Pattern Recognition and Machine Learning Bonus Project Report Personality Prediction

Shreya Sachan B20EE065

May 1, 2022

Abstract

Personality refers to a characteristic pattern of thoughts, behavior, and feelings that makes a person unique. Personality Prediction from text has become popular in Natural Language Processing(NLP).The data obtained from social media is precious to determine the user personality type.The Myers-Briggs Type Indicator (MBTI) is a well-known personality test that assigns a personality type to a user by using four traits dichotomies.In this project We particularly focused on feature engineering techniques for text data and provide an insight comparison of different Machine learning and Deep learning models.

Index Terms

MBTI, Personality, NLP, K-Neighbors Classifier, Random Forest Classifier, XGBoost Classifier, SVM, CNN, Logistic Regression, TfidfVectorizer, Stemming, Lemmatization

Table of Contents

1. Introduction

In this project we are going to predict the personality from text data extracted from social media. Myers-Briggs Type Indicator (MBTI) is a used personality test. In total there are 16 types of personalities. The text data is preprocessed using the nltk library and then text data is converted to vector by use of TfidfVectorizer. Then finally we trained different ML and DL models. For the project, we used the MBTI dataset.

2. Libraries for the Model

Here we installed some dependencies to work with text data such as :

- Sklearn: library for machine learning in Python
- TensorFlow: which provides a collection of workflows to develop and train models using Python.
- nltk: a toolkit built for working with NLP in Python

3. Data Description, Visualization, and Preprocessing

3.1 Data Description

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)

This dataset contains over 8600 rows of data, on each row, is a person's:

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

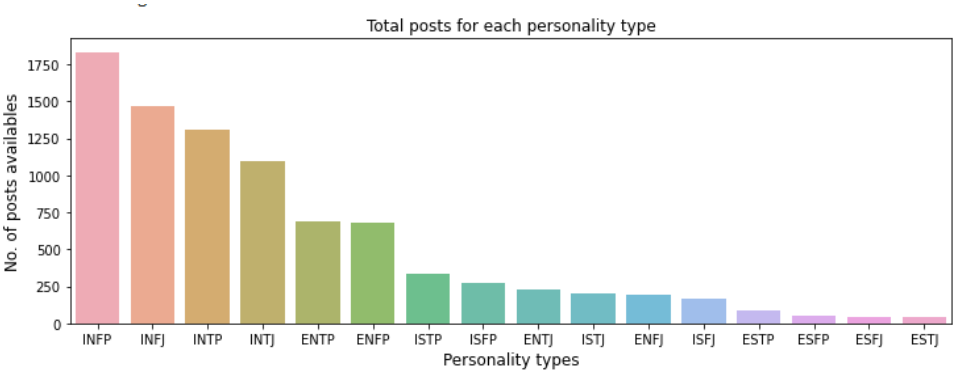
3.2 Data Preprocessing

- Text Preprocessing
    - Lower Casing - Lower casing is a common text preprocessing technique. The idea is to convert the input text into the same casing format so that 'text', 'Text', and 'TEXT' are treated the same way. This is more helpful for text featurization techniques like frequency, tfidf as it helps to combine the same words together thereby reducing the duplication and getting correct counts / tfidf values. This may not be helpful when we do tasks like Sentiment Analysis (where upper casing refers to anger and so on).
    - Removing MBTI Personality Words from the post- In order to get valid model accuracy estimation for unseen data.
    - Removal of URLs
    - Removal of Punctuations - This is again a text standardization process that will help to treat 'hurray' and 'hurray!' in the same way. But for our case some symbols like "!" may be helpful for personality prediction. The string.punctuation in python contains the following punctuation symbols !"#$%&'()*+,-./:;<=>?@[\]^_{|}~`.We can add or remove more punctuations as per our needs.
    - Removal of stopwords - Stopwords are commonly occurring words in a language like 'the', 'a, and so on. They can be removed from the text most of the times, as they don't provide valuable information for downstream analysis. We can safely use them, the stopword list for english language from the nltk package.
    - Removal of Frequent words -In corpus, we might have some frequent words which are of not

so much importance to us. So we can remove the frequent words from the corpus.(Tfidf automatically takes care of it). We removed the top 10 frequent words.

