Predicting MBTI for Internet Users

Kyeongsoo Ko[1], Nakyeong Koh[2], Kyungsoo Han[1]

[1]Department of Statistics, Jeonbuk National University
[2]Department of Statistics, Sookmyung Women's University

Abstract

Recently, Myers-Briggs-Type Indicator (MBTI) has been used in various fields, such as being used in corporate marketing or referred to in the hiring process. This study aims to predict the MBTI type based on the words left by the user without any test. Data analysis was conducted by dividing it into a preprocessing step applying Lemmatization and Term Frequency–Inverse Document Frequency (TF-IDF) and a classification step using a support vector machine. The difference between the previous study and this study is that an intelligent optimizer that combines grid search and reinforcement learning techniques was used for hyperparameter tuning. For comparison, we present the fit results of other classification models such as logistic regression analysis and random forest.

Introduction

The Myers-Briggs-Type Indicator (MBTI) is a personality test that identifies preference trends in human perception and judgment. In MBTI, human personality types consist of a combination of four preferences, and it is emphasized that understanding the functions of the four preference measures more deeply, it helps not only one's understanding but also others' understanding to achieve a smoother human relationship.[1] Recently, MBTI has been used in various fields, such as being used in corporate marketing or referred to in the hiring process. Various studies are predicting MBTI, but only four personality types: E-I, S-N, T-F, and J-P are binary classified.[2][3][4] This study aims to predict the MBTI type based on the words left by the user without any test.

Model Selection

Data

- Used Kaggle's "MBTI Personality Types 500 Dataset"[5]
- The number of data is 106,067 and the training set and test set are divided into 8:2.
- Data consists of MBTI types of Internet users and 500 words used by Internet users.
- Figure 1 display frequency of each type of MBTI. it is unbalanced. Type I is more common than type E.
- Each class is not independent. After type I, type N is combined more frequently than type S. This means that the four binary classifiers composed of E-I, S-N, T-F, and J-P could destroy information of data.
- Therefore, we propose a classifier that classifies 16 classes at a time rather than 4 binary classifiers.

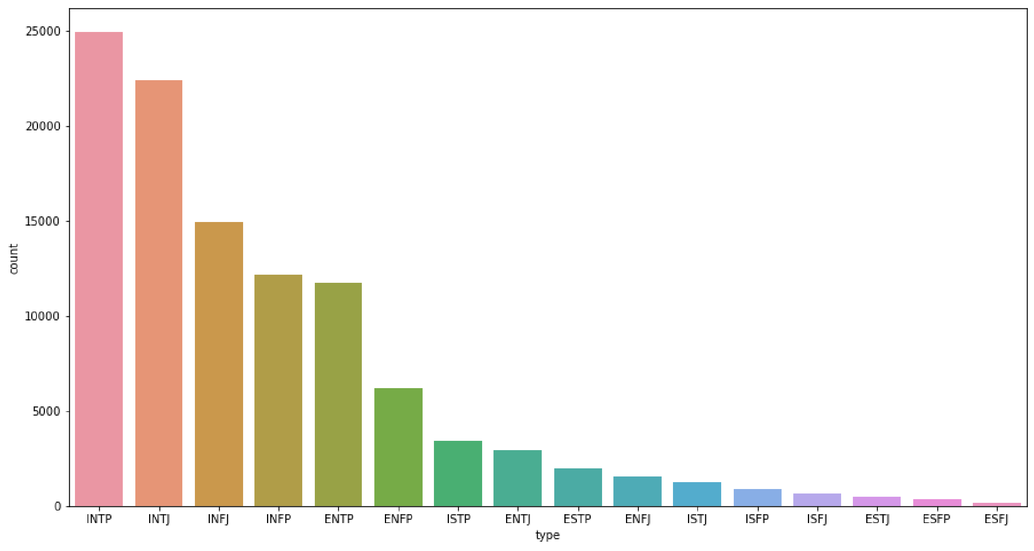

Fig1: Frequency plots for each type of MBTI.

Preprocessing

The preprocessing process proposed in this study.

Lemmatization

- Lemmatization is the process of determining the basic or dictionary form (cleaning up) for a given surface shape.[6]

TF-IDF Vectorization

- TF-IDF(Term Frequency - Inverse Document Frequency) determines how relevant a given word is in a particular document.[7]

Model

Support Vector Machine

- Proposed learning methods to solve two-way pattern recognition problems.

Validation Data

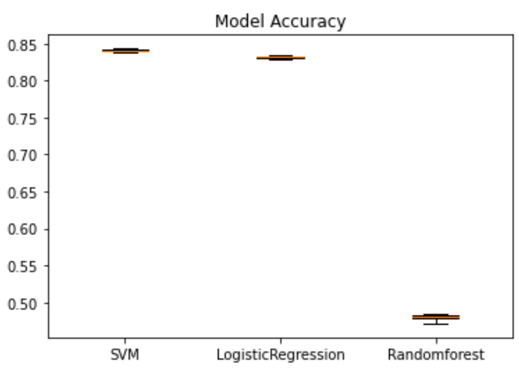

Fig2: The accuracy of each model.

