Classification of Personality Traits

Syam Sundar Herle (syampara@iu.edu) | Vignesh Sureshbabu Kishore (vsureshb@umail.iu.edu) | Vighnesh Nayak (vrnayak@indiana.edu)

Introduction

The Myers Briggs Type indicator(MBTI) is a personality type indicator that divides people into 16 personality type based on 4 base types as follows: Introversion (I) – Extroversion (E), Intuition (N) – Sensing (S), Thinking (T) – Feeling (F), Judging (J) – Perceiving (P). The base theory behind the test is that random variation in behavior is actually orderly and consistent due to the basic distinction in the ways people use their perception and judgement. One way to conduct this test would be using a questionnaire based on which peoples' personality types can be determined. However, our intuition is that the way a person expresses himself through his/her words might be a good enough indicator of their personality. Following this intuition, we are trying to determine the personality of Kaggle [1] users based on their comments. We have used the MBTI dataset from Kaggle [1] for the purpose of our analysis [1]. This dataset contains the contents of their comments and their corresponding personality type. In our work, we are trying to build prediction models based on some Natural Language Processing features.

Data

The data has following attributes,

- Type: This person 4 letter MBTI code/type
- Posts: A section of each of the last 50 things they have posted (Each entry separated by "|||" (3 pipe characters))

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

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

For an example if an individual is preferring introversion, intuition, thinking and perceiving, then in the MBTI the particular individual will be coded as INTP.

	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...

Figure 1Snipet of dataset

Data Pre-Processing

As the posts in the data were scrapped from the users written texts, pre-process methods were required to apply on the data before any of the Natural Language Processing task or any of the machine learning models are applied. The following task are applied,

- Replacing '|||' to '' in the posts.
- Removing punctuations and taking care of smileys.
- Removing extra space, as these are the post from user's and may have human error
- Format any URL links and hyperlinks to keyword 'url'.
- Remove digits from the posts.
- Changing all the words in the posts to lower case.
- Removing stop words from the posts. The stop words are created using 'scikit learn' feature extractor stop words.
- Lemmatize the words in the posts to limit different form of words to the single one. For example, 'oranges' to 'orange'.

Feature Models

Unigram Features:

According to [1] Text Analysis is a major application field for machine learning algorithms. However, the raw data, a sequence of symbols cannot be fed directly to the algorithms themselves as most of them expect numerical feature vectors with a fixed size rather than the raw text documents with variable length. In order to address we have decided to use scikit learn feature extract package [2] to extract words and letters and convert it to numerical values. The general process of extraction and creation of the features, we have employed the following,

- Tokenizing: Strings and giving an integer id for each possible token, for instance by using white-spaces and punctuation as token separators [2].
- Counting: The occurrence of each token in each post [2].
- Normalizing: The important tokens weights are calculated and normalized [2].

This specific task if collective done are known as, **Bag of Words** or **Bag of n-gram words** [2]. In our case, we have used tokenization and counting of our post to create a fixed number of features 5000 to feed into machine learning models.

Tf-Idf Features:

Tf-idf stands for *term frequency-inverse document frequency*, and the tf-idf weight is a weight often used in information retrieval and text mining [3]. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

Typically, the tf-idf weight is composed by two terms: the first computes the normalized Term Frequency (TF), aka. the number of times a word appears in a document, divided by the total number of words in that document; the second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears [3].

- **TF (Term Frequency)**: TF which measures how frequently a term occurs in a document. The TF is calculated as,
    - TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document) [3].
- **IDF (Inverse Document Frequency):** IDF measures how important the term is. While computing TF all terms are considered as of equal importance. But IDF weighs down and normalize all the words.
    - IDF(t) = log_e(Total number of documents / Number of documents with term t in it) [3].

Doc2vec Features:

For our work, we tried to train a doc2vec model with the genism library. Then, we tried use the numerical vectors for each individual's text to train a binary model which can be used for the classification problem. Doc2Vec tries to model a single document or paragraph as a unique a single real-value dense vector.

Supervised Classification Models

Typical machine learning algorithms or predictive models are used in this approach, typically the ML approach can be divided into two models, Supervised learning and Unsupervised learning. In supervised learning, a model is trained with the help of the labeled training data set and evaluated on the unseen data set which are known as test data set. After evaluating the test data classification accuracy are calculated to know how good the trained model is in classifying. For this work we have used following models for the classification problem.

