Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Software Department

Classifying Users' Personality on Social Media Using Light Gradient Boosting Machine and Optimization Techniques

A Thesis

Submitted to the Council of the College of Information Technology for Postgraduate Studies of University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology - Software

by

Ali Saadi Abbas Ali

Supervised by

Asst. Prof. Ahmed Habeeb Said Al-Azawei (PhD)

2023 A.D. **1444 A.H.**

اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ ﴿1﴾ خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ
﴿2﴾ اقْرَأْ وَرَبُّكَ الْأَكْرَمُ ﴿3﴾ الَّذِي عَلَّمَ بِالْقَلَمِ ﴿4﴾ عَلَّمَ
الْإِنْسَانَ مَا لَمْ يَعْلَمْ ﴿5﴾

Declaration

I hereby declare that this dissertation entitle "**Classifying Users' Personality on Social Media Using Light Gradient Boosting Machine and Optimization Techniques**", submitted to the University of Babylon in partial fulfillment of requirements for the degree of Master in Information Technology \ Software, has not been submitted as an exercise for a similar degree at any other University. I also certify that this work described here is entirely my own except for experts and summaries whose source is appropriately cited in the references.

Signature:

Name: Ali Saadi Abbas Al-Fallooji

Date: / / 2023

Supervisor Certification

I certify that the thesis entitled **"Classifying Users' Personality on Social Media Using Light Gradient Boosting Machine and Optimization Techniques"** is prepared under my supervision at the department of Software / College of Information Technology / the University of Babylon as partial fulfillment of the requirements of the degree of Master in Information Technology \ Software.

Signature:

Supervisor Name: Asst. Prof. Ahmed Habeeb Said Al-Azawei (PhD)

Date: / / 2023

The Head of the Department Certification

In view of the available recommendations, I forward the thesis entitled **"Classifying Users' Personality on Social Media Using Light Gradient Boosting Machine and Optimization Techniques"** for debate by the examination committee.

Signature:

Prof. Ahmed Saleem Abbas (PhD)

Head of Software Department

Date: / / 2023

Dedication

This work is dedicated to…

The martyrs of Iraq of all sects and nationalities.

The one who guides me to the way of God And God gave me

guidance on his hands,

my Shafi'i on the Day of Deen,

the Beloved Muhammad.

My father,

who left us with his body, but his spirit still flutters in the sky of our life.

My mother,

I ask Allah to protect you from all evil and I will not disappoint you.

To my beloved brothers and sisters,

who stood by me when things look very difficult.

Acknowledgements

All praise is for Allah. We praise Him, we seek His aid and we ask for His forgiveness. We seek Allah's refuge from the evils of ourselves and our evil actions. Whosoever Allah guides, no one can misguide him; and whosoever Allah misguides, there is no one who can guide him.

As humans, Allah has bestowed on us the nature to be grateful and we should thus express that gratitude not just to Allah but to the people whom we deal with as well. In many places in the Qur'an, Allah divides people as being grateful and as ungrateful to motivate us to join the camp of those who are grateful.

There are no proper words to convey my deep gratitude and respect for my supervisor, **Dr. Ahmed Habeeb Said Al-Azawei**: I am forever grateful for his patience, guidance, wise counsel and most importantly, he has provided positive encouragement and warm spirit to finish this thesis.

I also want to extend my thanks to my friends, and my second family in the maintenance of irrigation and drainage projects in Babylon: especially, the employees of the Machinery and Equipment Division for their constant support, cooperation, and encouragement at all times.

My beloved mother, thank you for being amazing in so many ways as always. You stood by me through thick and thin and I don't know how to repay you for all the sacrifices you've made.

My deepest gratitude goes to all of my family members. my brothers and sisters for the endless support. It would not be possible to write this thesis without their support.

Last but not least, I would like to thank those who helped me with a word of encouragement or scientific information that benefited me in this thesis: **DR. Raad Ghazi Hameed**, **DR. Ameer Alhaq Adil Sahib** and my friends **Mohammad Baqer Haleem**, **Ali Abd Khaleq Saiwan**, **Ali Saleem Haleem**, and all my teachers who have been so supportive along the way of doing my thesis. I thank them wholeheartedly.

Ali Saadi Abbas Al-Fallooji

Abstract

The use of social media sites (SMSs) becomes ubiquitous worldwide as the number of users is noticeably increasing. This has led to exploiting such sites by markets, business, and educational companies to deliver content that meets users' personal needs. However, this requires identifying users' personalities to respond to their individual preferences.

This study achieves three aims. First, it sought to analyze users' posts on SMSs. This leads to predicting their personality based on the Meyers-Briggs Type Indicator (MBTI) model. The study also compares the performance accuracy of different preprocessing and data mining techniques. Finally, the prediction accuracy of users' personality types is improved using several different steps. The used dataset includes 8668 records in which each raw contains fifty posts.

Four data mining techniques are applied. This includes Support Vector Machine (SVM), Logistic Regression (LR), Light Gradient Boosting Machine (LightGBM), and Long-Short Term Memory (LSTM). Two optimization methods are integrated into the proposed system.

The findings suggest that lightGBM with the application of stemming, lemmatization, and grid search optimization as well as removing stop-words outperformed other techniques. The prediction accuracies for the four personality dimensions namely, Introversion-Extroversion (I-E), Intuition-Sensing (N-S), Feeling-Thinking (F-T), and Judging-Perceiving (J-P) are 100%. Regarding the use of LSTM technique, the accuracy is 85.02%. The thesis outcomes are promising as the four dimensions of MBTI have been identified effectively. Such outcomes are also compared with earlier research on personality prediction. Thus, the thesis findings

can help SMSs providers, businesses, and educational institutions adapt their online sites based on users' posts, tweets, and comments that can be used to predict their personality behavior.

Declaration Associated with this Thesis

Some of the works presented in this thesis have been published or accepted. Appendix A refers to the paper that has been published.

❖ **First Paper:**

- **Title:** Predicting Users' Personality on Social Media: A Comparative Study of Different Machine Learning Techniques,
- **Author:** Ahmed Al-Azawei and Ali Saadi Al-Fallooji,
- **Journal:** Karbala International Journal of Modern Science,
- **Cite Score:** 4.7,
- **Quartile:** Q1,
- **DOI:** https://doi.org/10.33640/2405-609X.3262,
- **Volume:** 8,
- **Issue:** 4,
- **Pages:** 1 - 14,
- **Year:** 2022,
- **Publisher:** Elsevier.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Meaning
BERT	Bidirectional Encoder Representations from Transformers
BFT	Big Five Personality Traits
Bi-GRU	Bidirectional Gated Recurrent Units
Bi-LSTM	Bidirectional Long-Short Term Memory
BoW	Bag of Words
CatBoost	Category Boosting
CNN	Convolutional Neural Networks
DISC	Dominance Influence Steadiness Conscientiousness
DM	Data Mining
DRNN	Dilated Recurrent Neural Networks
DT	Decision Tree
EFB	Exclusive Feature Bundling
EHO	Elephant Herding Optimization
EWA	Earth-worm Optimization Algorithm
FN	False Negatives
FP	False Positives
GB	Gradient Boosting
GloVe	Global Vectors
GBDT	Gradient Boosting Decision Tree
GOSS	Gradient-based One Side Sampling
IDF	Inverse Document Frequency
KDD	Knowledge Discovery in Databases
LightGBM	Light gradient boosting machine
LR	Logistic Regression
LSTM	Long-Short Term Memory
MBO	Monarch Butterfly Optimization
MBTI	Myers-Briggs Type Indicator
MLP	Multilayer Perceptron
MS	Moth Search

List of Abbreviations

NB	Naive Bayes
NL	Natural Language
NLG	Natural Language Generation
NLP	Natural Language Processing
NLU	Natural Language Understanding
NN	Neural Networks
PLSTM	Phased Long Short Term Memory
RF	Random Forest
RNN	Recurrent Neural Network
SMOTE	Synthetic Minority Oversampling Technique
SMSs	Social Media Sites
SOTA	State-of-The-Art
SVM	Support Vector Machine
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
TN	True Negatives
TP	True Positives
Xgboost	Extreme Gradient Boosting

List of Algorithms

CHAPTER ONE

GENERAL INTRODUCTION

CHAPTER ONE

GENERAL INTRODUCTION

1.1 Background

Social Media Sites (SMSs) are used by people to build friendships with others who have similar interests in personal or vocational activities. SMSs have become a part of people's life because they use such sites to check, share, or like their friends' posts and perspectives [1]. In the last few years, there has been a huge surge in the amount of information that people have, especially in the form of textual data. People can send text messages on a lot of different websites such as Twitter, Facebook, Instagram, YouTube, and TikTok. Each day, the average time spent by a person on SMSs is two hours twenty-four minutes. This comes with the increase in the number of SMSs users which is two billion for YouTube, two billion for WhatsApp, 1.3 billion for Facebook Messenger, 1.3 billion for Twitter, and 1.2 billion for WeChat [2].

