ISSN : 2355-9365 e-Proceeding of Engineering : Vol.11, No.6 Desember 2024 | Page 6245

Personality Analysis Through Social Media Based on Machine Learning

1st Aldi Putra Jaya Prasaja
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
aldiputra@student.telkomuniversity.ac.id

2nd Kris Sujatmoko
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
krissujatmoko@telkomuniversity.ac.id

3nd Thomhert Suprapto
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
thomhert@telkomuniversity.ac.id

Abstrak - Ketidaksesuaian antara keterampilan, kepribadian, dan pekerjaan dapat menjadi masalah yang signifikan dalam dunia profesional. Ketidaksesuaian ini dapat menyebabkan penurunan kinerja dan membuat pekerja lebih sulit mencapai kinerja maksimal yang diharapkan. Salah satu faktor penyebabnya adalah banyak pekerjaan yang tidak sesuai dengan kepribadian seseorang. Untuk mengatasi masalah ini, penting bagi individu untuk mengenali potensi dan kepribadian mereka secara analitis, melalui peningkatan kesadaran diri, yang membantu mengidentifikasi kekuatan dan kelemahan pribadi. Pengenalan diri tersebut dapat difasilitasi dengan menggunakan program yang dirancang untuk menilai kemampuan seseorang dan memberikan rekomendasi pekerjaan yang relevan. Situs web PSYCHEE menawarkan alat pengenalan diri tersebut, yang mengidentifikasi kesadaran diri dan keterampilan potensial pengguna. Program tersebut kemudian menghasilkan rekomendasi pekerjaan berdasarkan kualitas data pengguna. Situs web PSYCHEE memberikan akurasi prediksi sekitar 89% untuk klasifikasi teks dari X atau Twitter dan akurasi 100% untuk kumpulan data yang diproses dengan perintah obrolan GPT. Situs web ini terintegrasi dengan Chat GPT sebagai alat bantu berupa prompt obrolan, dan kunci API digunakan untuk mengaktifkan interaksi antara situs web PSYCHEE dan obrolan GPT.Hasil pengujian menunjukkan bahwa model XGBoost memiliki akurasi tertinggi. Model yang dikembangkan ini akan digunakan untuk mengklasifikasikan kepribadian MBTI berdasarkan konten media sosial dan akan diimplementasikan ke dalam aplikasi web Psyche. Aplikasi web Psyche menjalani dua tahap pengembangan: pengembangan fungsional dan pementasan. Tahap pertama berfokus pada memastikan bahwa semua fitur berfungsi dengan benar. Hasil dari fase pengujian awal ini dapat dilihat pada Tabel 2.5. Fase pengujian kedua melibatkan responden yang menguji aplikasi web Psyche dan memberikan umpan balik melalui kuesioner. Temuan dari kuesioner menunjukkan bahwa responden secara umum setuju bahwa Psyche berfungsi dengan lancar.

Kata kunci: ketidaksesuaian pekerjaan, kepribadian, kesadaran diri, situs web PSYCHEE, model XGBoost, aplikasi web Psyche.

Abstract - Mismatch between skills, personality, and work can be a significant issue in the professional world. This mismatch can lead to a decrease in performance and make it more challenging for workers to achieve their expected maximum performance. One contributing factor is that many jobs do not align with an individual's personality. To address this issue, it is crucial for individuals to recognize their potential and personality analytically, through increased self-awareness, which helps in identifying personal strengths and weaknesses.
Such self-recognition can be facilitated by using programs designed to assess one's abilities and provide relevant job recommendations. The PSYCHEE website offers such a self-recognition tool, which identifies users' self-awareness and potential skills. The program then generates employment recommendations based on the user's data qualities. The PSYCHEE website provides approximately 89% prediction accuracy for text classification from X or Twitter and 100% accuracy for datasets processed with the GPT chat prompt. This website is integrated with Chat GPT as an assistive tool in the form of a chat prompt, and an API key is used to enable interaction between the PSYCHEE website and GPT chat.
Test results indicate that the XGBoost model has the highest accuracy. This developed model will be used to classify MBTI personalities based on social media content and will be implemented into the Psyche web application. The Psyche web application undergoes two stages of development: functional development and staging. The first stage focuses on ensuring that all features function correctly. The results from this initial testing phase can be seen in Table 2.5. The second testing phase involved respondents testing the Psyche web application and providing feedback via a questionnaire. The findings from the questionnaire indicated that respondents generally agreed that Psyche functions smoothly.