- Stemming - Stemming is the process of reducing inflected or derived words to their word stem, base, or root form. For example, walks and walking, stemming will stem the suffix to make them walk. There are several types of stemming algorithms. We will use porter stemmer from nltk package. We observed for 'silly' after stemming it became 'silli'. Words like private, propose to have their e at the end chopped off due to stemming.
- Lemmatization - Lemmatization is similar to stemming in reducing inflected words to their word stem but differs in the way that it makes sure the root word/lemma belongs to the language. It is generally slower than the stemming process. Here we are desiring better results so we will be using Lemmatization over Stemming.
- Spelling Correction - Typos are common in text data and we want to correct those spelling mistakes before we do our analysis.
- Feature Engineering - Label encoded the types of personalities/ classes.
- Used TfidfVectorizer for vectorizing the posts for the model.

3.3 Data Visualization



Ploted the most common words with WordCloud.

5 Training The Model

5.1 Machine Learning Models

K Nearest Neighbor

KNN is one of the simplest forms of machine learning algorithms mostly used for classification. KNN classifies the new data points based on the similarity measure of the earlier stored data points. We performed hyper parameter tuning using GridSearchCv and have taken best parameter, p: 2, n_neighbors: 29. Achieved accuracy of `37.03`% over testing.

Random Forest Classifier

Random forests algorithm is general technique of bootstrap aggregating to tree learners. Given a training set X with responses Y, bagging repeatedly selects a random sample with replacement of the training set and fits trees to these samples.Random forest was fitted with different parameters. We find the best accuracy of 51.940069% by taking parameter, `n_estimators=500.`

XGB Classifier

XGBoost, which stands for Extreme Gradient Boosting is an implementation of Gradient Boosted decision trees. In it decision trees are created in sequential form. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. These individual classifiers then ensemble to give a more precise model. We achieved an accuracy of 57.011141% over testing.

SVM

Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. We achieved accuracy of 53.361506%.

Logistic Regression

It is a single perceptron. A common classification algorithm. We achieved an accuracy of 54.053016%.