People utilize social media to express themselves on themes such as life and family, psychology, finance, society and environment, and politics [3]. A prior study found a high association between user personality behavior and his/her behavior on social media [3]. Some of the applications that can benefit from personal information are recruitment systems, personal consulting systems, and online marketing [4]. Thus, personalizing SMSs should highly rely on users' personality behavior.

A personality is a group of things that make a person unique from other people such as his/her characteristics, thoughts, feelings, and behaviors [5]. It is a term used in psychology that talks about how personality and psychological disorders affect job performance and job Satisfaction [6].

Behavior modification and modulation are the main areas that provide people with a reason to interact with each other and have balanced relationships. However, it can be hard to figure out which psychological type a person belongs to because they vary. Many different models of personality have been proposed in the literature of psychology such as the Myers-Briggs Type Indicator (MBTI) [7], Dominance Influence Steadiness Conscientiousness (DISC) [8], Strength Finder [9], and Big Five Personality Traits (BFT) [10]. In this study, the MBTI model is used because it is the most commonly used theory in adaptive technology [10, 11].

MBTI helps people understand how they work and learn. Understanding peoples' personalities represent a successful way to build relationships, be more positive, and do well [12]. MBTI includes four dimensions which are: Introversion (I) vs. Extroversion (E), Intuition (N) vs. Sensing (S), Feeling (F) vs. Thinking (T), and Judging (J) vs. Perceiving (P). The four pairs combine dimensions into 16 different types of personality, and they can be coded as a set of sixteen different types of personality. These are ESTJ, ISTJ, ESTP, ISTP, INFP, ISFP, INTJ, ISFJ, INTP, INFJ, ENFP, ENTJ, ESFJ, ENFJ, ENTP, and ESFP. The model has been widely used in business for several different reasons, such as employee profiling and/or promotion [13].

In personalizing information systems research, a lot of different ways can be used to figure out how people think. Such methods can be categorized into two groups, namely, explicit and implicit methods. The formal is to collect direct feedback from users, such as filling out a questionnaire. This method, on the other hand, has a lot of issues. Respondents may not be 100% honest with their answers [14]. Furthermore, there is a chance that some questions are not answered. Users may also understand or interpret questions wrongly. Finally,

surveys may cause people fatigue if they are too long [15]. To avoid such drawbacks, a new direction called the implicit approach is proposed in the literature to identify users' personalities based on analyzing their own comments, posts, and tweets.

Accordingly, it is important to build an effective model that can predict users' personalities based on their texts or posts on SMSs. The progress made in the domain of Natural Language Processing (NLP) can be exploited. NLP has made it possible for computers to interpret words or sentences written in human language [16, 17]. Part-of-speech (noun, verb, and adjective) and grammar structures are used in NLP [18]. In some ways of implicit analysis, users' profile pictures and other pictures that they share can also be used to figure out what kind of person they are and make information systems more personalized [19]. This can also be achieved based on a user's personal information in his/her SMSs account [20]. Other methods of implicit adaptation of information systems are: -

1. Tracking anonymous visitors based on geolocation and time, such as suggesting flight deals based on IP addresses [20].
2. Targeting the promotions of visitors based on their browsing history or shopping cart [21].
3. Detecting implicit expressions of emotion in a text [21].

1.2 Thesis Problem

A new direction called the implicit approach is proposed in the literature to identify users' personalities based on analyzing their own comments, posts, and tweets. In particular, previous research showed low to moderate accuracy in predicting users' personalities, particularly for Judging-Perceiving (J/P) and Sensing-Intuition (S/N) dichotomies of MBTI [22-24]. Thus, the prediction of users' MBTI personalities based on using their posts on SMSs still needs further research.

1.3 Thesis Questions

This thesis aims at answering this question:

Is it possible to predict personality behavior based on users' comments and posts on SMSs?

According to this question, other sub-questions are drawn:

A. Can the prediction accuracy of users' personalities on SMSs be enhanced?

B. What are the differences in predicting personality behavior on SMSs based on traditional data mining and deep learning techniques?

1.4 Thesis Objectives

The key goal of this thesis is to predict users' personalities on social media based on one of the models available for personality assessment MBTI. This can help personalize such platforms according to users' needs and preferences. To achieve this aim, the following objectives will be covered:

1. Analyzing users' comments on SMSs by following many preprocessing approaches and building a classification model to predict personality behavior based on MBTI.
2. Comparing the prediction ability of traditional machine learning and deep learning methods.
3. Enhancing the accuracy of the classifier model in predicting personality behavior and comparing the thesis findings with previous research by using optimization techniques.

1.5 Thesis Significance and Contributions

1.5.1 Theoretical implications

1. The study follows a precise preprocessing technique that helps significantly improve the prediction accuracy of the proposed model.
2. The overall findings of this thesis outperform related literature.
3. To investigate the effect of the optimization techniques on prediction accuracy, the study applies the grid and random search optimizers.
4. The overall results are compared with earlier research based on implementing and not implementing the optimizers.

1.5.2 Practical implications

1. This thesis attempts to improve the accuracy of predicting personality on social media.
2. The findings can be applied in many sectors that provide different services for individuals such as marketing or educational services.
3. Information systems or SMSs can be personalized based on users' behavior. For example, educational hypermedia systems can be adapted based on students' individual needs.
4. The thesis outcomes could also assist organizations in recruiting and selecting the appropriate personality methods.
5. Identifying users' personalities can lead to choosing the most suitable areas of work that most fit their personalities.

1.6 Thesis Challenges

There are many challenges faced during different stages of this thesis:

1. The first challenge faced is the ambiguity issue inherent in natural language, which is considered one of the most issues that needs to

be addressed in predicting the personalities on SMSs.

2. The second challenge is overfitting as the data downloaded from SMSs such as posts, comments, and tweets is generally imbalanced.

1.7 Thesis Scope

Although there are several SMSs worldwide, the Personality Café forum represents one of the most discussion and chat forums on personality [16]. This study focuses on predicting users' personalities within social media based on MBTI personality model by using the Personality Café forum. The forum includes a single language which is the English language used by many people.

1.8 Related Work

Previous studies attempt to predict users' personalities following several different approaches and techniques. After a comprehensive review that has been conducted in this research, it is found that two key directions have been adopted in related literature. The most dominant direction is based on using machine learning techniques which represents the main focus of this present research. Other studies implemented deep learning techniques to identify users' individual personalities on SMSs.

1.8.1 Machine learning

Previous literature exploits the advantages of NLP to predict users' personalities on SMSs.

In [23], the MBTI model is used to analyze the performance of different classifiers in which users' personalities are predicted based on their online text. Two supervised algorithms are applied, which are Bidirectional Encoder Representations from Transformers (BERT) and Naive Bayes (NB). The best accuracy achieved is 61 % based on the NB algorithm.

In another study [24], MBTI dimensions are predicted on social media. The employed methods are Support Vector Machine (SVM), NB, Random Forest (RF), and Logistic Regression (LR). The best accuracy was 65.4%.

Another research study suggested a new MBTI dataset based on the Reddit social media network for personality prediction [25]. The XGBoost technique is employed, in which the highest accuracy obtained was 76.1%.