Keywords: job mismatch, personality, self-awareness, PSYCHEE website, XGBoost model, Psyche web application

I. INTRODUCTION

One of the issues in Indonesia's employment sector is the mismatch or gap between job opportunities and industrial demands, which is created by a mismatch in education and skills [1]. The employment demand and supply imbalance forced many higher education graduates in Indonesia to seek jobs outside their field of study, regardless of the major they studied in college. This scenario is referred to as a horizontal mismatch, and it has become a severe issue, particularly for college graduates, because certain majors have more graduates than demand. As a result, many graduates must accept occupations unrelated to their majors [2]. A mismatch between education and employment is an issue because when an employee works in an area that does not match their educational background, they must work harder to acquire the necessary skills or competences for the job. It means that employees must learn new things and dive into different information than what they previously knew, which may make them feel uncomfortable with their employment and lead to poor levels of job satisfaction. This will also lead to diminished productivity and firm growth [3]. The definition of NLP in the Encyclopedia of Systemic NLP and NLP New Coding is patterns or programming created from the relationship between the brain (neuro), language (linguistic) and body state (body state). (Dilts, R dan Delozier. 2000. Enyclopedia of Systemic NLP an NLP) [4]. Researchers can use NLP methodologies to explore individuals' complex views and attitudes regarding the specified health services, therefore enhancing their grasp of public opinion [5]. One of the causes of the mismatch is a lack of high-quality human resources, which is also caused by individuals' lack of self-knowledge. Self-development is one strategy for developing human resources [6]. Self-development can be achieved by identifying one's own character, also known as self-awareness, to realize one's potential, limitations, and personality.

As a result, a person might use self-evaluation as an effort or step toward increasing self-quality and eventually becoming a quality human resource [7]. The Big Five personality theory, created by Paul Costa and Robert McCrae, is one of the most widely recognized methods in personality psychology. This theory highlights the five basic qualities of human personality: openness to experience, conscientiousness, extraversion, agreeableness, and neuroticism. Each of these categories represents a range of characteristics that might define individual personality differences. Extensive empirical study has demonstrated the Big Five theory's remarkable validity in a number of cultural contexts and situations, making it a significant tool in personality studies [8].

II. THEORY REVIEW

A. Myers-Briggs Type Indicator (MBTI) personality type

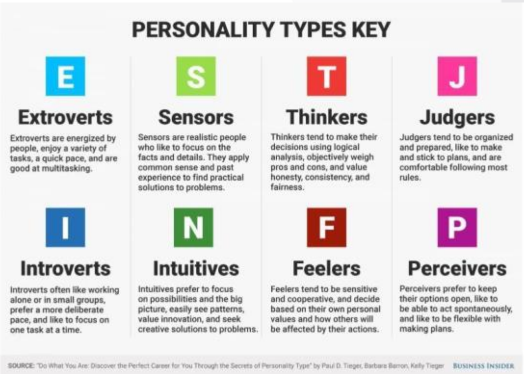

Figure 1. MBTI Based Personality

MBTI is a personality inventory. It has been implemented by career development specialists to assist a client in selecting the ideal career path as one part of a comprehensive self-assessment[11]. Based on Carl Gustav Jung's theory of personality, Katharine Cook Briggs and her daughter Isabel Briggs Myer developed the MBTI[18]. The profiling of these behaviors is diversified using a model based on four key factors, each corresponding to a major dimension of MBTI personality type theory. The Extrovert (E) vs. Introvert (I) dimension reflects how energized a person feels about their tasks or job role. Sensing (S) vs. Intuition (N) describes how a person interprets information related to their job characteristics. The Thinking (T) vs. Feeling (F) dimension determines an individual's decision-making process. Finally, Judging (J) vs. Perceiving (P) reveals the lifestyle a person adopts in their job or how it influences their job satisfaction.[9]. 16 personality types, including ESTJ, ENTJ, ESFJ, ENFJ, ESTP, ENTP, ESFP, ENFP, INFP, ISFP, INTP, ISTP, INFJ, ISFJ, INTJ, and ISTJ, can be obtained by combining the four dimensions[10].

