Personality Prediction

Aayush Gautam(B20EE002)

***Abstract*—This Paper reports my experience on building a MBTI personality classifier. I used the MBTI dataset that was shared in with the report file for my classification. This paper reports my observations and the performance of different models on classification and compare their results.**

I. Introduction

The Myers-Briggs Type Indicator(MBTI) is the name of a personality test designed to assess personality type of a person. It was Developed by Katherine Briggs and her daughter Isabel Myers during the World War II . The MTBI is popular with recruiters and managers because studies showing this assessment show clusters of different personality types in different professions.[1]

In this paper I have used different models to predict what MBTI type a person belongs to depending on the social media posts of the person. The models are based on how certain words appear more often in certain personality types and using this the model tries to assign a personality type.

The following images show how different words are used by different personalities types.

INTJ

ESTJ

Dataset

MBTI dataset:

The dataset has 8675 rows and 2 columns containing:

- type Columns indicating the type of personality
- Posts column having social media posts

II. Methodology

A. Overview

I have implemented the following classification models and compared the performance.

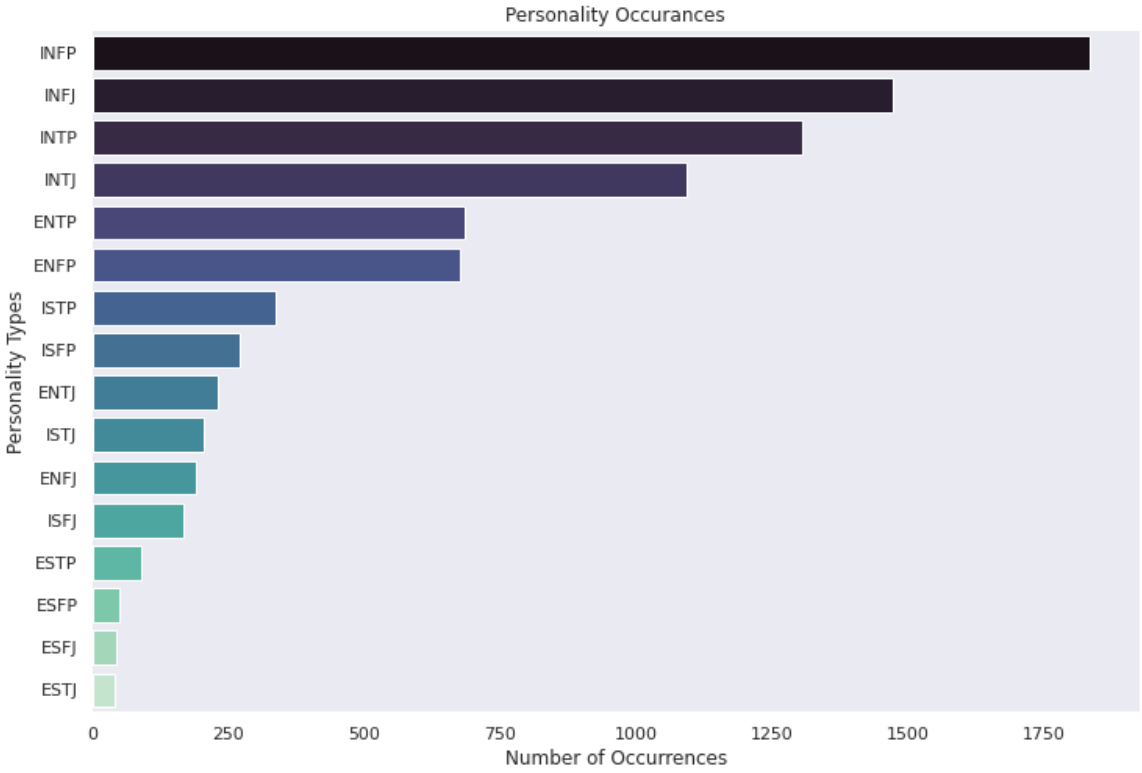
- Random Forest Classifier
- Decision Tree Classifier
- Support Vector Machine(Linear SVM).
- Multinomial Naive Bayes
- AdaBoost Classifier
- Xgboost Classifier

B. Data Exploration and Preprocessing

The dataset didn't have any NULL values.