In [22], the prediction of the Judging-Perceiving (J/P) dichotomy is low. This was because it is difficult to anticipate the J/P dichotomy, as it involves looking at people's actions and behaviors. The J/P binary is not related to the number of posts or comments. There is also difficulty in predicting Sensing-Intuition (S/N) dichotomy because the Intuitive type dominated the dataset and this, in turn, led to an unbalanced dataset [23].

1.8.2 Deep neural networks

In terms of using more advanced prediction techniques, deep learning has been widely applied to enhance the performance accuracy of predicting users' personalities. Earlier literature that implemented deep learning achieved an overall accuracy between 59% and 88% [26, 27].

In [26], a dataset from Twitter is used to predict MBTI dimensions on social media. The algorithms used are Dilated Recurrent Neural Networks (DRNN) [28], Bidirectional Long-Short Term Memory (Bi-LSTM) [29], Bidirectional Gated Recurrent Units (Bi-GRU) [30], and Phased Long Short Term Memory (PLSTM) [29]. The pre-trained Bi-LSTM model has the best accuracy of 81%. In machine learning, the algorithms used are Category Boosting (CatBoost) [31], Extreme Gradient Boosting (XGBoost) [32], Random Forest (RF), and Decision Tree (DT), and the pre-trained CatBoost model has the best accuracy of 85.2%.

In [27], the Big Five Personality Traits (BFT) model is used to analyze the performance of different classifiers in which users' personalities are predicted based on their posts and comments. Three algorithms are applied which are Neural Networks (NN), Gradient Boosting (GB), and SVM. The best accuracy achieved is 70.7 % based on the GB algorithm when trained on the combined features set selected by the boosting approach (X-union-B).

In [33], a comparative analysis is conducted of four predictive models: SVM, NB, and Long-Short Term Memory (LSTM) to predict MBTI dimensions on social media. The best accuracy achieved is 89.51 % for Introversion-Extroversion (I-E), 89.84 % for Intuition-Sensing (N-S), 69.09 % for Feeling-Thinking (F-T), and 67.65 % for Judging-Perceiving (J-P) based on the LSTM algorithm.

In [34], the BFT model is used to analyze the performance of different classifiers in which users' personalities are predicted based on their online text. Three supervised algorithms are applied which are SVM, Convolutional Neural Networks (CNN), and Multilayer Perceptron (MLP). The best accuracy achieved is 57% based on the CNN algorithm.

This study, therefore, aimed to solve some of the abovementioned problems in previous literature since the achieved accuracy is ranging between 59% and 88%. Table 1.1 summarizes earlier research. It encompasses the dataset used in previous literature to predict users' personalities on SMSs, the key methods used in the data preprocessing, the classification techniques, and the highest accuracy achieved for each dimension.

Table 1.1: A summary of the related work

Reference	**Dataset**	**Method**	**Feature Extraction**	
[35]	Kaggle MBTI	SVM and Neural Network	TF-IDF + LIWC	
	Lemmatization	**Stemming**	**Stop-words**	
	✓	✓	X	
	Accuracy			
	IE	**FT**	**SN**	**JP**
	84.9	88.4	87.0	78.8
	77.0	86.3	54.1	61.8
Reference	**Dataset**	**Method**	**Feature Extraction**	
[36]	Kaggle MBTI	Random Forest and KNN	TF-IDF	
	Lemmatization	**Stemming**	**Stop-words**	
	✓	✓	✓	
	Accuracy			
	IE	**FT**	**SN**	**JP**
	94.95	71.19	98.93	74.06
	93.79	96.82	53.67	76.16
Reference	**Dataset**	**Method**	**Feature Extraction**	
[23]	Kaggle MBTI	Naïve Bayes and BERT	TF-IDF + N-gram	
	Lemmatization	**Stemming**	**Stop-words**	
	✓	X	X	
	Accuracy			
	IE	**FT**	**SN**	**JP**
	52.0	62.0	57.0	57.0
	60.0	67.0	63.0	59.0
Reference	**Dataset**	**Method**	**Feature Extraction**	
[25]	Kaggle MBTI	XGBoost	TF-IDF	
	Lemmatization	**Stemming**	**Stop-words**	
	X	X	X	
	✓	✓	✓	
	Accuracy			
	IE	**FT**	**SN**	**JP**
	78.1	86.0	71.7	65.7
	79.0	85.9	74.1	65.4
Reference	**Dataset**	**Method**	**Feature Extraction**	

[37]	Kaggle MBTI	BI-LSTM and SVM	X and TF-IDF	
	Lemmatization	**Stemming**	**Stop-words**	
	X	X	X	
	Accuracy			
	IE	**FT**	**SN**	**JP**
	83.59	93.22	80.00	77.40
	82.15	87.32	80.49	72.70
Reference	**Dataset**	**Method**	**Feature Extraction**	
[38]	Kaggle MBTI	BERT + MLP	X	
	Lemmatization	**Stemming**	**Stop-words**	
	X	X	X	
	Accuracy			
	IE	**FT**	**SN**	**JP**
	78.8	86.3	76.1	67.2
Reference	**Dataset**	**Method**	**Feature Extraction**	
[39]	Twitter	BERT	X	
	Lemmatization	**Stemming**	**Stop-words**	
	X	X	X	
	Accuracy			
	IE	**FT**	**SN**	**JP**
	75.83	74.41	75.75	71.90
Reference	**Dataset**	**Method**	**Feature Extraction**	
[33]	Kaggle MBTI	LSTM	X	
	Lemmatization	**Stemming**	**Stop-words**	
	X	X	X	
	Accuracy			
	IE	**FT**	**SN**	**JP**
	89.51	89.84	69.09	67.65

Note:
BERT: Bidirectional Encoder Representations from Transformers,
MLP: Multilayer Perceptron,
LIWC: Linguistic Inquiry and Word Count,
KNN: K-nearest Neighbors.

1.9 Thesis Organization

This thesis is structured as follows:

- **Chapter Two (Theoretical Background):** This Chapter presents a general review of the main concepts of data mining, SMSs, personality analysis, preprocessing techniques, machine learning algorithms, deep learning algorithms, model evaluation, and

prediction methods.

- **Chapter Three (The Proposed System):** This Chapter shows the preprocessing steps, features generation, and features identification. Following this, the prediction models are created to predict individuals' personalities on SMSs.

- **Chapter Four (The Results and Discussion):** This Chapter presents and discusses the results of the proposed system. It also explains the performance evaluation and the differences between the findings of the machine and deep learning algorithms.

- **Chapter Five (Conclusions and Future Work):** This Chapter summarizes the key concepts and findings of this thesis and highlights possible future directions for further research.

CHAPTER TWO

THEORETICAL BACKGROUND

CHAPTER TWO
THEORETICAL BACKGROUND

2.1 Overview

This Chapter explains the basic concepts related to the theoretical background of data mining, social media sites (SMSs), personality analysis using Myers-Briggs Type Indicator (MBTI), and natural language processing (NLP). Moreover, it discusses the basic concepts of text analysis and classification, and the preprocessing technique, feature generation, feature selection, and prediction algorithms. It is noteworthy to mention that this Chapter mainly focuses on the main techniques and methods adopted in this work.

2.2 Social Media Sites (SMSs)

The use of social media sites has become an essential part of daily communication [40], as they are defined as interactive technologies that facilitate the creation and exchange of information, ideas, interests, and other forms of expression through communities virtual. Statista has estimated that approximately 4.65 billion people globally will be using social media sites (SMSs) as of April 2022 such as Facebook, Twitter, YouTube, Instagram, Telegram, TikTok and Other social media that typically features user-generated content and personalized profiles where users can communicate, share, exchange information and engage in social behavior. By 2027, the number of social media users is expected to rise to about 5.85 billion [41]. A user enjoys a tremendous global appeal and this may be because allowing the user to express his/her all opinions and ideas freely. SMSs can also be used to accepting many ideas and presenting strange and unusual ideas for individuals to be recognized [42]. Accordingly, this study exploits the widespread use of SMSs to analyze the personalities of users.