B. GPT-3.5-Turbo-0125

GPT-3.5-Turbo-0125 is a variation of OpenAI's Generative Pre-trained Transformer (GPT) model. It employs a Transformer architecture meant to comprehend and create text in difficult situations. GPT-3.5-Turbo-0125 is trained on a massive dataset, allowing it to produce very natural and useful text in a wide range of circumstances. This model has a shorter training period than previous models because of OpenAI's optimization and computationally efficient methodologies. This model's key characteristics are its capacity to create very realistic and diverse text, as well as its strong scalability, which allows it to be employed in applications with real-time needs. Despite the model's great complexity, which necessitates significant processing resources, its ability to generalize across multiple text kinds and circumstances is exceptional, making it extremely helpful in practical applications such as virtual assistants and content production [12].

C. XGBoost

XGBoost (Extreme Gradient Boosting) is a boosting algorithm known for its speed and high performance. It works by sequentially building an ensemble of decision trees, where each new tree corrects the errors of the previous tree. XGBoost makes use of regularization techniques to avoid overfitting and improve overall model performance. The training time of XGBoost may vary depending on the size of the dataset and the complexity of the model, but it is generally considered efficient. The model excels in scalability, allowing it to handle large datasets with stable performance. XGBoost's generalization capabilities are excellent, and the model is also known to be easy to use in various machine learning applications thanks to its extensive documentation and active community. However, careful hyperparameter tweaking is necessary to produce the best results, and training time may grow with greater data volumes and more iterations [13].

D. CatBoost

CatBoost (Categorical Boosting) is a boosting technique designed by Yandex to handle categorical data without requiring considerable preparation, such as one-hot encoding. CatBoost employs specific strategies to handle category data, resulting in more efficient and accurate processing. CatBoost's training period is often shorter than that of other boosting models, due to the algorithm optimization and computational methodologies employed. The model is less sophisticated than some other boosting models, making it easier to construct and utilize. CatBoost has good generalization capabilities and can handle big datasets with consistent performance. CatBoost is extremely useful in machine learning applications, particularly when working with categorical data, making it an excellent choice for a variety of tasks [14].

E. Gradient Boosting

Gradient boosting is a boosting technique that creates a prediction model progressively. Each new model is trained to lower the error of prior models using a gradient-based optimization approach. This makes gradient boosting extremely successful for regression and classification problems. One of the most important elements of gradient boosting is its capacity to handle various types of input while producing extremely accurate models. However, training time might be lengthy, depending on the size of the dataset and the number of iterations employed. To minimize overfitting and achieve optimal performance, the hyperparameters must be carefully adjusted. While the complexity of this model can be challenging, the scalability of gradient boosting is excellent, and it can handle large datasets with stable performance, making it a strong choice for various machine learning applications [15].

III. RESEARCH METHODS

Starting with gathering and cleaning the dataset (data pre-processing) that will be used to train the machine learning model, this design testing approach proceeds. Cleaned data will be tokenize, lemmatize, and vectorize. Pre-processed data will split into test and training sets. Using all the chosen models, testing conduct on a small collection of data. Additional research is conducted leveraging an expanded collection of machine learning models. Optimized to acquire superior accuracy, machine learning models with the best accuracy on big datasets with default hyperparameters will When choosing the last model for additional estimation, one takes consideration on its training duration and accuracy value.