Multinomial Naive Bayes

Naive Bayes classifiers are linear classifiers that are known for being simple yet very efficient. The probabilistic model of naive Bayes classifiers is based on Bayes' theorem, and the adjective naïve comes from the assumption that the features in a dataset are mutually independent. In practice, the independence assumption is often violated, but naive Bayes classifiers still tend to perform very well under this unrealistic assumption. Bayes' theorem forms the core of the whole concept of naive Bayes classification. The 'Posterior probability' is calculated by the given formula,

Posterior Probability = (conditional probability * prior probability)/evidence [4]

The Naive-Bayes is one of the probabilistic classifier, which works based on probabilities value to determine class, the Naive-Bayes model predicts the posterior probability of a class to which

the posts belongs to, based on distribution of words in the posts. For a given features it calculates the probability of the label assuming all the features are independent to each other, the Naive-Bayes follows the following equation to classify the class to which the posts belong's to given the features, the features for our work are generated by the uni-gram, tf-idf, word2vec.

$$P\left(\frac{label}{features}\right) = \frac{P(label) * P\left(\frac{f_1}{label}\right) * .. * P\left(\frac{fn}{label}\right)}{P(features)} \text{ [4]}$$

Logistic Regression

Multinomial Logistic regression is used when you we categorical dependent variable with two or more unordered level. It is related to logistic regression except we have more than one possible outcome. In our work, we have used three different possible features, unigram, tf-idf, word2vec. The most common form of the model is a logistic model that is a generalization of the binary outcome of standard logistic regression involving comparisons of each category of the outcome to a referent category.

Support Vector Machine

Support Vector Machine model works based on the linear separation of the features which separates features in the search space to separate them based upon their classes, SVM is one of the linear classifier used in supervised learning ML approach. In SVM, the features are transformed to a higher dimensional state space and hyperplanes are used to separate the features, hyperplanes are determined by support vectors, which are features closer to the decision boundary of the separation of the features. For *n* class of class the model needs *(n – 1)* support vectors.

Evaluation Metric and Results

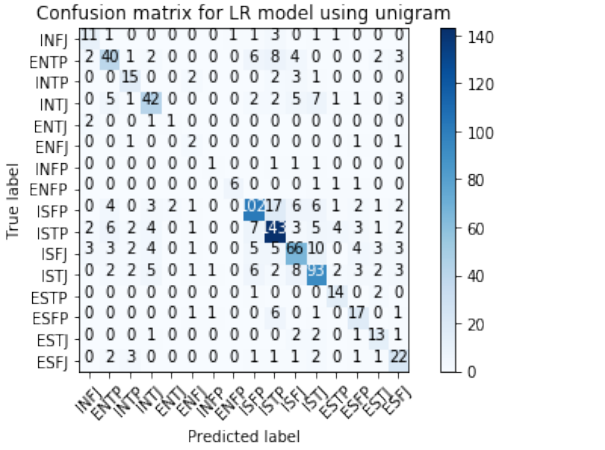
Results

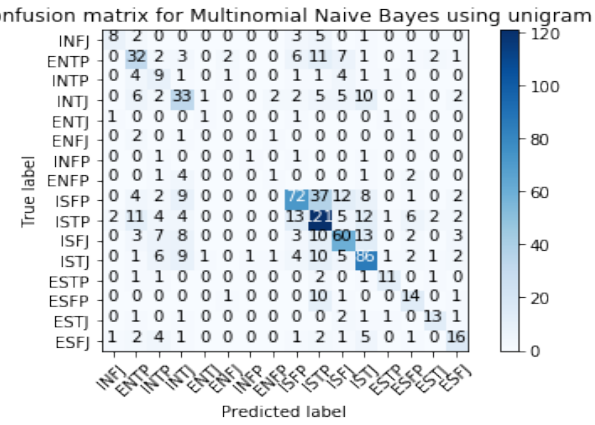
Model	Unigram	Tf-idf	Doc2vec
Multinomial Naïve Bayes	56.9%	29.4%	29.9%
Logistic Regression	67.4%	11.7%	33.1%
Support Vector Machine	49.6%	61.9%	31.9%