2.3 Myers-Briggs Type Indicator (MBTI)

Myers-Briggs Type Indicator (MBTI) is one of the most used psychological tools worldwide [3]. It is a model used to determine a person's personality type in how people perceive the world and make decisions, preferences, and strengths [43]. Figure 2.1 represents the dimensions of MBTI. The model attempts to determine four categories namely:

- Introversion (I) vs. Extroversion (E).
- Intuition (N) vs. Sensing (S).
- Feeling (F) vs. Thinking (T).
- Judging (J) vs. Perceiving (P).

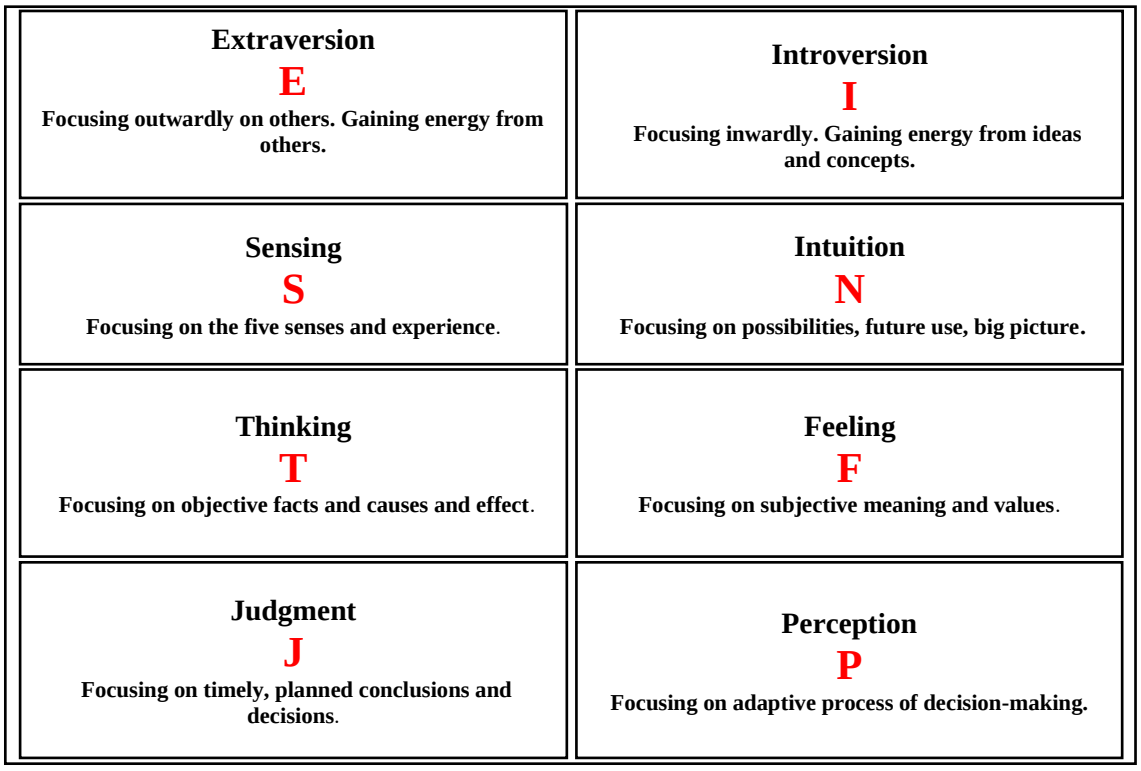


Figure 2.1: The four categories of Myers–Briggs Type Indicator

- E vs. I: Extraversion people are energized by reacting with others, taking part in activities, and are noted for responding swiftly. They are also excited about ideas, thinking, and working alone. Introversion people usually think about what they are going to do before they do it [44].

- N vs. S: Intuition people concentrate on patterns, and future possibilities, and take pleasure in abstract thought. The main focus of Sensing people, on the other hand, is on facts and their own real-world experiences [43].
- F vs. T: Feeling people are careful about taking into account persons, feelings, and different points of view. On the other side, Thinking people are logical, highly analytical, and capable of evaluating the facts [43].
- P vs. J: Perceiving people are nimble and quick to adjust to changes in their surroundings. Judging people make goals and lists, stick to a schedule, and keep track of all needs to do [45].

The four pairs combine dimensions into 16 different types of personality, and they can be coded as a set of sixteen different types of personality, As shown in Table 2.1.

Table 2.1: The sixteen possible personality types of MBTI

Type	Expansion
INFJ	**I**ntroverted- I**n**tuitive- **F**eeling- **J**udging
INFP	**I**ntroverted- I**n**tuitive- **F**eeling- **P**erceiving
ISFJ	**I**ntroverted- **S**ensing- **F**eeling- **J**udging
INTP	**I**ntroverted- **S**ensing- **T**hinking- **P**erceiving
ISTJ	**I**ntroverted- **S**ensing- **T**hinking- **J**udging
ENTP	**E**xtraverted- I**n**tuitive- **T**hinking- **P**erceiving
ESTJ	**E**xtraverted- **S**ensing- **T**hinking- **J**udging
ESFP	**E**xtraverted- **S**ensing- **F**eeling- **P**erceiving
INTJ	**I**ntroverted- I**n**tuitive- **T**hinking- **J**udging
ESFJ	**E**xtraverted- **S**ensing- **F**eeling- **J**udging
ISTP	**I**ntroverted- **S**ensing- **T**hinking- **P**erceiving
ENTJ	**E**xtraverted- I**n**tuitive- **T**hinking- **J**udging
ESTP	**E**xtraverted- **S**ensing- **T**hinking- **P**erceiving
ISFP	**I**ntroverted- **S**ensing- **F**eeling- **P**erceiving
ENFP	**E**xtraverted- I**n**tuitive- **F**eeling- **P**erceiving
ENFJ	**E**xtraverted- I**n**tuitive- **F**eeling- **J**udging

2.4 Natural language Processing (NLP)

Natural language Processing (NLP) comprises wide attention in the field of computer science and artificial intelligence [16]. NLP studies include methods and theories that can communicate between computers and humans in natural language (NL). It combines computational linguistics, mathematics, and computer science with the main goal of translating human language into commands that computers can execute [18]. It is divided into two research areas [16]:

- Natural Language Understanding (NLU): The major purpose of NLU is to understand human language by analyzing documents and extracting useful information.
- Natural Language Generation (NLG): NLG is the get of text in NL that humans can understand based on the giving of structured data, audio, text, video, and graphics.

However, dealing with NL directly is difficult since they contain noise [46]. Machine learning cannot process such noise directly. As a result, data must be cleaned before it can be used.

2.5 Data Mining Concepts

Data mining (DM) is the process of detecting patterns, correlations, and extracting unseen information within large datasets [47]. DM is an essential part of data analytics in general and one of the core disciplines in data science. Advanced analytics techniques are used to find useful information in datasets. On a more granular level, DM is a step in Knowledge Discovery in Databases (KDD) Process. Data science refers to the methodology of collecting, analyzing, and processing data [48]. DM process involves a series of steps to define a business problem as shown in Figure 2.2.