A. Data Pre-processing

Figure 2. Raw Data

Figure 2 shows the raw data obtained. The raw data still contains distracting characters such as URL links, unique characters, excessive spaces, numbers, and others. This will affect the model's performance when using this data for training.

```python
import re
import pandas as pd
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
import nltk
from tqdm import tqdm

# Download stopwords
nltk.download('stopwords')

# Initialize the lemmatizer and stopwords list once to optimize performance
lemmatizer = WordNetLemmatizer()
useless_words = stopwords.words("english")
```

Figure 3. Library Used for Cleaner

```python
unique_type_list = [x.lower() for x in ['INFJ', 'ENTP', 'INTP', 'INTJ', 'ENTJ', 'ENFJ', 'INFP', 'ENFP',
                                        'ISFP', 'ISTP', 'ISFJ', 'ISTJ', 'ESTP', 'ESFP', 'ESTJ', 'ESFJ']]

def pre_process_sentence(sentence, remove_stop_words=True, remove_mbti_profiles=True):
    # Remove URL links
    sentence = re.sub(r'http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|(?:%[0-9a-fA-F][0-9a-fA-F]))+', ' ', sentence)
    # Remove non-words and excess spaces
    sentence = re.sub(r'[^a-zA-Z]+', ' ', sentence).lower()
    # Remove repeating letters
    sentence = re.sub(r'([a-z])\1{2,}', '', sentence)
    # Tokenize, remove stop words, and lemmatize
    words = sentence.split()
    if remove_stop_words:
        words = [word for word in words if word not in useless_words]
    words = [lemmatizer.lemmatize(word) for word in words]
    sentence = ' '.join(words)
    # Remove MBTI personality words if required
    if remove_mbti_profiles:
        sentence = re.sub(r'\b(?:' + '|'.join(unique_type_list) + r')\b', '', sentence)

    return sentence

def clear_text(data, remove_stop_words=True, remove_mbti_profiles=True):
    cleaned_text = []
    for sentence in tqdm(data['posts']):
        # Preprocess the sentence using the pre_process_sentence function
        cleaned_sentence = pre_process_sentence(sentence, remove_stop_words, remove_mbti_profiles)
        cleaned_text.append(cleaned_sentence)
    return cleaned_text
```

Figure 4. Data Cleaner Code

```python
# Menggunakan TfidfVectorizer untuk mengubah teks menjadi fitur
vectorizer = TfidfVectorizer()
X_vect = vectorizer.fit_transform(x)

# Membagi data menjadi set pelatihan dan pengujian
X_train, X_test, y_train, y_test = train_test_split(X_vect, y, test_size=0.1, random_state=42)

# Mengubah label menjadi bentuk numerik
label_encoder = LabelEncoder()
y_train_encoded = label_encoder.fit_transform(y_train)
y_test_encoded = label_encoder.transform(y_test)
```

Figure 5. Data Splitting and Vectorizing Code

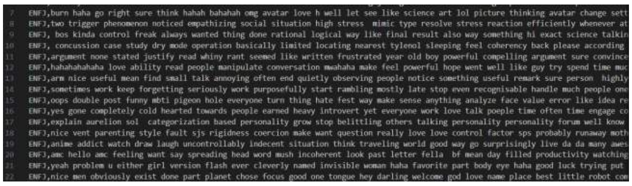

Figure 6. Pre-processed Data

The pre-processed data in Figure 6 contains data that has been cleaned of intrusive characters and applied tokenization, lemmatization, stop words, and vectorization. Pre-processed data will split into test and training sets with using ratio 90% training data and 10% test data because of the total data is 100.867 data.

