can describe themselves. So, we needed to create a website that would collect these few sentences where the user is self-describing, reference it back to our NLP library and spit out a prediction of the user’s personality type.

The model is also associating the words in the library with positive or negative sentiments, so that, when it predicts your personality type, it can also rate the sentences provided by the user as positive or negative. This feature can be useful when trying to understand if a person sees themselves positively or negatively when describing themselves.



**(To be added)**

**Obstacles and Issues**

Our issues were mostly coming from the size of the data and how the tools we were using would crash or could not handle such large sources. The NLP library, when combining the possible word combinations, provided millions of outcomes and that source became too large to reference, so the team needed to limit the number of possible outcomes and decrease the amount of social media posts being analyzed.

The dataset about the countries was difficult to graph and edit, so the information regarding the types of personalities in each country had to be reduced to just the most popular personality type in each country. The tableau version used by our team was not very stable as well, so the program would crash unexpectedly and that made it difficult to generate the visualizations we desired.

Text

Description automatically generated

**(To be added)**

**Findings**

We found that our model provided about 20% accuracy in all scenarios that it could predict and that the most popular personality type across the world was INFP (80.30%), followed by ENFP (12.55%). We also found that it is very difficult to predict a personality type because people will adjust their inputs based on the environment their environment or base on how they want to be perceived by their peers. So, since the dataset is based on social media posts, the group understands that it is skewed on how people want to be viewed publicly on the web.



**(To be added)**

**Resources**

* [**https://www.kaggle.com/datasnaek/mbti-type?select=mbti\_1.csv**](https://www.kaggle.com/datasnaek/mbti-type?select=mbti_1.csv)
* [**https://www.kaggle.com/yamaerenay/mbtitypes-full?select=countries.csv**](https://www.kaggle.com/yamaerenay/mbtitypes-full?select=countries.csv)
* [**https://www.myersbriggs.org/**](https://www.myersbriggs.org/)

**Website created by the group**

* <https://jarnold-personality-prediction.herokuapp.com/>