The data was split in 80:20 ratio for training and testing.

The columns values of type and Posts were both of the form string.



On plotting the count of unique values of different types present and plotting the values by sorting them in descending order we get the following plot.

C. Noise Removal

The posts column had a lot of noise like links to websites, three bars separating sentences special characters and so on. These were removed because they are not relevant. We also try to do lexicon normalisation on the data as well as object standardisation[2]. To help us get the data in the desired format. I used ‘nltk' module for removing the noise.

D. Feature Engineering

Here after the noise removal we convert the textual data into features Term Frequency - Inverse Document Frequency(TF-IDF). This method converts the text directly into numbers.

After feature engineering our dataset now has 85754 columns.

E. Implementation of Classification Algorithms

- Random Forest Classifier : Random Forest classifier is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.
    - Two types of Random Forest Classifier were used:
        - Random Forest Classifier with no limit on max depth
        - Random Forest Classifier with a max depth of 36.
- Decision Tree Classifier : Decision Tree Classifier is a supervised learning algorithm. The algorithm uses the training data to create rules that can be represented by a tree structure.
    - Two types of decision Tree Classifier were used:
        - Default Decision Tree classifier from sklearn
        - Decision Tree classifier with a max depth of 12
- Support Vector Machine : SVM is a supervised learning algorithm. It plots the data in a n dimensional space and classified by drawing hyper planes
    - Two Types of SVM used
        - SVM with linear Kernel
        - SVM with RBF Kernel

• Multinomial Naive Bayes : Multinomial Naive Bayes is a type of Bayes Classifier that assumes the features to follow a multinomial distribution an does the classification on this assumption

  • A simple Multinomial Naive Bayes classifier was used.

• AdaBoost Classifier : AdaBoost also called Adaptive Boosting is an ensemble technique. The most simple AdaBoost only has decision trees with one split.

  • A simple AdaBoost model was used

• XGBoost classifier : It is an optimized distributed gradient boosting library designed to be highly efficient.[4]

  • A simple XGBoost classifier was used.

No Dimensional Reduction technique was used because the feature matrix is sparse and PCA and LDA do not support Spare inputs.

No Sequential Feature selection was used because considering 85 thousand columns the amount of time to select even a few columns Is huge.

Grid search to find the optimal parameters was performed only on few models because these models took a lot of time to even train hence grid searching computationally expensive models were avoided.

III. Evaluation of Models

The following are the accuracies obtained for the different models.

S.No	Evaluation of Models		
	Model Name	***Parameter***	***Accuracy(%)***
1	Random Forest Classifier	Max Depth = None	40.00
2	Random Forest Classifier	Max Depth = 36	38.15
3	Decision Tree Classifier	Max Depth = default	46.80
4	Decision Tree Classifier	Max Depth = 12	50.14
5	Support Vector Machine	Linear Kernel C = 1.0	66.62
6	Support Vector Machine	RBF kernel C = 1.0	61.26
7	Multinomial Naive Bayes	-	21.44
8	AdaBoost Classifier	-	33.14
9	XGBoost Classifier	-	65.76

IV. Results and Analysis

From the above observed results we can see that SVM does better than most of the models whether we use linear kernel or an RBF kernel. XGBoost comes close to linear SVM but it is far more computationally heavier than it taking nearly 10-15 minutes to complete whereas linear SVM took less than a minute.

Random forest classifier experiences a loss in accuracy was the max depth was changed from None to 36 whereas the opposite happened for decision tree when setting the limit for max depth the accuracy increased this might be because Random forest uses more than one decision trees and averages the final prediction, where for the latter we only have one single tree.

Similarly adaboost also does bad considering it is similar to random forest but employs a slightly different algorithm.

Multinomial Naive bayes does the worst here which might be due to the assumption that textual data follows a multinomial Distribution.

Overall SVM is the preferred model..

Identify applicable funding agency here. If none, delete this text box.

REFERENCES

1. Myers-Briggs Type Indicator | Wikipedia
2. Personality Profile Prediction | Kaggle
3. Support Vector Machine | Towards Data Science
4. XGBoost Documentation.