B. Model Testing on Small Datasets

1. GPT-3.5-Turbo-0125

The GPT-3.5 Turbo-0125 model will also tested for classification tasks after being finetuned. Fine-tuning is accomplished by training the model on datasets partitioned into training, validation, and test sets. The dataset partition ratio is 80:10:10, with 180 data points collected from datasets from the same source. There are changes in dataset processing before fine-tuning this model. The dataset is just cleaned of distracting characters and links/urls in the data; no lemmatization is performed. This ensures that the existing text data retains context. As opposed to previous machine learning models, GPT has been trained on large data sets and recognizes text context. The goal of fine-tuning is to increase GPT performance on specific tasks, in this case MBTI personality categorization.

```python
import csv
import json

# Nama file input CSV dan output JSONL
input_csv_file = 'new_test4.csv'
output_jsonl_file = 'new_test41.jsonl'

# Template pesan sistem
system_message = {"role": "system", "content": "Identify the MBTI personality type of a people's tweets using the text provided."}

# Fungsi untuk membaca CSV dan mengubahnya menjadi JSONL
def convert_csv_to_jsonl(input_csv_file, output_jsonl_file):
    with open(input_csv_file, mode='r', encoding='utf-8') as csv_file:
        csv_reader = csv.DictReader(csv_file)

        with open(output_jsonl_file, mode='w', encoding='utf-8') as jsonl_file:
            for row in csv_reader:
                # Membuat struktur pesan
                messages = [
                    system_message,
                    {"role": "user", "content": row['posts']},
                    {"role": "assistant", "content": row['cleanlab_corrected_label']}
                ]

                # Menulis ke file JSONL
                jsonl_file.write(json.dumps({"messages": messages}) + '\n')

# Memanggil fungsi untuk konversi
convert_csv_to_jsonl(input_csv_file, output_jsonl_file)
```

Figure 7. Dataset Conversion into JSONL

Dataset that will be use to fine-tuning GPT model should be on JSONL format not in CSV. In order to convert the dataset format, code that is used can be seen in Figure 7.

```python
# Upload dataset
training_file = client.files.create(
    file=open("new_train4.jsonl", "rb"),
    purpose='fine-tune'
)

validation_file = client.files.create(
    file=open("new_val4.jsonl", "rb"),
    purpose='fine-tune'
)

print("training file id :", training_file.id)
print("validation file id :", validation_file.id)

training file id : file-IqkybtYWgqI3hLR1C45YDrHS
validation file id : file-An6CfhBXbmYUC6lZszcsdB6A
```

Figure 8. Uploading Training and Validation File

```python
train_id = 'file-IqkybtYWgqI3hLR1C45YDrHS'
val_id ='file-An6CfhBXbmYUC6lZszcsdB6A'

response = client.fine_tuning.jobs.create(
    training_file=train_id,
    validation_file=val_id,
    model='gpt-3.5-turbo'
)
```

Figure 9. Create Fine-tuning Job

```python
job = client.fine_tuning.jobs.retrieve(job_id)
print(job)

FineTuningJob(id='ftjob-iFikCrBgTkgvQfnHAWCdWXCJ', created_at=1722520452, error=Error(code=None, message=None, param=None), fine_tuned_model='ft:gpt-3.
```

Figure 10. Retrieve Fine-tune Job Information

After the dataset is converted into JSONL, training and validation data will be uploaded in order to get the file id that can be seen in Figure 8. File id is used for making fine-tuning job by selecting the model and using file id as training and validation file for the fine-tuning job that can be seen in Figure 9. Fine-tune job id is use to retrieve information about the fine-tuning job and get the fine-tune model id as shown in Figure 10.

2. XGBoost

The XGBoost model will be trained with a dataset of the same size as that used in the GPT model. The dataset used for training is 180 data with a ratio of 80% training data and 20% test data. The dataset used in this model does not need to be converted into JSONL but only in CSV form.

3. Gradient Boosting

The Gradient Boosting model also will be trained with a dataset of the same size as that used in the GPT model. The dataset used for training is 180 data with a ratio of 80% training data and 20% test data. The dataset used in this model does not need to be converted into JSONL but only in CSV form.

4. CatBoost

The CatBoost model also will be trained with a dataset of the same size as that used in the GPT model. The dataset used for training is 180 data with a ratio of 80% training data and 20% test data. The dataset used in this model does not need to be converted into JSONL but only in CSV form.

C. Model Testing on Large Datasets

The model will be trained using a larger dataset than before, with a dataset of 100,867 data. The data that will be used for training has gone through the preprocessing stage first. The models that will be tested using this large dataset are only the Gradient Boosting, XGBoost, and CatBoost models. After training the three models using the default hyperparameters, the model with the best accuracy level will be selected and the model training time will be considered. The selected model will be optimized by performing hyperparameter tuning to get the model with the best accuracy.