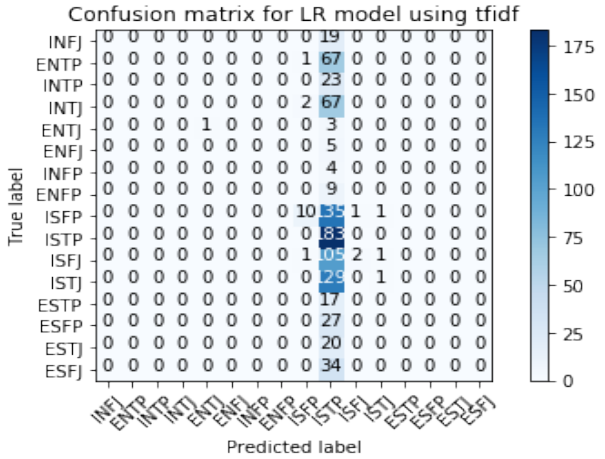
Figure 2 Model Accuracies

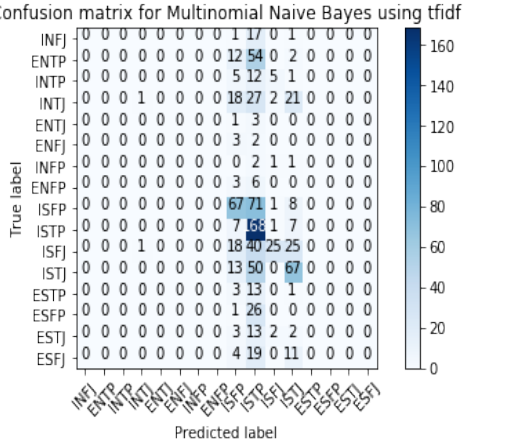
Evaluation metric

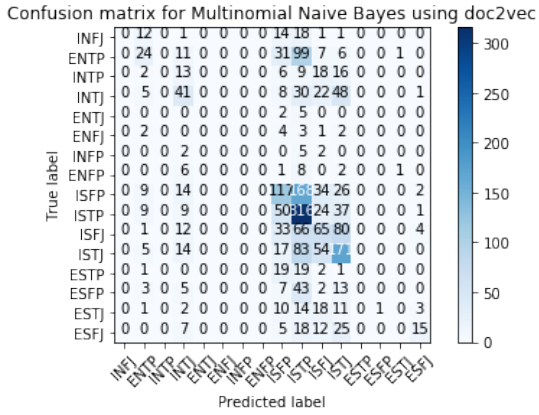
For the evaluation metric of models for all the features we will use confusion matrix, confusion matrix is tabular representation of the 'True Positive' and 'False Positive'. Confusion matrix is a good representation of the model performance and it will also useful to detect overfitting.

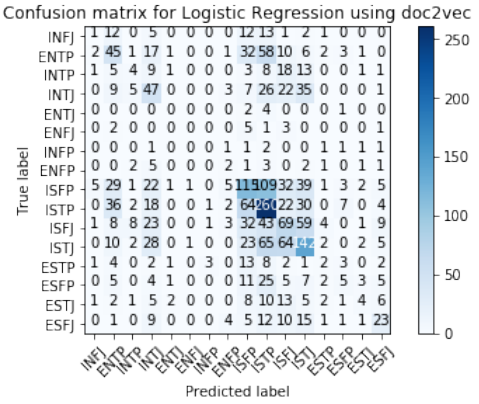

Confusion matrix for LR model using unigram
True label
Predicted label
INFJ ENTP INTP INTJ ENTJ ENFJ INFP ENFP ISFP ISTP ISFJ ISTJ ESTP ESFP ESTJ ESFJ
140 120 100 80 60 40 20 0

