**Accredited Ranking SINTA 2**
Decree of the Director General of Higher Education, Research and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

**JURNAL RESTI**
**(Rekayasa Sistem dan Teknologi Informasi)**
Vol. 7 No. 5 (2023) 1140 - 1146 ISSN Media Electronic: 2580-0760

# Personality Detection on Reddit Using DistilBERT

Alif Rahmat Julianda[1], Warih Maharani[2]
[1,2]Informatics, School of Computing, Telkom University, Bandung, Indonesia
[1]alifrahmatjulianda@student.telkomuniversity.ac.id, [2]wmaharani@telkomuniversity.ac.id

***Abstract***

*Personality is a unique set of motivations, feelings, and behaviors humans possess. Personality detection on social media is a research topic commonly conducted in computer science. Personality models often used for personality detection research are the Big Five Indicator (BFI) and Myers-Briggs Type Indicator (MBTI) models. Unlike the BFI, which classifies personalities based on an individual's traits, the MBTI model classifies personalities based on the type of the individual. So, MBTI performs better in several scenarios than the Big Five model. Many studies use machine learning to detect personality on social media, such as Logistic Regression, Naïve Bayes, and Support Vector Machine. With the recent popularity of Deep Learning, we can use language models such as DistilBERT to classify personality on social media. Because of DistilBERT's ability to process large sentences and the ability for parallelization thanks to the transformer architecture. Therefore, the proposed research will detect MBTI personality on Reddit using DistilBERT. The evaluation shows that removing stopwords on the data preprocessing stage can reduce the model's performance, and with class imbalance handling, DistilBERT performs worse than without class imbalance handling. Also, as a comparison, DistilBERT outperforms other machine learning classifiers such as Naïve Bayes, SVM, and Logistic Regression in accuracy, precision, recall, and f1-score.*

Keywords: *personality detection; reddit; distilBERT*

## 1. Introduction

Personality is a unique set of motivations, feelings, and behaviors humans possess. Personality strongly influences a person's life, life choices, attitudes, health, and desires[1]. Every person has a different personality. There have been many studies on personality in an individual or a group of individuals. Personality detection on social media is a research topic that is widely done in Computer Science. Shantika *et al.* performed research on detecting the Big Five personalities of Twitter users, where from 549,151 tweets and 287 Twitter accounts, the majority of users have an Openness personality 91 users while the least personality group of users is the Neuroticism personality with 43 Twitter users [2]. Besides the Big Five personality trait, personality can also be identified using the Myers-Briggs Type Indicator (MBTI) model. Extraversion-Introversion (E/I), Sensing-Intuition (S/N), Thinking Feeling (T/F), and Judging-Perceiving (J/P) are the four binary classes that define MBTI which are then combined into 16 personality types, while BFI defines a person's personality based on 5 scalars namely Extraversion, Neuroticism, Agreeableness, Conscientiousness, and Openness [3]. Unlike the Big Five model which classifies personalities based on an individual's traits, the MBTI classifies personalities based on the type of individual themselves [4]. The MBTI personality model performs better in several scenarios than the Big Five personality model [3].

Most of the research on personality detection is conducted on Twitter social media because it has a public API (Application Program Interface) that makes it easy for researchers to get quality data [5]. However, only some studies on personality detection are conducted on the online forum Reddit. According to Proferes et al., Reddit has started to approach Twitter for ease of collecting data and providing additional benefits for research [6]. For example, the structure of subreddits makes it easier for researchers to search for data based on topics relevant to their research. Topics obtained on Reddit are also of higher quality because of the moderators and rules that must be obeyed in a subreddit [7].