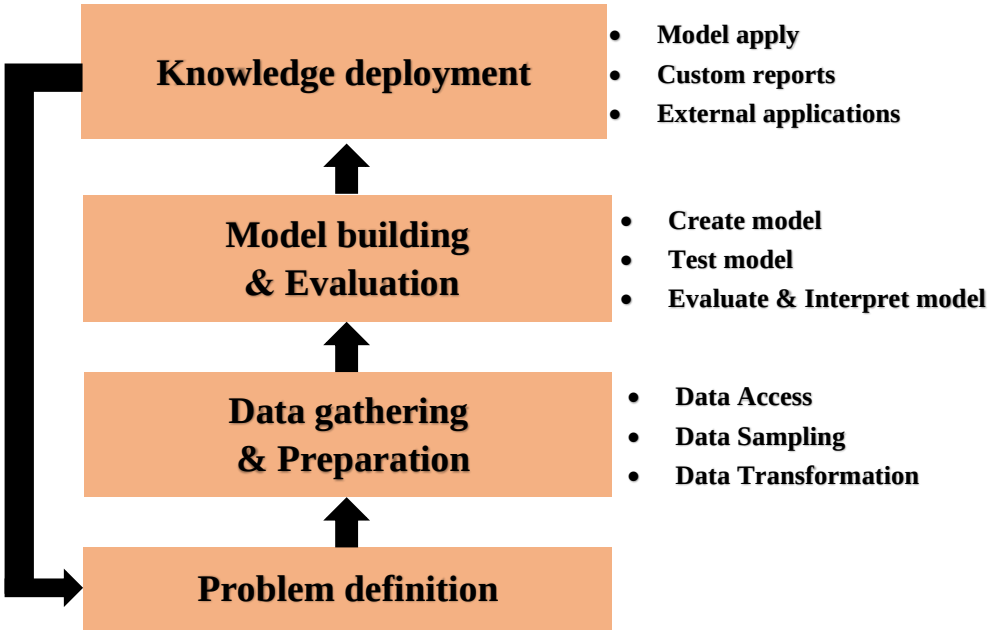


Figure 2.2: A general diagram of the data mining process [49]

In general, data mining tasks are divided into two types as shown in Figure 2.3.

1. Predictive: this type refers to the prediction of future or unknown values of another dataset of interest. It includes supervised techniques that can be used to conduct classification, regression, time series analysis, and prediction [47].
2. Descriptive: this type referes to the describtion of patterns and access new and important information based on the available dataset. The descriptive type includes unsupervised techniques of clustering, summarization, association rules, and sequence discovery [48].

A sub of data mining is text mining. Text mining is a process of extracting high-quality information from text. Text mining is used with unstructured or semi-structured datasets, whereas data mining tools are utilized to manipulate structured data from dataset [50].

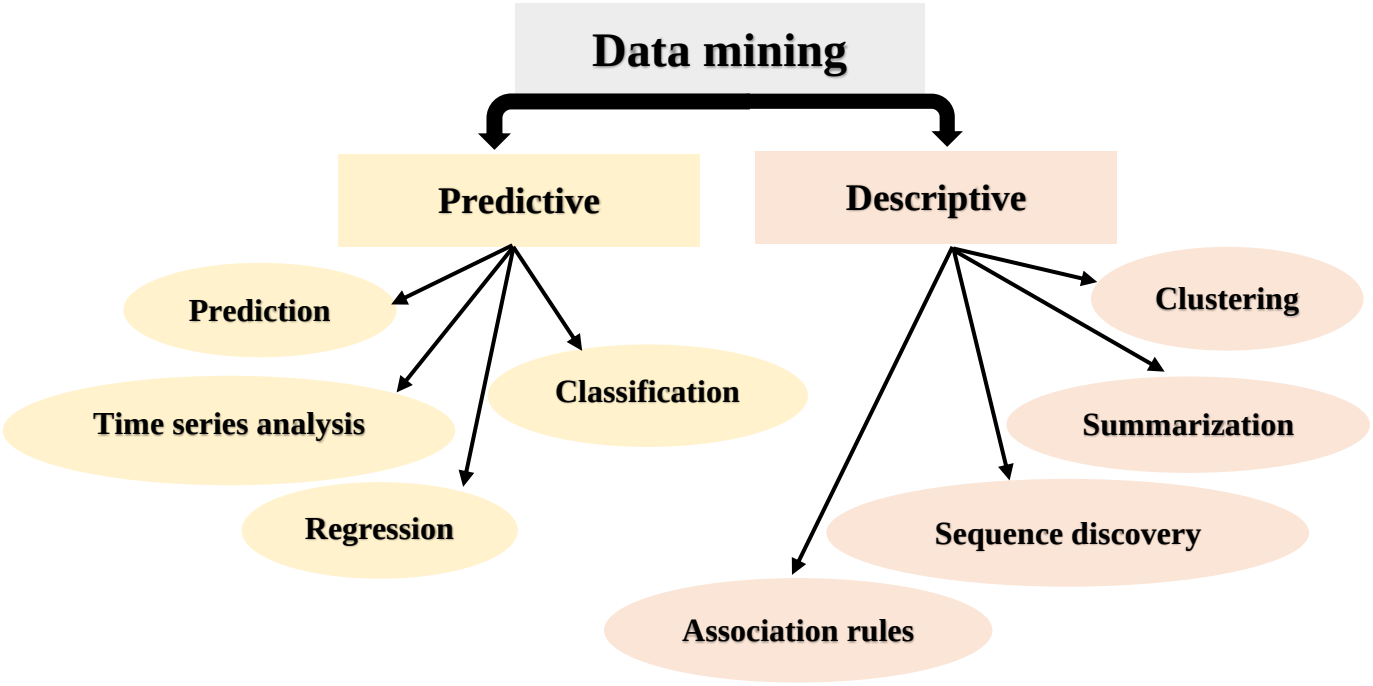


Figure 2.3: Data mining tasks and models [47]

2.5.1 Text Mining

Text mining is an interdisciplinary field that aims to extract useful information from unstructured data. Text mining is based on machine learning, information retrieval, data mining, computational linguistics, and statistics [50]. Text mining research is still ongoing due to the vast amount of information available, including email messages, technical papers, websites, blogs, and digital libraries. As a result, one important goal of text mining is to be able to extract high-quality information from text. The only difference between text mining and data mining is that text mining is used with unstructured or semi-structured datasets such as emails, HTML files, texts, and so on, whereas data mining tools are utilized to manipulate structured data from databases [50]. Text mining is concerned with NL text that is unstructured or semi-structured [51]. Several text mining techniques can be utilized for knowledge extraction, such as classification, text summarization, and clustering.

This thesis uses data mining to predict MBTI personality traits. Four

classifiers are implemented, and their findings are compared to highlight the best accuracy that can be achieved in MBTI four dimensions prediction. The implemented techniques are selected because of their popularity in earlier research and to compare the findings of this study with the previous literature [22, 52, 53]. The four classifiers were used, they are Logistic Regression (LR) [54], Support Vector Machine (SVM) [55], LightGBM [22], and LSTM [29]. Then, a deep learning technique is also used to compare the performance accuracy of traditional and deep learning techniques.

2.5.2 Text Mining Analysis

A) Text Preprocessing

Different methods of text preprocessing are used in this study. Lowercasing is one of the simplest and most effective types of text preparation. It can be applied to most text mining and NLP problems and can aid in circumstances where the dataset is not particularly large [22]. Such techniques can improve the prediction accuracy of a classifier. Another preprocessing approach is stemming which refers to the process of reducing the inflection of words. Stemming is a primitive heuristic procedure that chops off the ends of words in the hope of appropriately changing them into their base form [52]. The goal of stemming is to reduce word count, match stems accurately, memory space, and save time [56]. Moreover, lemmatization is quite similar to stemming in its overall purpose. It eliminates inflections and maps a word to its root form. It returns the changed words to their root. Another preprocessing step is stop-words elimination. Stop-words refer to a collection of widely used words in a language. Examples of stop-words in English are "a", "the", "is", "are", etc. The idea behind using stop-words is to eliminate low-information terms from the text [57]. The pretreatment process also includes:

- Removing punctuations is the removal of uninformative information such as punctuation is a necessary stage in text preprocessing. The method of removing punctuation is to delete punctuations that frequently appear and usually have no meaning. Punctuations help only the composition of sentences and how they should be read such as '#', '$', '!', '(', "", '&', '%', ')', '+', """, ',', '-', '/', '*', '.', '<', ';', '=', '@', ':', '?', '\\', '^', '>', '_', '`', '}', '[', '|', '{', ']', '~'.

- Removing links this method was utilized to remove all URLs inside posts because the URL does not carry information related to the analysis of personality based on the posts. Removing very short words.

- Removing tags from comments and posts.

- Removing words that consist of two letters or less.