D. Hyperparameters Tuning

The model that has been selected based on the results of accuracy and training time will be optimized by setting the hyperparameters to get the model with the best accuracy. in this test, the hyperparameters that will be set are only iteration /n_estimator, max depth, and learning rate. The choice for only these parameters is because these three parameters have a very significant impact on the performance of the model.

IV. RESULT AND ANALYSIS

To measure accuracy, each model's training, validation, and test datasets include 180 data points. The GPT-3.5-Turbo-0125 model is tested by fine-tuning it on the same three datasets, each with a different prompt. This is done to see if there is a substantial change in the accuracy of the GPT-3.5-Turbo-0125 model with different prompts. Another case involves comparing the Gradient Boosting, XGBoost, and CatBoost models on a dataset of 100,867 data points. This is done to determine whether there is a substantial variation in accuracy outcomes for machine learning models with varying quantities of training data. Each model runs the complete dataset, and the test results are summarized in an Excel file. The acquired data is then examined to see how accurate each model is, and the findings are compared.

A. GPT Model Accuracy on Small Datasets

The GPT-3.5-Turbo-0125 model, which has been fine-tuned using the same dataset with various prompts on each dataset, may be evaluated against previously produced test data to determine the correctness of each model. The hyperparameters used to set the model during inference were chosen based on the classification requirements to achieve the highest possible accuracy.

Model ID	Temperature	Top-p	Accuracy
ft:gpt-3.5-turbo-0125:personal::9qCfG6gM	0.1	0.7	80%
ft:gpt-3.5-turbo-0125:personal::9qDygi2N	0.1	0.7	86.66%
ft:gpt-3.5-turbo-0125:personal::9qEgoeku	0.1	0.7	86.56%

Table 1. Accuracy of GPT Model

Table 1 shown the test results for the GPT-3.5-Turbo-0125 model, which was fine-tuned using three initial datasets totaling 180 data with different prompts, revealed that different prompts on the datasets used when fine-tuning the GPT-3.5-Turbo-0125 model affect the model's accuracy when measuring with test data. This occurs because GPT is a transformers architecture-based model that employs a vast number of data-driven training approaches to understand the context of the input/prompt. The fine-tuning of the GPT model produces a model that is programmed to adapt and increase its capabilities in response to a preset job. In this test, the GPT-3.5-Turbo-0125 model is programmed to accomplish the MBTI personality categorization task using text.

B. Machine Learning Model Accuracy on Small Datasets

Classification was performed using machine learning models such as gradient boosting, XGBoost, and CatBoost. In this test, the model was trained with the same dataset as was used to fine-tune the GPT-3.5-Turbo-0125 model, which had 180 data points. The data split ratio is 80% training and 20% testing from a total of 180 data points. The dataset used to train this model was lemmatized.

Classfier	Iteration	Learning Rate	Max Depth	Accuracy
Gradient Boosting	100	0.3	3	33.33%
XGBoost	100	0.4	6	38.89%
CatBoost	100	0.4	6	39.89%

Table 2. Accuracy of ML Models on Small Dataset

Testing machine learning models using Gradient Boosting, XGBoost, and CatBoost classifiers trained on the original dataset with 180 data points showed worse accuracy than the GPT-3.5-Turbo-0125 model with the same number of datasets. This is because complicated models, like XGBoost and CatBoost, can learn datasets extremely well, but on relatively small datasets, the diversity and patterns required to do correct classification are insufficient, therefore the model cannot provide high accuracy.

The MBTI consists of 16 personality types separated into four dimensions, making the categorization task quite difficult. The small dataset used for this test has enough characteristics and patterns from prior data to execute such a demanding classification assignment. Unlike the GPT model, which has been trained on massive amounts of data and employs a transformer architecture that can understand the context of the text we input and then fine-tune it to focus on a specific task, models such as Gradient Boost, XGBoost, and CatBoost learn to classify different classes using patterns and features in the data. The datasets utilized in this training are additionally lemmatized prior to training the models.