Personality detection on social media often uses machine learning and deep learning methods to classify sentiment. From study about personality detection on Twitter social media conducted by Ayu and Maharani, the IndoBERT method has the highest accuracy value among the two classification scenarios

Accepted: 03-07-2023 | Received in revised: 28-09-2023 | Published: 01-10-2023

applied (0.46 and 0.46) compared to ordinary machine learning methods such as Logistic Regression (0.46 and 0.42), AdaBoost (0.35 and 0.23), and Decision Tree (0.31 and 0.38) the data used in this study is crawled using the Twitter API and gathered a total of 258 data with questionnaire to label each data to it's Big Five personality type [8]. Zenico et al [9] researched the big five personalities in Twitter users using Term Frequency Inverse Document Frequency (TF-IDF) and Logistic Regression. The data were crawled manually from 143 users who filled out the questionnaire. The crawled data gathered are 351.197 tweets. With social features and the influence of feature attributes from the Twitter scenario, 69% accuracy was obtained from the evaluation. As for the TF-IDF scenario with the social approach, the accuracy obtained rose to 76.20%.

BERT, or Bidirectional Encoder Representation from Transformers, is a Natural Language Processing (NLP) language model designed to pre-train deep bidirectional representations of unlabeled data. It allows pre-trained BERT models to be fine-tuned for diverse NLP tasks [10]. Additionally, because of BERT's ability to process large sentences and the ability for parallelization thanks to the transformer architecture [11]. Therefore, BERT has an advantage over the other deep learning models, such as RNN and LSTM, on NLP tasks that struggle with large sentence processing and parallelization. However, BERT also has drawbacks in terms of performance because it has many parameters [12]. In 2019 Victor Sanh, et al conducted research to do knowledge distillation on BERT which was later named DistilBERT with a 40% smaller size and 60% faster performance than BERT and still held 97% language knowledge capability than the original BERT language model [13]. Therefore, with the smaller size and faster performance of DistilBERT compared to BERT while regaining 97% language knowledge of BERT and while previous research mainly uses conventional machine learning methods such as Logistic Regression, AdaBoost, Decision Tree, and BERT. In this study, we conduct MBTI personality detection on Reddit using DistilBERT.

## 2. Research Methods

The system design process in this study begins with data acquisition. After the data is obtained, preprocessing and splitting of the data will be carried out before training on the DistilBERT model that was built. Then the testing process will be carried out on the model that has been trained and analyzing the results of the testing process. For more details, see Figure 1.

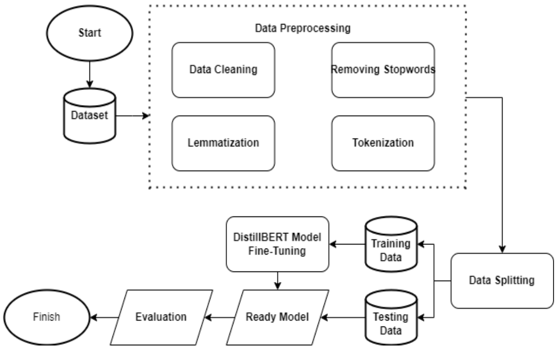

Figure 1. Research stages

### 2.1 Dataset

The dataset used in this study uses the Myers Briggs Personality Tags on Reddit Data version 0.0.1 by Dylan Storey. The dataset consists of approximately 1.7 million records of Reddit posts scraped using Google Big query [14]. The data consists of three main columns, i.e., 'author_flair_text', which is the user's MBTI personality type, 'body', which is the posts that the Reddit users posted; and 'subreddit', which denotes where the post is submitted to in Reddit. This study only used 120,000 data samples from the original dataset.

### 2.2 Data Preprocessing

Data preprocessing is done so that the data used for model implementation is efficient and high-performance. In the system built, data preprocessing is divided into four stages. First, data cleaning is a preprocessing stage to clean text data from URLs, emojis, emoticons, and special characters. Examples of data cleaning stages can be seen in Table 1.