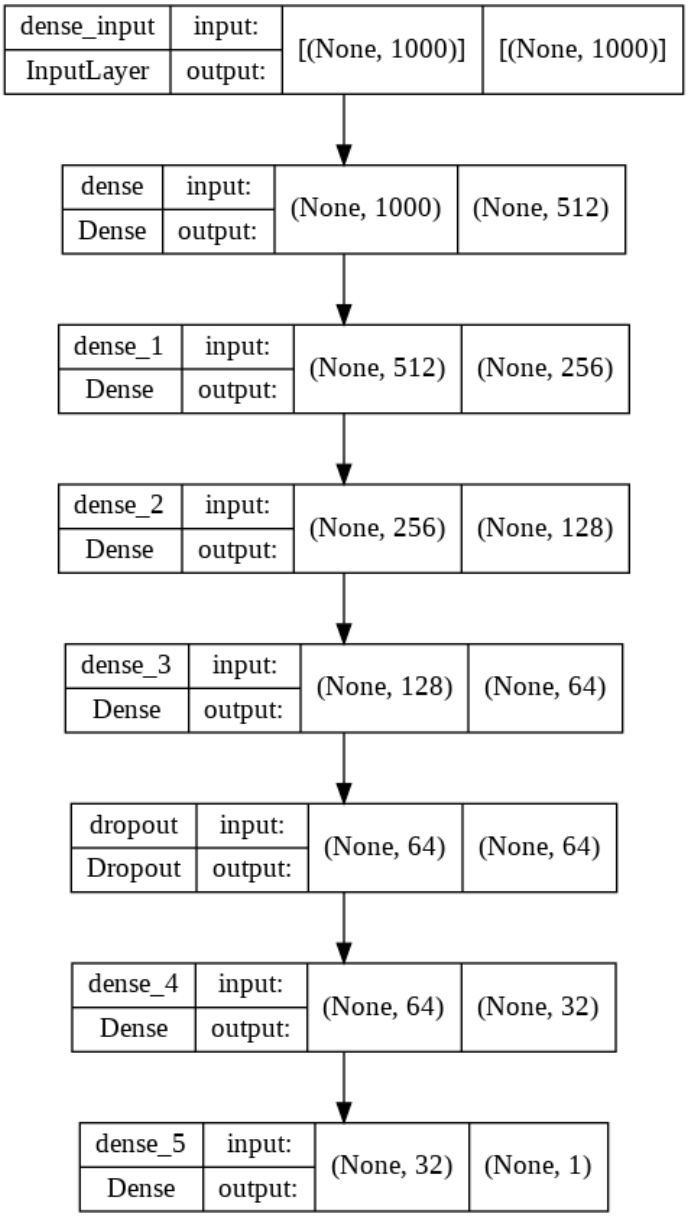
5.2 Deep Learning Models

Neural Network

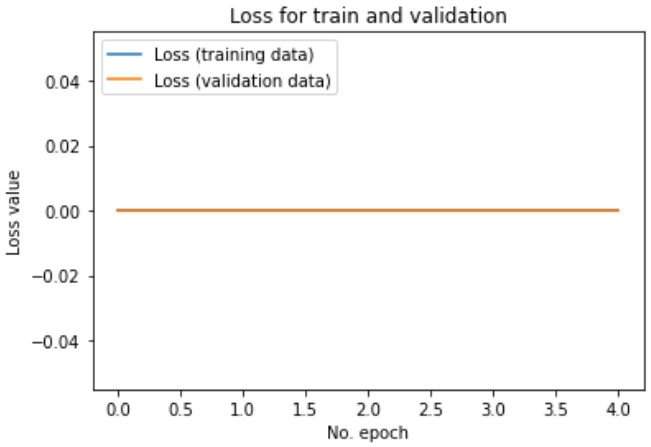
A 5 hidden layers Neural Network was trained. The activation function for all hidden layer was Relu (Rectified Linear Unit), and the last hidden layer utilized Softmax activation function. A Dropout layer with dropout rate of 0.07 was added. The Dropout layer randomly sets input units to 0 at each step during training time, which helps prevent overfitting.The model is then compiled with loss='categorical crossentropy' and optimizer='adam' and runned for 5 epochs.

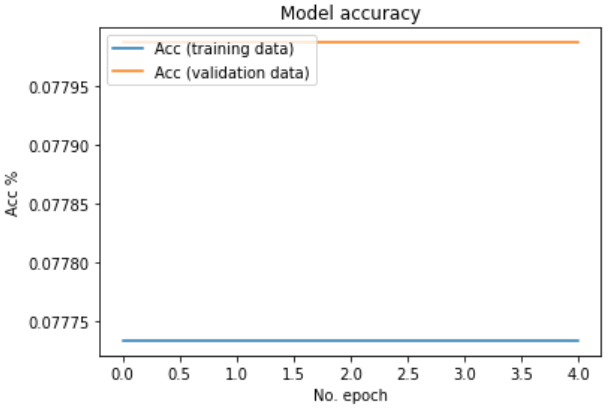
The performance of the model is very poor we could not see any increase in accuracy. And due to session clash issue was not able to train for more epochs.

CNN Architecture



Loss and Accuracy plots for train and validation data





6. Comparison Table

	Accuracies(%)
Random Forest	51.940069
XG Boost	57.011141
Logistic Regression	54.053016
KNN	37.034191
SVM	53.361506
CNN	7.800000