Lemmatization is conducted on the dataset during pre-processing before it is utilized to train the model. Lemmatization is used to reduce words to their simplest form. This tries to make it easier for the model to discover patterns in the data and use them as a reference to decide which patterns belong to a certain class. Computational performance also improves when the feature load in the data is lowered during lemmatization.

C. Machine Learning Model Accuracy on Large Datasets

To perform classification tests, the same machine learning models of Gradient Boosting, XGBoost, and CatBoost were used. In this test, the model was trained on a different dataset than previously. The dataset for this exam has a total of 100,867 data points, including 16 MBTI personality types and text. The data split ratio was 90% training and 10% testing, with a total of 100,867 data points. This split ratio helps the model to get the most out of the large quantity of training data while still having enough assessment capabilities to quantify the model's generalization. The dataset used to train this model is preprocessed. In this test, the model hyperparameters are set to default.

Classfier	Iteration	Time	Accuracy
Gradient Boosting	100	934m	87.81%
XGBoost	100	54m	89.06%
CatBoost	100	72m	85.73%

Table 3. Accuracy of ML Models on Small Dataset

During the training duration and accuracy of the tested models are considered, the XGBoost model delivers the greatest accuracy and the shortest training time of the three. As a result, the XGBoost model has been chosen as the model to conduct classification and incorporate into the website. The previously trained XGBoost model can be optimized again by modifying the model's hyperparameters. To achieve the greatest accuracy results, the model will be retrained using various hyperparameter settings. In this test, the hyperparameter values "learning_rate" and "max_depth" will be used to assess the difference in model accuracy outcomes. In this test, the model "n_estimators" is iterated 100 times. After doing model training with different hyperparameter values, the results and model hyperparameter values may be found in the Table 4.

n_estimators	learning_rate	max_depth	training time	accuracy
100	0.1	6	47m5s	88.69%
100	0.2	6	48m7s	88.97%
100	0.3	6	38m22s	89.06%
100	0.4	6	37m4s	89.08%
100	0.5	6	38m19s	88.63%
100	0.4	7	47m7s	88.77%
100	0.4	5	28m55s	89.13%
100	0.4	4	23m15s	89.55%
100	0.4	3	18m22s	89.35%

Table 4. Accuracy of XGBoost Model

Testing the same machine learning model on a bigger dataset produced an enormous improvement in accuracy when compared to a model trained on a relatively small dataset. This occurs because the model has sufficient training data to comprehend the patterns and characteristics required to execute complicated classification tasks, resulting in higher accuracy. The dataset utilized in this test was likewise lemmatized before training the model. The quantity of data used to train the model will have an impact on how long it takes to train. A model trained with a large dataset takes longer to train than one trained with a small dataset. Model training time is also determined by the number of iterations, learning rate, and model complexity.

Hyperparameters settings for the trained model will also have an impact on its results and performance. In this test, the model was first trained with the default hyperparameters on a dataset of 100,867 observations. Based on this test, the XGBoost model achieves the highest accuracy of 89.06% with a training time of 54 minutes. The XGBoost model was chosen for usage after assessing its accuracy and training duration. The XGBoost model is initially tuned by hyperparameter adjustment and retraining. In the testing, the model with the highest accuracy was 89.55%.

V. CONCLUSION

Based on the test results, it is determined that the XGBoost model has the highest accuracy by having 89.55% of accuracy score. The developed model will be used to categorize MBTI personalities based on content from social media. The model will be implemented into the Psyche web application to carry out the work.

REFERENCE

[1] L. Mahdiah, "Ketidaksesuaian antara Pendidikan dan Kebutuhan Tenaga Kerja Masih Besar," databoks. Diakses: 22Oktober 2023. [Daring]. Tersediapada: https://databoks.katadata.co.id/datapublish/2019/05/28/ketidaksesuaian-antarapendidikan-dan-kebutuhan-tenaga-kerja-masih-besar

[2] A. P. Yonanda dan H. Usman, "Determinan Status Horizontal Mismatch pada Pekerja Lulusan Pendidikan Tinggi di Indonesia," *Jurnal Ketenagakerjaan*, vol. 18, no. 2, hlm. 142–157, Agu 2023, doi: 10.47198/jnaker.v18i2.239.