Table 1. Data cleaning example

| Text | Cleaned Text |
|---|---|
| Drought and there's a disease spreading among the lettuce crops at the moment. Estimated about 1/3 of the yield this fall https://www.cbc.ca/news/canada/montreal/romaine-lettuce-shortage-montreal-restaurant-1.6648798l. | Drought and there's a disease spreading among the lettuce crops at the moment. Estimated about 1/3 of the yield this fall |
| I'm out of the office from June 3 to June 9. I'll be happy to answer your message once I return! ✈ | I'm out of the office from June 3 to June 9. I'll be happy to answer your message once I return! |

The second stage of data preprocessing is stopword removal. Unimportant words such as conjunctions, pronouns, articles, and prepositions are removed at this stage. An example of the removing stopwords stage can be seen in Table 2.

Table 2. Stopwords removal example

| Text | Cleaned Text |
|---|---|
| It was 7 minutes after | 7 minutes midnight. dog |

| Text | Cleaned Text |
|---|---|
| midnight. The dog was lying on the grass in the middle of the lawn in front of Mrs Shears' house | lying grass middle lawn Shears' house |
| Drought and there's a disease spreading among the lettuce crops at the moment. Estimated about 1/3 of the yield this fall | Drought disease spreading lettuce crops moment. Estimated 1/3 yield fall |

The third stage in data preprocessing is Lemmatization. Lemmatization is applied to remove inflectional endings from words to return them to their base form, known as lemmas. An example of this Lemmatization stage can be seen in Table 3.

Table 3. Lemmatization example

| Text | Cleaned Text |
|---|---|
| I like dogs. You always know what a dog is thinking. It has four moods. Happy, sad, cross, and concentrating. Also, dogs are faithful and they do not tell lies because they cannot talk. | I like dog. You always know what a dog be think. It have four mood. Happy, sad, cross and concentrate. Also, dog be faithful and they do not tell lie because they can not talk. |
| Drought and there's a disease spreading among the lettuce crops at the moment. Estimated about 1/3 of the yield this fall | Drought and there 's a disease spread among the lettuce crop at the moment. Estimate about 1/3 of the yield this fall |

The final stage of the data preprocessing is tokenization. Tokenization separates a text into tokens, which can be either words, characters, or subwords [15]. Embedding tokenized sentences allows text data to be converted into numbers that machine-learning models can read. According to Rahman et al., "Word embedding is a method of word insertion, converting words into a continuous vector of a predefined length so that it will not be limited by a larger vocabulary" [16]. BERT itself uses WordPiece Tokenizer to represent text into tokens where the first token of every sequence is a special classification token [CLS], and the last token is a separator token [SEP] [10]. The details of BERT tokenization can be seen in Figure 2. As quoted by Song et al., "Given a vocabulary, WordPiece tokenizes words by repeatedly choosing the longest prefix of the remaining text that matches a vocabulary token until the entire word is segmented. If a word cannot be tokenized, the whole word is mapped to a special token“ [17]. WordPiece tokenization distinguishes word pieces at the start of a word from word pieces starting in the middle. The latter begins with a special symbol ## (in BERT) called the suffix indicator. In data preprocessing, we used Python libraries such as re, emojis, and NLTK to preprocess the data. While for word tokenization, we used the Python transformers module to tokenize the data.

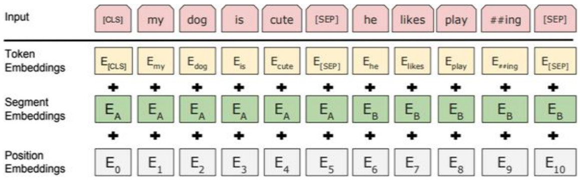

Figure 2. BERT tokenization illustration [10]

## 2.3. *Data Splitting*

Data splitting is a method of dividing data into two or more, forming a subset of data. The most common ratio of data splitting is 80:20, 70:30, and 60:40 [18]. In this case, data splitting aimed to get 80% training data from the main data to train the system built and 20% testing data from the main data to evaluate the system that has been trained.

## 2.4 Model Fine-Tuning

BERT, or Bidirectional Encoder Representation from Transformers, is a Natural Language Processing (NLP) language model designed to pre-train deep bidirectional representation of unlabeled data [8]. DistilBERT can find adjacent words because DistilBERT uses a self-attention mechanism. Self-attention is a mechanism of giving weight (attention) to each input word represented in the form of a vector by using the position of each input to extract the relationship between words in a sentence. Illustration of self-attention can be seen in Figure 3.

