Personality Detection

Multi-class classification

Abhishek Rajora (B20CS002)

Abstract :

This paper reports our experience with building a personality classifier. We have a dataset generated using 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. Recently there was a model of 8 distinct functions, thought processes or ways of thinking that were suggested to be present in the mind. Later this work was transformed into several different personality systems to make it more accessible, the most popular of which is the MBTI. I used various classification algorithms and compared their results in this report. Apart from this I used the custom text and classified it using the best model.

(*Source : MBTI Dataset | Kaggle*)

I. Introduction

Dataset Used :

MBTI: The file mbti_1.csv is used as the dataset.

The train dataset contains 8674 rows where each row represents a type of personality associated to a social media post with 2 columns containing :

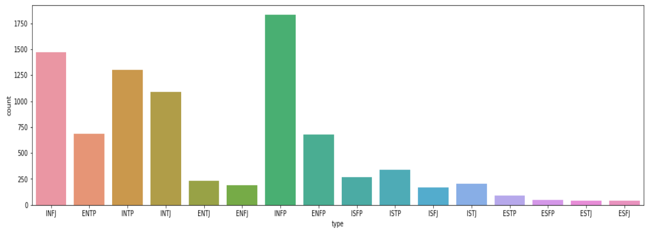
- Type of personality
- Text from the social network post

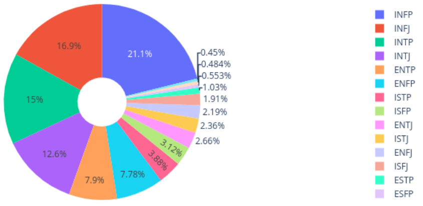
The dataset has been split into train and test with test size of 0.2

Posts dataset : 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)

The personality distribution is shown:





II. Methodology

Overview:

The Classification Algorithm implemented in this project:

- LGBMClassifier
- Logistic Regression
- LinearSVC
- Support Vector Classifier(SVC)
- XGBoost Classifier
- K Nearest Neighbors(KNN)
- CatBoostClassifier

  We also make use of NLP preprocessing techniques including vectorization.

Pipeline:

#Importing Modules and Analyzing Data

On counting the number of NULL values in the train dataset , it was found that there are no NULL values present. The class distribution was found to be imbalanced.

#Data Preprocessing

- The **clean_text** function is defined which takes in use of a **regex** module to remove links and other symbols from the text data.
- After cleaning the text density is visualized.
- Initially, **Bert tokenizer** was used which didn't give us good results. So we switched to **TfidVectorizer** provided by **sklearn**.

#Vectorizing data

- Class **Lemmatizer** is created to make out and update the stemmed words to meaningful words.
- The word dictionary used is developed in **WordNetLemmatizer**.

#Classification Models

Classification algorithms were implemented for this project:

- Lightgbm:
  - Boosting Type 'gdbt'(gradient boosting decision tree) is used
  - Parameters set are num_leaves = 3, max_depth = 10
  - Lgbm classifier with 400 n_estimators

- Logistic Regression : Logistic Regression model is widely used for binary classification but modified multinomial logistic regression can be used for multi-class classification.
  - Changing logistic regression from binomial to multinomial probability requires a change to the loss function used to train the model (e.g. log loss to cross-entropy

    loss), and a change to the output from a single probability value to one probability for each class label
    - Simple Logistic Regression is used

- LinearSVC : This class supports both dense and sparse input and the multiclass support is handled according to a one-vs-the-rest scheme.
    - Simple LinearSVC with C=0.3 is used

- SVM : In SVM , data points are plotted into n-dimensional graphs which are then classified by drawing hyperplanes.
    - Simple SVM with rbf kernel is used
    - Decision function type is set to 'ovo'(one vs one)

- XGBoost: XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework.
    - XGBoostClassifier with max depth of 6 is used
    - n estimators are kept at 50
    - Tree method is set to auto
- KNN(k- nearest neighbors) : KNN are supervised algorithms which classify on the basis of distance from similar points.Here k is the number of nearest neighbors to be considered in the majority voting process.
    - Simple KNN with n-neighbors being 5 is used
    - Weights are set to uniformly distributed

- CatBoost : CatBoost is an algorithm for gradient boosting on decision trees. It provides a gradient boosting framework which among other features attempts to solve for Categorical features using a permutation driven alternative compared to the classical algorithm.
    - CatBoostClassifier with loss function MultiClass is used
    - evaluation metric is set to MultiClass
    - Number of iterations are kept to 50 to avoid excess training time

#Custom TeXt Prediction

The text is passed on for pre-processing and vectorizatio. The most accurate model is chosen for the personality prediction of custom text.

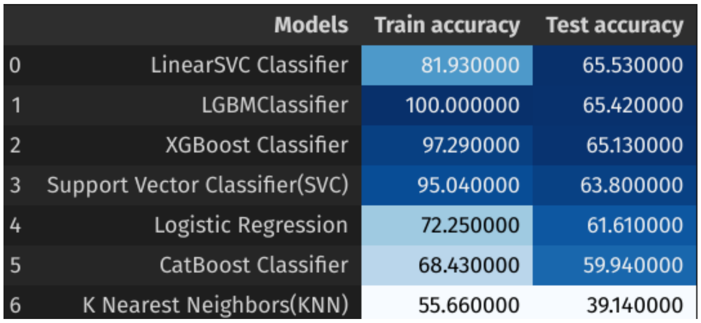
*****Note :*** *All the steps are briefly explained and visualized in the Notebook.*

III. Evaluation of Models

Results and Analysis:

The Accuracy of Models according to testing data is arranged below:

Table 1



	Models	Train accuracy	Test accuracy
0	LinearSVC Classifier	81.930000	65.530000
1	LGBMClassifier	100.000000	65.420000
2	XGBoost Classifier	97.290000	65.130000
3	Support Vector Classifier(SVC)	95.040000	63.800000
4	Logistic Regression	72.250000	61.610000
5	CatBoost Classifier	68.430000	59.940000
6	K Nearest Neighbors(KNN)	55.660000	39.140000

The models implemented were evaluated using techniques like - Classification report : precision , recall , f1 score and support. The score for each model shown below is the weighted average score of class wise predictions.

Table 2

	Precision	Recall	F1 Score	Accuracy Score
Lightgbm	0.66	0.65	0.65	65.42
Logistic Regression	0.64	0.62	0.58	61.61
LinearSVC	0.66	0.66	0.64	65.53
SVC	0.65	0.64	0.62	63.8
XGBoost	0.65	0.65	0.64	65.13
KNN	0.66	0.56	0.54	39.14
CatBoost	0.59	0.60	0.59	59.94

The table shows that all classifiers had nearly equally efficient performance. LinearSVC, Lightgbm, XGBoost and SVC performed fairly better than the rest of the models. LinearSVC however came out to be the most efficient one, both in terms of accuracy and time complexity. Therefore LinearSVC is preferred over other models and is used as the final classification model for the custom data.
The classification of text ***"We don't run away. We work in the dark to serve the light."*** is predicted as INTJ which stands for Introversion, Intuition, Thinking and Judging. This is a pretty good approximation for personality judgment. Moreover Oversampling can be used to increase the accuracy of the given model.

***End*