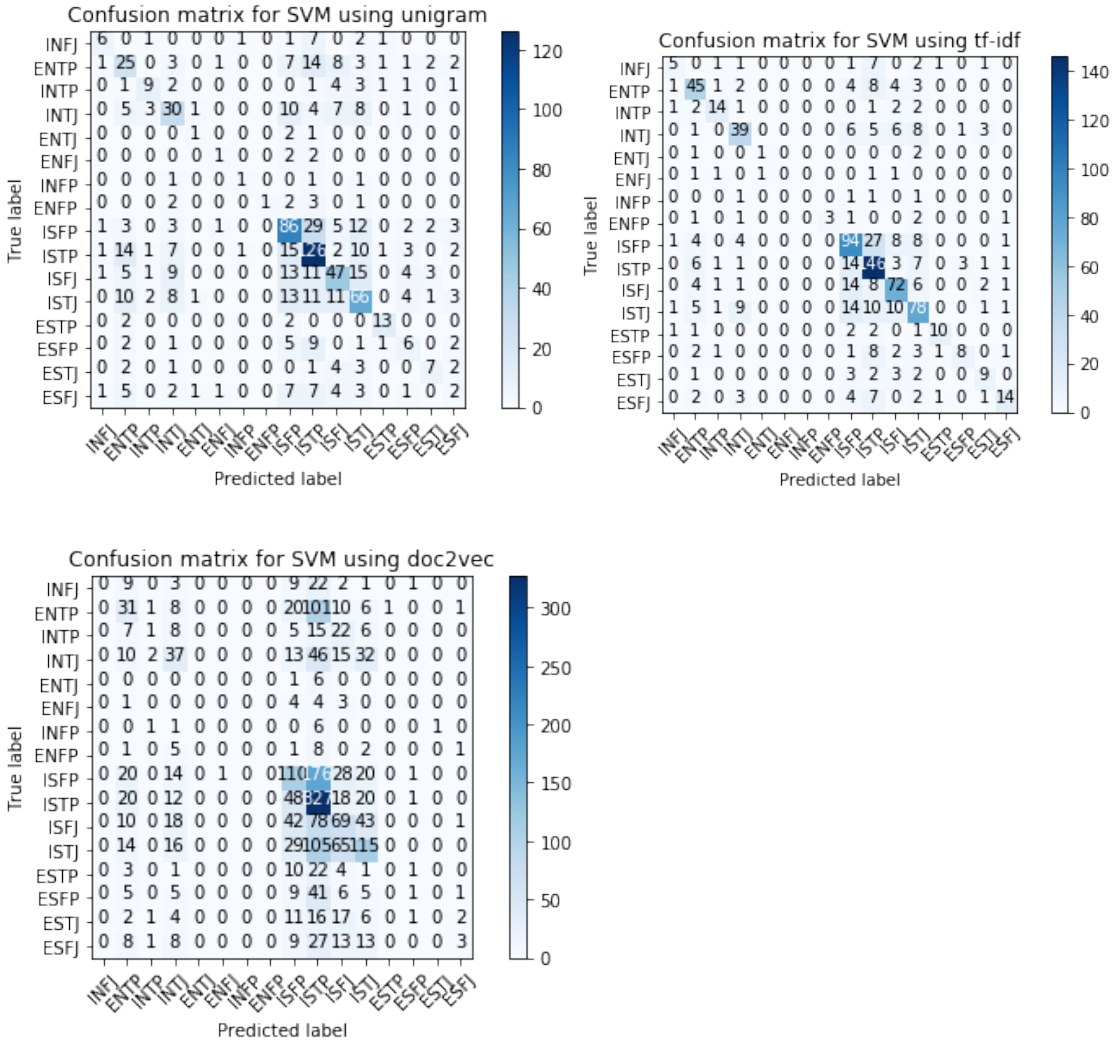

Confusion matrix for Multinomial Naive Bayes using unigram
True label
Predicted label
INFJ ENTP INTP INTJ ENTJ ENFJ INFP ENFP ISFP ISTP ISFJ ISTJ ESTP ESFP ESTJ ESFJ
120 100 80 60 40 20 0


Confusion matrix for LR model using tfidf
True label
Predicted label
INFJ ENTP INTP INTJ ENTJ ENFJ INFP ENFP ISFP ISTP ISFJ ISTJ ESTP ESFP ESTJ ESFJ
175 150 125 100 75 50 25 0