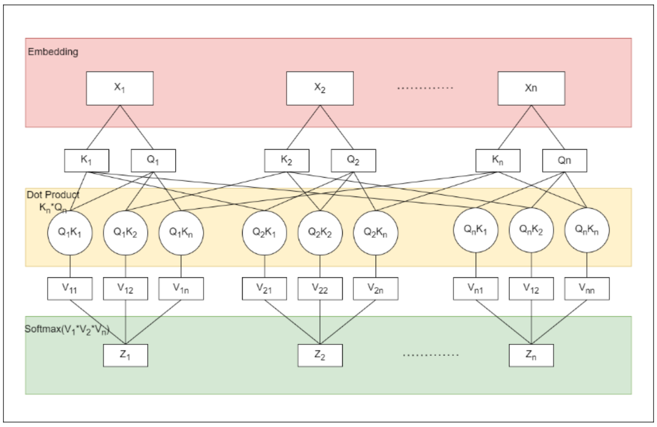

Figure 3. Self-attention illustration

The way attention works can be summarized as follows, each input vector (X) will be mapped with 3 metrics, namely query (Q), key (K), and value (V) [11]. The self-attention value/score (Z) is obtained from the softmax result between the dot product between the query, key, and value of each input vector. Then the attention value is used in the DistilBERT feed forward stage.

With BERT, we do not need to train from scratch because BERT has been trained with a large dataset for a simple NLP task of predicting missing words in a text, so we only need to freeze the weight of BERT and only train additional layers [19]. We can see how BERT is trained/fine-tuned in Figure 4. On the left is the pre-training process, and on the right is the fine-tuning process.

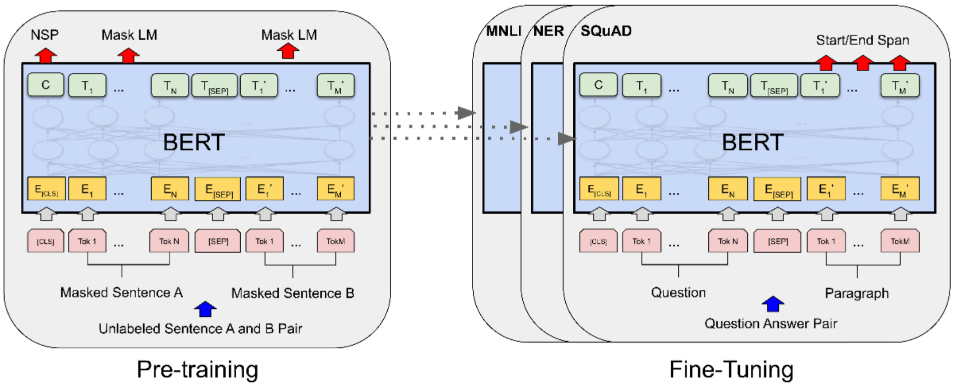

Figure 4. BERT pre-training and fine-tuning process [10]

Because of BERT's large number of parameters, it is very computationally heavy. As such, there is a smaller and faster version of BERT called DistilBERT [13]. The model used in this study is DistilBERT and will be fine-tuned with the recommended configuration of 16-32 batches, a learning rate of range between 2e-5 and 5e-5, and an epoch of 2-4 [10]. To fine-tune DistilBERT we used the Python transformers module. Then after completing the training stage, it will produce a DistilBERT classification model to test and analyze existing testing data.

## 2.5 Evaluation

The evaluation was performed to measure the model's prediction performance quality. There will be four scenarios for the evaluation phase in which the evaluation will be observed with how different preprocessing steps can affect the overall performance of DistilBERT on the prediction. The first scenario is fine-tuning the model with fully preprocessed data (cleaning the data, stop word removal, and lemmatization), the second is fine-tuning the model with only doing data cleaning as the preprocessing method, third is fine-tuning the model with cleaning data and stop words removal for the data preprocessing step. Lastly, the final scene is fine-tuning the model with data that went through data cleaning and lemmatization preprocessing steps. The model will be evaluated using a confusion matrix for multi-class classification to get the model performance value with accuracy, precision, recall, and f1-score. The multi-class confusion matrix can be seen in Figure 5.