Based on different preprocessing steps followed in this thesis, the dataset was split into eight states based on particular preprocessing steps. For example, one state includes stop-words, whereas the other does not contain them. In another state, stemmers are used to convert all words into their stem, while the input of classifiers is without stemming in another state. Moreover, lemmatization is used to remove the additions from the words, while in another state, words are kept without lemmatization. Another state applies stop-words, stemming, and lemmatization, whereas the original dataset was also used with stop-words and without stemming and lemmatization. In this thesis, the preprocessing steps are structured as follows:

1. Removing tags, links, punctuations, and stop-words.
2. Splitting the words in the dataset.
3. Applying stemming and lemmatization.
4. Removing words that consist of two letters or less.

B) Feature Extraction

A feature extraction procedure is used to extract attributes or significant features from a raw text in order to be prepared for feeding to machine learning or statistical methods. This method is referred to as vectorization, since it produces numerical vectors from raw text tokens. The reason is that machine learning algorithms work on numerical vectors and cannot work directly on raw text data because it includes formats that are not acceptable to these algorithms. Feature extraction methods are used in order to feed extracted features into machine learning algorithms for learning patterns that may be applied to new data points.

There are several feature extraction techniques such as Word2Vec, Bidirectional Encoder Representation from Transformers (BERT), Global Vectors (GloVe), term frequency-inverse document frequency (TF-IDF) and a bag of words (BoW). TF-IDF calculates the frequency of a term and the inverse document frequency when weighing each word, whereas BoW can be used to calculate the frequency of occurrence of a word or group of words in a text. However, the working principle of TF-IDF is based on the BoW approach [57]. Accordingly, this study adopts the TF-IDF technique according to the findings obtained.

1) *Word2Vec*

Word2Vec is used in NLP. The effectiveness of Word2Vec comes from its ability to group different vectors that have similar words. Given a large enough dataset, Word2Vec can make strong estimates about a word's meaning based on their occurrences in the text. These estimates yield word associations with other words in the corpus [58].

2) *Bidirectional Encoder Representation from Transformers (BERT)*

BERT is a pre-trained language model. It utilizes transformer model architecture to achieve State-of-The-Art (SOTA) performance for some NLP

problems. BERT model can be used with two approaches which are fine-tuning-based approach and feature-based approach. In the feature-based process, BERT represents text data as fixed feature vectors using a pre-trained model. BERT can produce vector representations that take the position and context in a sentence into account, while fine-tuning is a process that takes a model that has already been trained for one given task and then tunes or tweaks the model to make it perform a second similar task [59].

3) *Global Vectors (GloVe)*

GloVe is a technique that takes advantage of two different approaches. The first is a count-based, whereas the other is a direct prediction such as word2vec. Unlike word2vec which relies solely on local information from words with local context windows, the GloVe algorithm also combines word co-occurrence information or global statistics to obtain semantic relationships between words in the corpus. The GloVe model aims to study the vector of words in such a way that the dot product of those words is equal to the logarithm of the probability of words appearing together or the probability of their co-occurrence [60].

4) *Term Frequency-Inverse Document Frequency (TF-IDF)*

TF-IDF is a metric of determining how significant a term is in a text. There are three major implementations of TF-IDF which are in: machine learning, information retrieval, and keyword extraction/text summarization. Machine learning algorithms often utilize numeral data, to handle textual data or any NLP errand which is a sub-field of machine learning/artificial intelligence [23]. Thus, data needs to be switched to a vector of numeral data by an operation known as vectorization. TF-IDF vectorization calculates the TF-IDF weight for every word in the dataset and then puts that information into a vector. Hence, each post or comment in the dataset has its own vector, and the vector would have a TF-IDF score for every singular word in the dataset. After obtaining these vectors, they can be used

for many different aspects such as seeing if two words are similar by comparing their TF-IDF vectors. In text summarization and keyword extraction, TF-IDF is also used [61]. Because TF-IDF weights words based on how important they are, this method can be used to figure out which words are the most important and this, in turn, can help determine keywords for a dataset.

Mathematically, TF-IDF is the outcome of two measures which are Term Frequency (TF) and Inverse Document Frequency (IDF). It can be calculated based on Equation 2.1 [61] for each term.

$$TF{-}I(t, d, \mathrm{D}) = TF(t, d) * IDF(t, \mathrm{D}) \qquad (2.1)$$

where:

TF: the number of occurrences of a term t in document d.

IDF: a statistical weight to determine the importance of a term in a set of documents.

Equation 2.2 [61] can be used to calculate TF.

$$\mathrm{TF}\ (\mathrm{t}) = \text{Number of times term appears in a document} / \text{Total number of terms in the document} \qquad (2.2)$$

Equation 2.3 [61] can be used to calculate IDF.

$$IDF\ (t) = log\ \mathrm{n}\ /\ (\mathrm{df}(\mathrm{t})) \qquad (2.3)$$

where:

n: the overall number of documents in the document collection.

df (t): the document frequency of t.

After transforming texts using TF-IDF or any other method, they can be utilized by machine learning algorithms.

5) Bag of words (BoW)

The Bag of words (BoW) technique is one of the easiest representation ways in extracting features from the text. The document representation method of BoW is also known as a vector space model. The core of the

vector space model is to represent and convert each document as a vector. This vector represents the frequency of all terms in the specific document [22]. Every row represents an observation, and each feature represents a unique term.

2.5.3 Methods of Classification Techniques

A) Support Vector Classifier (SVM)

Support Vector Classifier (SVM) is a supervised machine learning that can be used to solve issues such as regression and classification. SVM is a good classification approach that can tackle linear and non-linear problems. It is also a practical learning method based on statistical learning theory, so statistical learning theory is the basis of SVM [55]. Figure 2.4 shows the separation of data by functional margin based on the support vector classifier.

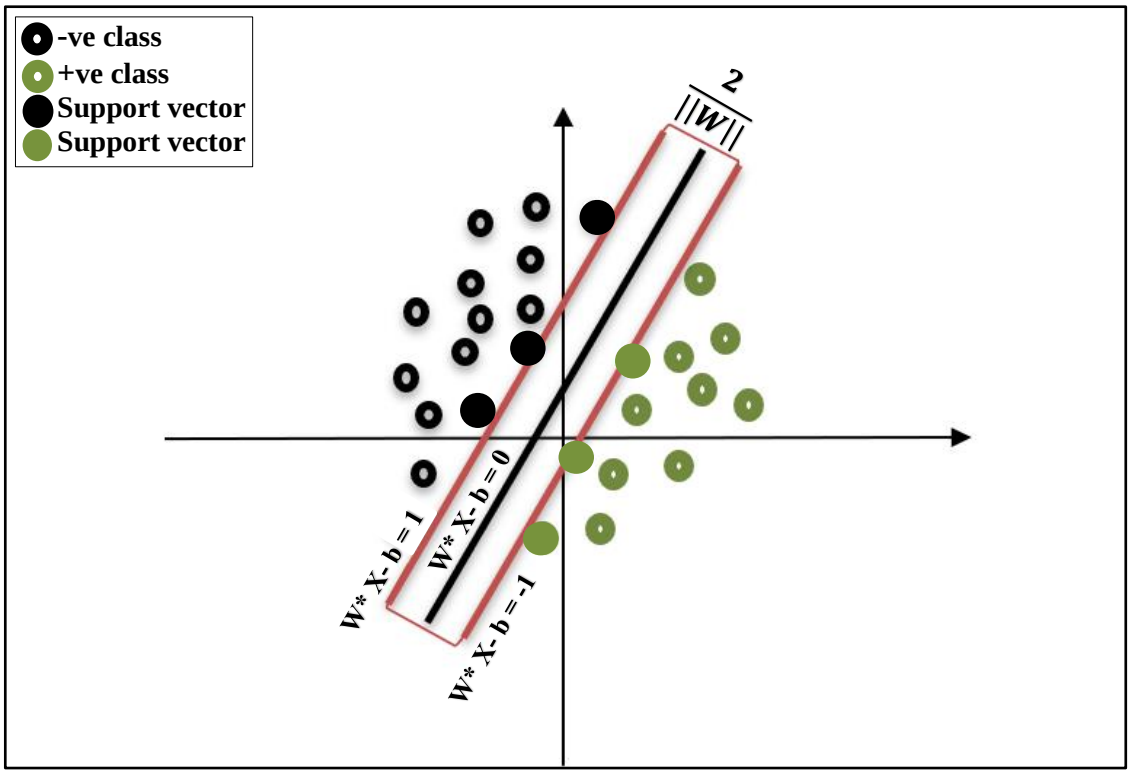


Figure 2.4: The separation of data by functional margin using SVM

Linear SVM is a classifier that seeks the hyperplane with the greatest margin since decision boundaries with large margins have lower

generalization errors than those with tiny margins, known as a maximal margin classifier. Each decision boundary is related to a pair of hyperplanes, the first of which is got by moving a parallel hyperplane away from the decision boundary until it touches the nearest support vector from the right side, and the second by moving the hyperplane until it touches the nearest support vector from the left side. The distance between these two hyperplanes is known as the classifier's margin.