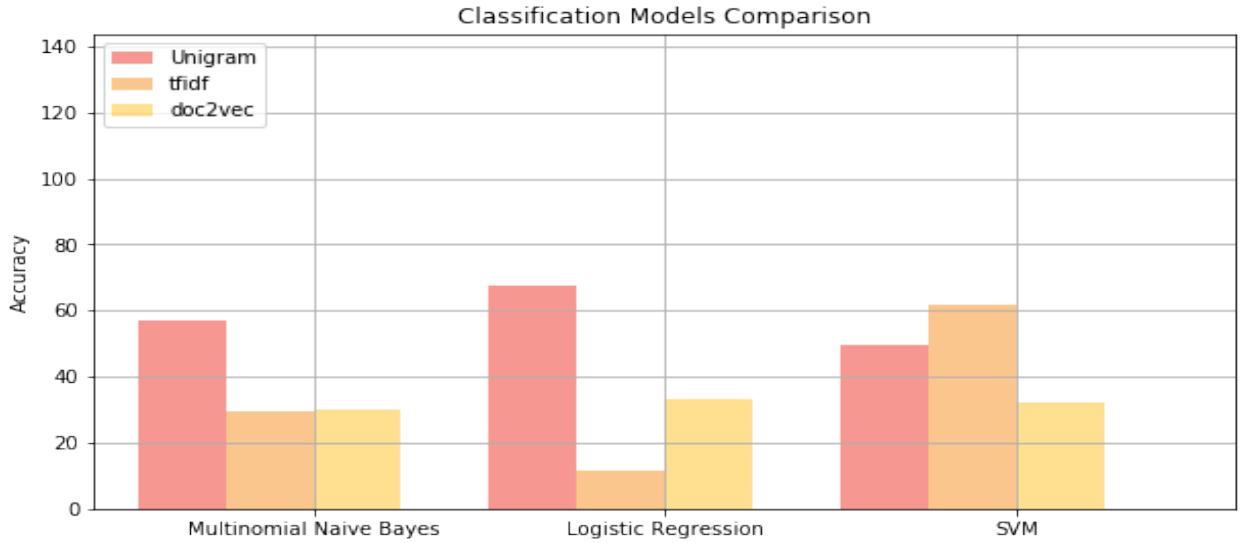

Confusion matrix for Multinomial Naive Bayes using tfidf
True label
Predicted label
INFJ ENTP INTP INTJ ENTJ ENFJ INFP ENFP ISFP ISTP ISFJ ISTJ ESTP ESFP ESTJ ESFJ
160 140 120 100 80 60 40 20 0


Confusion matrix for Multinomial Naive Bayes using doc2vec
True label
Predicted label
INFJ ENTP INTP INTJ ENTJ ENFJ INFP ENFP ISFP ISTP ISFJ ISTJ ESTP ESFP ESTJ ESFJ
300 250 200 150 100 50 0


Confusion matrix for Logistic Regression using doc2vec
True label
Predicted label
INFJ ENTP INTP INTJ ENTJ ENFJ INFP ENFP ISFP ISTP ISFJ ISTJ ESTP ESFP ESTJ ESFJ
250 200 150 100 50 0



Comparison of accuracies



Conclusion

In our work, we tried to build predictive models like Support Vector Machine, Multinomial Logistic Regression and Multinomial Naïve Bayes based on different input features ranging from unigram, tfidf and doc2vec to predict/classify the personality of the users from the posts written by them. Based on above model's accuracy comparison Figure 2 Model Accuracies, we can see that for unigram features Multinomial Logistic Regression works excellently with an accuracy of 67.4% and Multinomial Naïve Bayes works good yielding an accuracy of 56.9% and SVM works fairly yielding 49.6%. For 'tfidf' input features Support Vector Machine works excellently at an accuracy of 61.9%. Overall Multinomial logistic regression works well for unigram and SVM works well for 'tfidf' and 'doc2vec' input features seems to be not good for Personality Trait classification.

Bibliography

[1] Standford, "Unigram," 2107. [Online]. Available: https://lagunita.stanford.edu/c4x/Engineering/CS-224N/asset/slp4.pdf.

[2] s. learn, "scikit," [Online]. Available: http://scikit-learn.org/stable/modules/feature_extraction.html.

[3] tf-idf. [Online]. Available: http://www.tfidf.com/.

[4] standford, "Standford," [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/6.pdf.

[5] Wikipedia, MBTI, vol. 219, indiana: wikipedia.