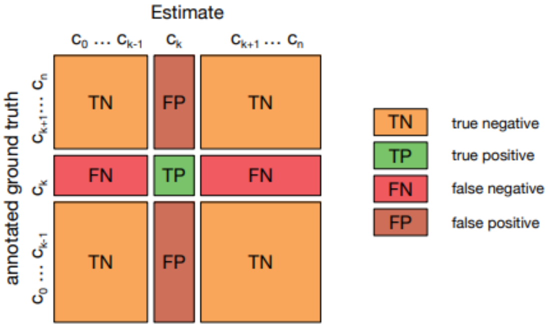

Figure 5. Multiclass confusion matrix [20]

TP or True Positive is a Positive prediction result with positive actual data, TN of True Negative is a Negative prediction result with negative actual data, FP or False Positive is a Positive prediction result with negative actual data, FN or False Negative is a Negative prediction result with positive actual data. Lastly, C is a class or label from the dataset. From Figure 4, we can do calculations for accuracy, precision, recall, and f1-score to evaluate the performance of the model, which can be seen in Formula 1, 2, 3 and 4 [21]:

$$Accuracy = \frac{\sum_{i=1}^{N} TP(C_i)}{\sum_{i=1}^{N}\sum_{j=1}^{N} C_{i,j}} \quad (1)$$

$$Precision = \frac{TP(C_i)}{TP(C_i)+FP(C_i)} \quad (2)$$

$$Recall = \frac{TP(C_i)}{Tp(C_i)+FN(C_i)} \quad (3)$$

$$F1 - Score = 2 \cdot \frac{Recall \cdot Precision}{Recall+Precision} \quad (4)$$

DOI: https://doi.org/10.29207/resti.v7i5.5236

## 3. Results and Discussions

Before the data is used for fine-tuning the model, let us first view the class distribution of the data used. The data class distribution can be seen in Figure 6.

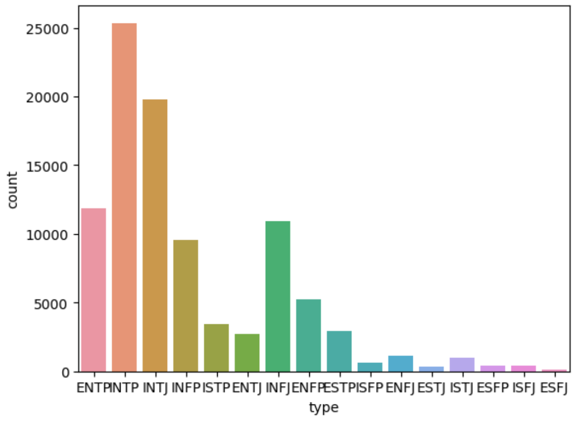

Figure 6. Class data distribution

As shown in Figure 6, the dataset used is imbalanced. Personality type INTP and INTJ has the most record in the dataset, while ESFJ has the least record on the dataset. All the model in each scenario is fine-tuned with 32 text batches, a learning rate of 2e-5, and an epoch value of 2.

### 3.1 Model Performance Evaluation

The result of the evaluation process carries on as follows. The first scenario scored an accuracy of 0.336, where the model was fine-tuned using the fully preprocessed data, where the dataset was cleaned from URLs, emojis, special characters, and stop words, and was lemmatized. The second scenario scored an accuracy of 0.347, the highest out of all the scenarios where the data used only went through the data cleaning preprocess stage. The third scenario scored the lowest accuracy, where the data used was preprocessed with data cleaning and stop words removal. Lastly, the fourth scenario scored an accuracy score of 0.345, the second highest of the whole scenario, where the data used was preprocessed with data cleaning and lemmatization. The full classification report is shown in Table 4.