The binary classification problem consists of N number of training states. Each instance is denoted by a set (Xi, Yi), where {Xi, ..., Xn} is a dataset and Yi ∈ {-1, 1} denotes its class label. There are several mathematical equations. As shown in Figure 2.4 above:

The decision boundary of the classifier can be represented as in Equation 2.4 [62]:

$$W * X + b = 0 \qquad (2.4)$$

where:

W: weight vector.

X: input vector.

b: bias.

Equations 2.5 and 2.6 below represent the decision boundaries [62]:

$$W * X + b = 1 \qquad (2.5)$$

$$W * X + b = -1 \qquad (2.6)$$

Margin is the space between these two hyperplanes which can be calculated by Equation 2.7 [62].

$$d = 2/(||W||) \qquad (2.7)$$

After choosing the parameters W and b, Equations 2.8 and 2.9 as shown below are satisfied so that the SVM model starts learning [62]:

$$W * X + b \geq 1 \text{ for } Yi = +1 \qquad (2.8)$$

$$W * X + b < 1 \text{ for } Yi = -1 \qquad (2.9)$$

All data must be correctly categorized by decision boundaries by Equation 2.10 [62]:

$$Yi(W.X + b) \geq 1, \qquad i = 1,2,3, \dots, N \qquad (2.10)$$

The SVM algorithm is a good classifier for text classification, and this is due to several reasons. Firstly, the number of irrelevant features is few. Secondly, most of the text classification problems can be linearly separable. These reasons provide a theoretical guide that SVM should perform excellently for text classification.

B) Logistic Regression (LR)

Logistic Regression (LR) is a statistical model that belongs to linear regression. It allows describing a binomial variable in terms of a collection of random variables, whether categorical or numerical. With additional knowledge about variable values that can be explained or related to that event, it is used to predict the probabilities. Several predicted variables, which might be categorical or numerical, are used in LR. The Logit model or the generic classifier of entropy is another name for LR. The threshold of 0.5 is adopted here based on previous literature [63]. This modeling technique is commonly employed in various scientific and commercial applications, and it is one of the most commonly used modeling techniques in the field of machine learning [54]. LR can be calculated based on Equation 2.11 [64]. Figure 2.5 depicts the separation of data by the decision boundary using the LR classifier.

$$P = \frac{e^{a+bx}}{1+e^{a+bx}} \qquad (2.11)$$

where:

p: the predicted value.

e: the base of the natural logarithm (about 2.72).

x: the input value.

a: the bias or intercept term.

b: the coefficient for input (x).

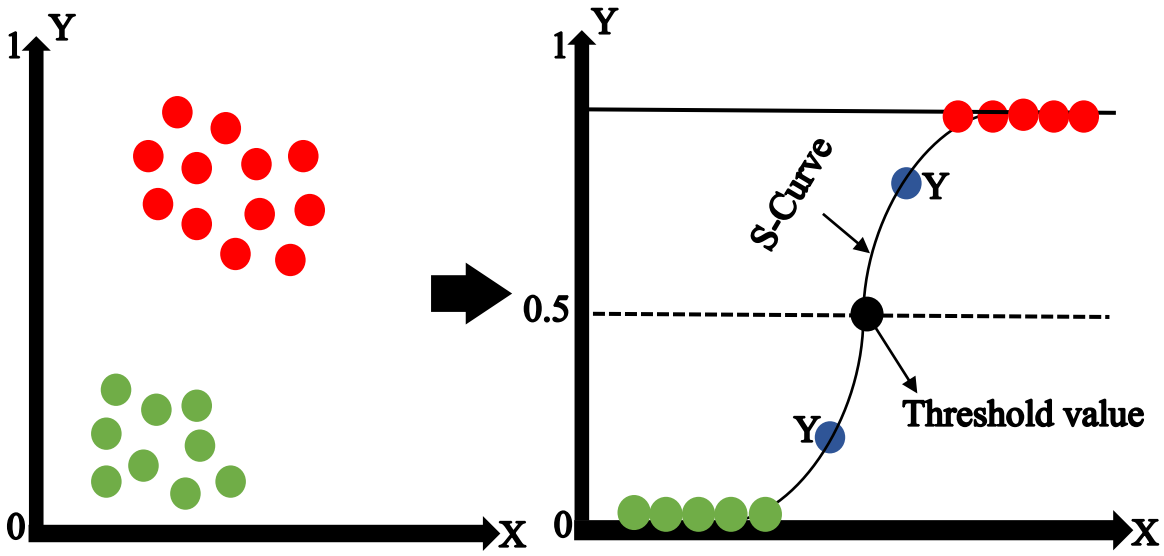


Figure 2.5: The separation of data by decision boundary using LR

C) Light Gradient Boosting Machine (LightGBM)

Light Gradient Boosting Machine (LightGBM) is a powerful machine-learning technique. It is a gradient-boosting framework that is based on a decision tree (DT) which can provide quick training and high efficiency [65]. LightGBM includes several parameters, termed hyperparameters. The hyperparameters have a significant impact on the performance of the LightGBM algorithm. They are typically set manually. The LightGBM algorithm also helps obtain low memory utilization, large-scale data handling, and high accuracy. It grows vertically as shown in Figure 2.6, whereas the other algorithms grow horizontally. Although the leaf-wise approach is superior to the level base in terms of minimizing loss and improving accuracy, it is complex and may result in overfitting [22]. Therefore, this technique has many advantages such as quick training, minimal memory usage, high model precision, supporting parallel learning, and adequate for big data [22].

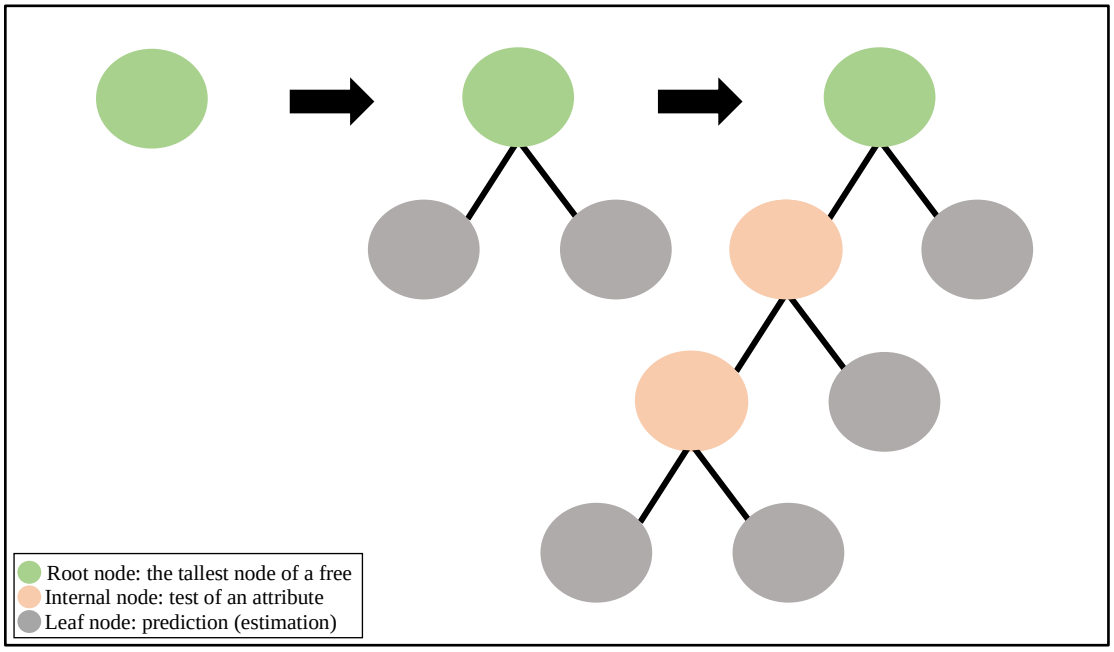


Figure 2.6: The key idea of the LightGBM algorithm [65]