[3] T. Hoturu, A. Dilly, G. Papuling, P. Studi, A. Bisnis, dan U. H. Namotemo, "Dampak Mismatch Pendidikan-Pekerjaan Terhadap Pengembangan Keahlian Karyawan Di Halmahera," *Jurnal Ilmu Manajemen dan Akutansi*, vol. 10, no. 2, hlm. 102–103, 2022.

[4] R. B. Dilts and J. A. Delozier, The Encyclopedia of Systemic NLP and NLP New Coding. 2000

[5] D. L. Girsang, N. A. Sidiq, and N. T. S. Elenaputri, "Analisis Sentimen Masyarakat terhadap Layanan BPJS Kesehatan dan Faktor-Faktor Pendukung Opini dengan Pemodelan Natural Language Processing (NLP)," Emerging Statistics and Data Science Journal, vol. 1, no. 2, pp. 238–249, Jun. 2023, doi: 10.20885/esds.vol1.iss.2.art24.

[6] A. Saoqillah, "PENGEMBANGAN SDM MELALUI MANAJEMEN DIRI," *Jurnal Riset Manajemen dan Bisnis (JRMB) Fakultas Ekonomi UNIAT*, vol. 1, no. 2, hlm. 145–152, Okt 2016, doi: 10.36226/jrmb.v1i2.18.

[7] I. Setyawati, "Self-Awareness Untuk Memperbaiki Diri dan Memperbaiki Hidup," Binus University. Diakses: 23 Oktober 2023. [Daring]. Tersedia pada: https://binus.ac.id/binusian-journey/2022/08/08/self-awareness-untuk-memperbaikidiri-dan-memperbaiki-hidup/#:~:text=Kenapa sih self-awareness menjadi,sebagai langkah peningkatan kualitas diri

[8] T. Hoturu, A. Dilly, G. Papuling, P. Studi, A. Bisnis, dan U. H. Namotemo, "Dampak Mismatch Pendidikan-Pekerjaan Terhadap Pengembangan Keahlian Karyawan Di Halmahera Utara," *Jurnal*

Ilmu Manajemen dan Akutansi, vol. 10, no. 2, hlm. 102, 2022.

[9] M. H. Yadav, "A Study of the Effects of an Individual's Personality and Characteristics on Job Behavior Using the Myers-Briggs Type Indicator," International Journal of Innovative Research in Engineering and Management, vol. 10, no. 3, pp. 49–52, Jun. 2023, doi: 10.55524/ijirem.2023.10.3.9.

[10] A. Wijaya and dan Henilia Yulita, "MBTI PERSONALITY TYPES for CAREER DEVELOPMENT (SMK Santo Lukas-Jakarta)," Jurnal Pengabdian dan Kewirausahaan, vol. 3, no. 2, [Online]. Available: http://journal.ubm.ac.id/

[11] S. Dee and C. Mallari, "Myers-Briggs Type Indicator (MBTI) Personality Profiling and General Weighted Average (GWA) of Nursing Students Myers-Briggs Type Indicator (MBTI) Personality Profiling and General Weighted Average (GWA) of Nursing Students Myers-Briggs Type Indicator (MBTI) Personality Profiling and General Weighted Average (GWA) of Nursing Students 2," 2017.

[12] OpenAI, "GPT-3.5-Turbo Model Card," 2023. [Online]. Available: https://www.openai.com/research/gpt-3-5-turbo

[13] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016. [Online]. Available: https://arxiv.org/abs/1603.02754.

[14] A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: unbiased boosting with categorical features," arXiv preprint arXiv:1810.11363, 2018. [Online]. Available: https://arxiv.org/abs/1810.11363.

[15] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," Annals of Statistics, vol. 29, no. 5, pp. 1189-1232, 2001. [Online]. Available: https://projecteuclid.org/euclid.aos/1013203451.