Table 4. Classification report for each scenario

| Scenario | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Scenario One | 0.336 | 0.336 | 0.341 | 0.338 |
| Scenario Two | 0.347 | 0.347 | 0.349 | 0.348 |
| Scenario Three | 0.335 | 0.335 | 0.344 | 0.339 |
| Scenario Four | 0.345 | 0.345 | 0.352 | 0.348 |

The scenario with the highest precision score is scenario two, with a score of 0.347, while scenario three has the lowest precision score of 0.335. Scenario four has the highest recall score of 0.352, while the lowest is scenario one, with a score of 0.341. Lastly, scenarios two and four have the highest f1-score of the entire scenario, 0.348, while scenario one has the lowest f1-score of 0.338. Based on the result, it can be observed that stop words reduce the model's performance, and with lemmatization, the recall score of the model improves while the precision score decreases very slightly in our case.

The low score of the classification report is due to the unbalanced data, as seen in Figure 6. In addition, the training and testing data used still contain noise, such as imperfect URL deletion, the presence of non-ASCII characters, data consisting of only 1-2 words which causes a lack of context to train the model. In Table 4, the preprocessing scenario has very little effect on the model performance because DistilBERT’s transformers architecture uses all the information in a sentence thanks to its ability to process long sentences [11]. In contrast, other machine learning models can only use a little of the information obtained from a sentence because they cannot process long sentences. The most influential preprocessing stage is the preprocessing stage with the removal of stopwords, with the removal of stopwords resulting in reduced context that can be learned by DistilBERT.

### 3.2 Handling Class Imbalance

From Figure 6, the dataset used is very imbalanced. To handle this class imbalance, we used resampling techniques to overcome the data imbalance and manually predefined class weight to train the DistilBERT model rather than randomly assigning them.

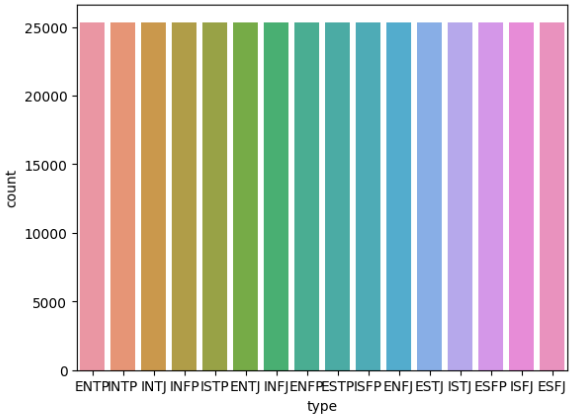

Figure 7. Class data distribution with ROS

The resampling techniques we used were random oversampling (ROS) and random undersampling (RUS). As shown in Figure 7, we oversample our dataset by randomly duplicating examples from the minority class to even out the data. While for undersampling, we do the opposite of random oversampling, where we randomly delete the majority class so that it is even across the other classes. Class data distribution with random undersampling can be seen in Figure 8.

DOI: https://doi.org/10.29207/resti.v7i5.5236

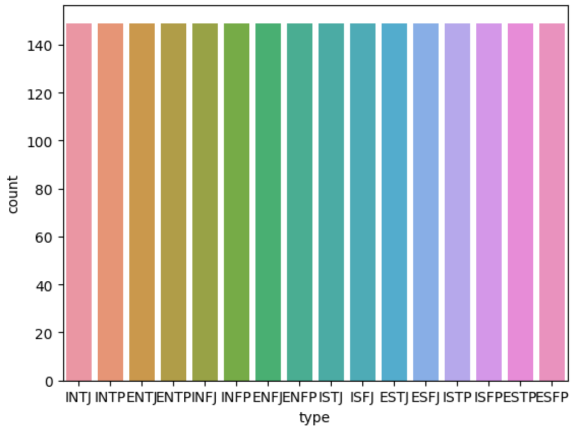

Figure 8. Class data distribution with RUS

To determine the initial class weight of the model, we used the sci-kit-learn class weight Python module to compute the class weight to even out the bias and weight in the initial learning step of the model. The result of fine-tuning the model with the resampled data and manual class weight assignment on the fourth scenario can be seen in Table 5.