Model	Accuracy
SVM	**0.8441**
Logistic Regression	0.8383
Random Forest	0.4871

Table1: The accuracy of each model.

- When fitting the model, consider different models such as support vector machines, logistic regression, and random forests.
- The accuracy of the support vector machine was the highest at 84.41%.

Hyperparameter Tuning

- SVM hyperparameter: kernel, gamma, C
- If the kernel is set to 'linear', only C can be hyperparameter tuned.

Intelligent Optimizer

- Gridsearch is a simple and powerful tool for finding hyperparameters. However, grid search is an expensive method when examining all ranges at once.
- It is usually necessary to implement an algorithm that gradually reduces from a wide-range to a narrow-range.
- This study proposes an intelligent optimizer that combines grid search and reinforcement learning techniques.

Grid Search

- Navigation to find the highest performing hyperparameters by sequentially entering the values that can be placed in the model hyperparameters.
- The results of the grid search are as follows and determine that there are optimal hyperparameters between 0.1 and 1.0.

Hyperparameter	Accuracy
C = 0.1	0.8353
C = 0.5	**0.8444**
C = 1.0	0.8407

Table2: The accuracy of each model.

Reinforcement Learning(using DQN)

- For narrow-range search, Deep Q-Network Agents (DQN agent), one of the reinforcement learning methods, is employed.
- In reinforcement learning, the agent learns the optimal policy by interacting with the environment.
- In this study, the agent is used as an optimizer for tuning hyperparameters.
- We set the state space $\mathcal{S} = [0.1, 1.0]$ as a range of hyperparameters to be optimized.
- The action a_t is set to a value between $(-0.1, 0.1)$. The environment returns the next state $s_{t+1} = s_t + a_t$ and reward r_t according to the agent's action. The reward used the accuracy of the model.
- Figure 3 shows the results of learning with DQN. The more the agent is trained, the higher the accuracy.

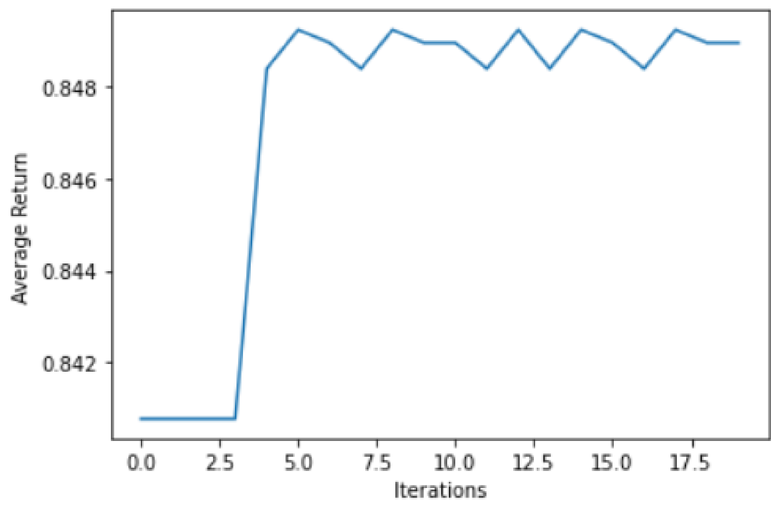

Fig3: The learning process of DQN agent.

Test Data

The results of fitting the model to the test data are as follows.

Hyperparameter	Accuracy
C = 0.246	0.8488

Table3: The accuracy of final model.

Reference

[1] 김정택, 심혜숙. (1990). 성격유형검사 (MBTI) 의 한국 표준화에 관한 일연구. *한국심리학회지: 상담 및 심리치료*, 3(1), 44-72.
[2] Amirhosseini, M. H., Kazemian, H. (2020). Machine learning approach to personality type prediction based on the myers–briggs type indicator®. *Multimodal Technologies and Interaction*, 4(1), 9.
[3] Ontoum, S., Chan, J. H. (2022). Personality Type Based on Myers-Briggs Type Indicator with Text Posting Style by using Traditional and Deep Learning. *arXiv preprint arXiv:2201.08717*.
[4] Hernandez, R. K., Scott, I. (2017). Predicting Myers-Briggs type indicator with text. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*.
[5] MBTI Personality Types 500 Dataset. Kaggle. https://www.kaggle.com/datasets/zeyadkhalid/mbti-personality-types-500-dataset
[6] Kanerva, J., Ginter, F., Salakoski, T. (2021). Universal Lemmatizer: A sequence-to-sequence model for lemmatizing Universal Dependencies treebanks. *Natural Language Engineering*, 27(5), 545-574.
[7] Ramos, J. (2003). Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning* (Vol. 242, No. 1, pp. 29-48).

LaTeX TikZposter