Table 5. Classification report with class imbalance handling

| Scenario | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| DistilBERT | 0.345 | 0.345 | 0.352 | 0.348 |
| DistilBERT+ ROS | 0.267 | 0.267 | 0.267 | 0.267 |
| DistilBERT+ RUS | 0.027 | 0.027 | 0.032 | 0.029 |
| DistilBERT+ Class Weight | 0.242 | 0.242 | 0.242 | 0.242 |

Even with class imbalance handling, we can see that the model performs worse than the original model without resampling and with random class weight initialization. The decrease in model performance with data resampling and balanced class weight methods is caused by poor data quality. Where the random sampling technique chooses random samples from train data that have noise, causing a decrease in the performance of the model.

### 3.3 Comparing with other methods

Our fine-tuned DistilBERT model will also be compared with other machine-learning methods. The methods used include Naïve Bayes, SVM, and Logistic Regression. We used scenario four as our baseline because it is one of the best scenarios for the DistilBERT model. The performance comparison between these methods and the DistilBERT model can be seen in Table 6.

Table 6. Comparing DistilBERT with other methods

| Method | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| DistilBERT | 0.345 | 0.345 | 0.352 | 0.348 |
| Naïve Bayes | 0.270 | 0.270 | 0.303 | 0.286 |
| SVM | 0.299 | 0.299 | 0.299 | 0.299 |
| Logistic Regression | 0.291 | 0.291 | 0.291 | 0.291 |

It can be seen that the model with the highest performance is DistilBERT because DistilBERT's transformer-based model can learn data in parallel and, with the attention used by the transformer model, can map the correlation between words in a sentence so that the model can learn data efficiently compared to traditional machine learning models that directly learn data without considering the correlation between words in the sentence and without the ability of parallelization [11].

## 4. Conclusion

The scenario that produces the best performing DistilBERT model for MBTI personality detection on Reddit is the model in scenarios two and four, where the data only went through data cleaning and lemmatization preprocessing stages. It can also be observed that removing stop words can affect the model’s overall performance, as seen in scenarios one and three. The dataset used in this study is also imbalanced; because of that, we tried applying some imbalanced data handling techniques such as random oversampling, random undersampling, and adding class weight. With data imbalance handling, the model performs worse than the original model with random class weight assignments and without over/undersampling. Also, as a comparison, we compare our best model with several machine learning classifiers such as Naïve Bayes, SVM, and Logistic Regression. The comparison result shows that DistilBERT is the best-performing model compared to the other models because it can learn on the train data in parallel, and each word in the text is given a correlation weight between each other. For further research, we recommend using more data samples, better data preprocessing strategy to eliminate data noise, and a balanced dataset to fine-tune the model.

## References

[1] Y. Mehta, N. Majumder, A. Gelbukh, and E. Cambria, “Recent trends in deep learning based personality detection,” *Artif. Intell. Rev.*, vol. 53, no. 4, pp. 2313–2339, 2020, doi: 10.1007/s10462-019-09770-z.

[2] S. V. Therik, E. B. Setiawan, and U. Telkom, “Deteksi Kepribadian Big Five Pengguna Twitter,” vol. 8, no. 5, pp. 10277–10287, 2021.

[3] F. Celli and B. Lepri, “Is big five better than MBTI? A personality computing challenge using Twitter data,” *CEUR Workshop Proc.*, vol. 2253, 2018.

[4] E. J. Choong and K. D. Varathan, “Predicting judging-perceiving of Myers-Briggs Type Indicator (MBTI) in online social forum,” *PeerJ*, no. 2015, 2021, doi: 10.7717/peerj.11382.

[5] A. Karami, M. Lundy, F. Webb, and Y. K. Dwivedi, “Twitter and Research: A Systematic Literature Review through Text Mining,” *IEEE Access*, vol. 8, pp. 67698–67717, 2020, doi: 10.1109/ACCESS.2020.2983656.

[6] N. Proferes, N. Jones, S. Gilbert, C. Fiesler, and M. Zimmer, “Studying Reddit: A Systematic Overview of Disciplines, Approaches, Methods, and Ethics,” *Soc. Media Soc.*, vol. 7, no. 2, 2021, doi: 10.1177/20563051211019004.

[7] M. Gjurković and J. Šnajder, "Reddit: A gold mine for personality prediction," *Proc. 2nd Work. Comput. Model. PFople's Opin. Personal. Emot. Soc. Media, PEOPLES 2018 2018 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. T*, pp. 87–97, 2018, doi: 10.18653/v1/w18-1112.

[8] T. Ayu and W. Maharani, "Personality Detection on Twitter Social Media Using IndoBERT ( Indonesian BERT Model )," vol. 99, no. 99, pp. 1–7, 2022, doi: 10.47065/bits.v9i9.999.

[9] R. Zenico, E. B. Setiawan, and F. N. Nugraha, "Prediksi Big Five Personality dengan Term Frequency Inverse Document Frequency ( TF – IDF ) Menggunakan Metode Logistic Regression pada Pengguna Twitter," *e-Proceeding Eng. Telkom Univ.*, vol. 6, no. 2, pp. 9939–9945, 2019.

[10] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, pp. 4171–4186, 2019.

[11] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, vol. 3, no. September, pp. 111–132, 2022, doi: 10.1016/j.aiopen.2022.10.001.

[12] Q. Jia, J. Cui, Y. Xiao, C. Liu, P. Rashid, and E. F. Gehringer, "ALL-IN-ONE: Multi-Task Learning BERT models for Evaluating Peer Assessments," 2021.

[13] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," pp. 2–6, 2019.

[14] D. Storey, "Myers Briggs Personality Tags on Reddit Data," Jul. 2018, doi: 10.5281/ZENODO.1482951.

[15] M. K. Dahouda and I. Joe, "A Deep-Learned Embedding Technique for Categorical Features Encoding," *IEEE Access*, vol. 9, pp. 114381–114391, 2021, doi: 10.1109/ACCESS.2021.3104357.

[16] M. D. Rahman, A. Djunaidy, and F. Mahananto, "Penerapan Weighted Word Embedding pada Pengklasifikasian Teks Berbasis Recurrent Neural Network untuk Layanan Pengaduan Perusahaan Transportasi," *J. Sains dan Seni ITS*, vol. 10, no. 1, 2021, doi: 10.12962/j23373520.v10i1.56145.

[17] X. Song, A. Salcianu, Y. Song, D. Dopson, and D. Zhou, "Fast WordPiece Tokenization," *EMNLP 2021 - 2021 Conf. Empir. Methods Nat. Lang. Process. Proc.*, no. October, pp. 2089–2103, 2021, doi: 10.18653/v1/2021.emnlp-main.160.

[18] V. R. Joseph, "Optimal ratio for data splitting," *Stat. Anal. Data Min.*, vol. 15, no. 4, pp. 531–538, 2022, doi: 10.1002/sam.11583.

[19] D. Grießhaber, J. Maucher, and N. T. Vu, "Fine-tuning BERT for Low-Resource Natural Language Understanding via Active Learning," *COLING 2020 - 28th Int. Conf. Comput. Linguist. Proc. Conf.*, pp. 1158–1171, 2020, doi: 10.18653/v1/2020.coling-main.100.

[20] F. Krüger, "Activity, Context, and Plan Recognition with Computational Causal Behaviour Models," *ResearchGate*, no. August, 2018.

[21] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, "Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem," *Technologies*, vol. 9, no. 4, 2021, doi: 10.3390/technologies9040081.

DOI: https://doi.org/10.29207/resti.v7i5.5236