LightGBM employs two new technologies: Exclusive Feature Bundling (EFB) and Gradient-based One Side Sampling (GOSS) that meets the boundaries of the graphics-based algorithm that is mainly used in all frames of the Gradient Boosting Decision Tree (GBDT). The properties of the LightGBM algorithm come from the two EFB and GOSS techniques described below. They work together to ensure that the model works effectively and provides it with a leading edge in comparison to other GBDT frameworks [66].

1. EFB for LightGBM: High-dimensional data are typically scattered, which provides an almost lossless-free way to reduce the number of features. In particular, many characteristics in sparse feature space are mutually exclusive, not taking nonzero values simultaneously. The unique features can be securely combined into a single feature. As a result, the complexity of constructing the histogram changes from O(#data × #feature) to O(#data × #bundle), while #bundle<<#feature. Thus, the speed of the training frame is improved without compromising accuracy [66].

2. GOSS for LightGBM: A variety of data instances play different roles in the calculation of information gain. This makes Under-trained instances with large gradients more effective at gaining information. GOSS retains cases with large gradients and abandons cases with small gradients at random, to maintain the accuracy of the information gain estimation to configure the sub-instance set A. This method can result in a more accurate gain than uniformly random sampling at the same target sample rate, particularly when the value of information gain has a wide range. For the remainder of the set, Ac is generated by the (1 − a)% of cases with smaller gradients, a subset B of size b*|Ac| is random. Finally, the instance is partitioned based on the expected variance gain Vj * (d) on the subset A ∪ B. As shown in Equation 2.12 [66].

$$Vj*(\mathrm{d})=\frac{1}{n}\left(\frac{(\sum xi\in Al\ gi+\frac{1-a}{b}\sum xi\in Bl\ gi)^2}{n_l^j(d)}+\frac{(\sum xi\in Ar\ gi+\frac{1-a}{b}\sum xi\in Br\ gi)^2}{n_r^j(d)}\right) \quad (2.12)$$

where:

Al = {xi ∈ A : xij ≤ d}.

Ar = {xi ∈ A : xij > d}.

Bl = {xi ∈ B : xij ≤ d}.

Br = {xi ∈ B : xij > d}.

d: is the point in the data where the split is calculated to find the best gain invariance.

1 − ab: this coefficient is used to normalize the gradient sum over B back to the size of Ac.

D) Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN). It can find the hidden layer of each cell. LSTM is designed to store the information of the previous cell, LSTM method is used by classifying long-term data by storing it in memory cells. The

LSTM method consists of four main components, namely: cell, input gate, output gate, and forget gate [29]. The cell remembers values at random time intervals, and the three gates control the flow of data into and out of the cell. This can provide a better understanding of the context. These features are perfect for NLP problems [33]. The mesh structure is machined to produce optimum accuracy. In general, the training set can have a different number of cells in layers, but this study consists of six layers which are the embedding layer, four layers for the long-term memory layer, and one dense layer. Equations 2.13 and 2.14 clarify the appeal process and the general architecture of LSTM layer [38]. Figure 2.7 shows the key concepts of the LSTM model.

$$\text{layer} = \text{LSTM}(\text{NUM}, \text{AC}, \text{b}, \text{I}, \text{FB}, \text{R}, \text{C}, \text{D}) \qquad (2.13)$$

where:

NUM: number of units in the layer;

AC: activation;

b: use bias;

I: initializer;

FB: unit forget bias;

R: regularize;

C: constraint;

D: dropout;

LSTM: is a function from TensorFlow library that constructs an LSTM layer as output;

$$f(x) = \text{layer}(X, \text{MA}, \text{T}, \text{IS}) \qquad (2.14)$$

where:

X: Input;

MA: mask;

T: training;

IS: Initial State;

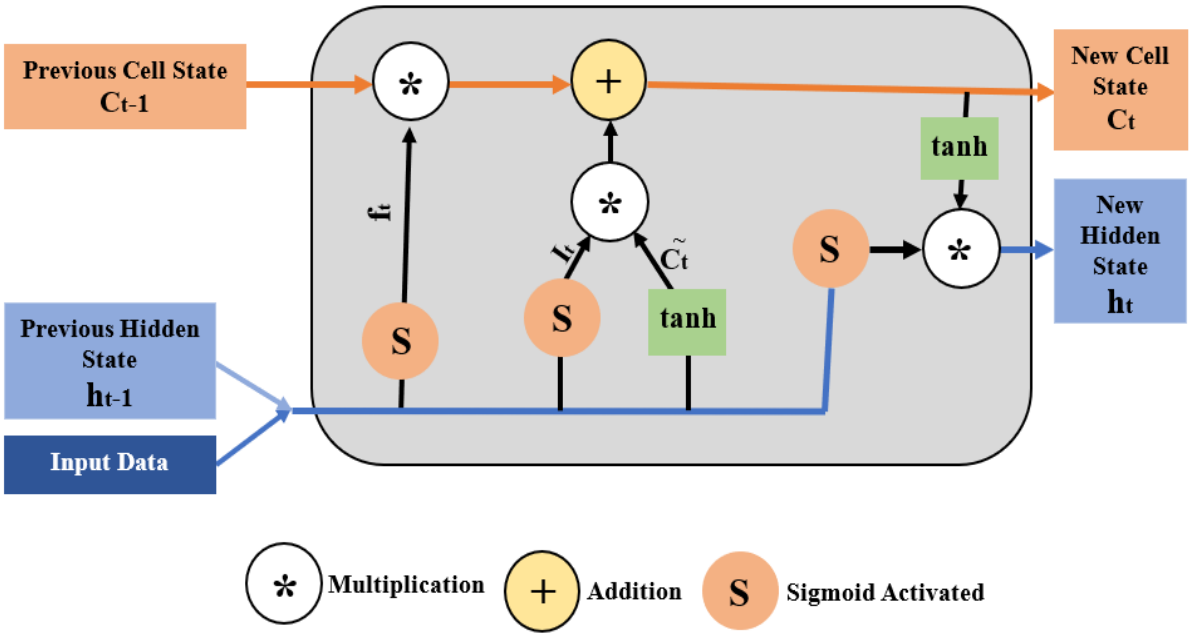


Figure 2.7: The LSTM architecture of a single neuron [67]

2.6 Dealing with unbalanced data

Many techniques can be used to deal with unbalanced data such as random oversampling of minority class, near miss-based undersampling, and Synthetic Minority Oversampling Technique (SMOTE). In this thesis, SMOTE is used because it is one of the dominant methods in such a case [68]. It is a machine learning method for solving the problem of the imbalanced dataset which is caused by having few examples of the minority class. Figure 2.8 shows the distribution of the dataset's labels in which '1' indicates the target numbers for the dimensions Extroversion, Feeling, Sensing, and Perceiving, while '0' refers to the other dimensions namely, Introversion, Intuition, Thinking, and Judging. This can be simply achieved by taking a copy of the minority-class examples while training the dataset before fitting the model. This helps balance the distribution of the class, but this does not provide any additional information for the model. An improvement in the repetition of examples from the minority class is the aggregation of new examples from the minority class [68].

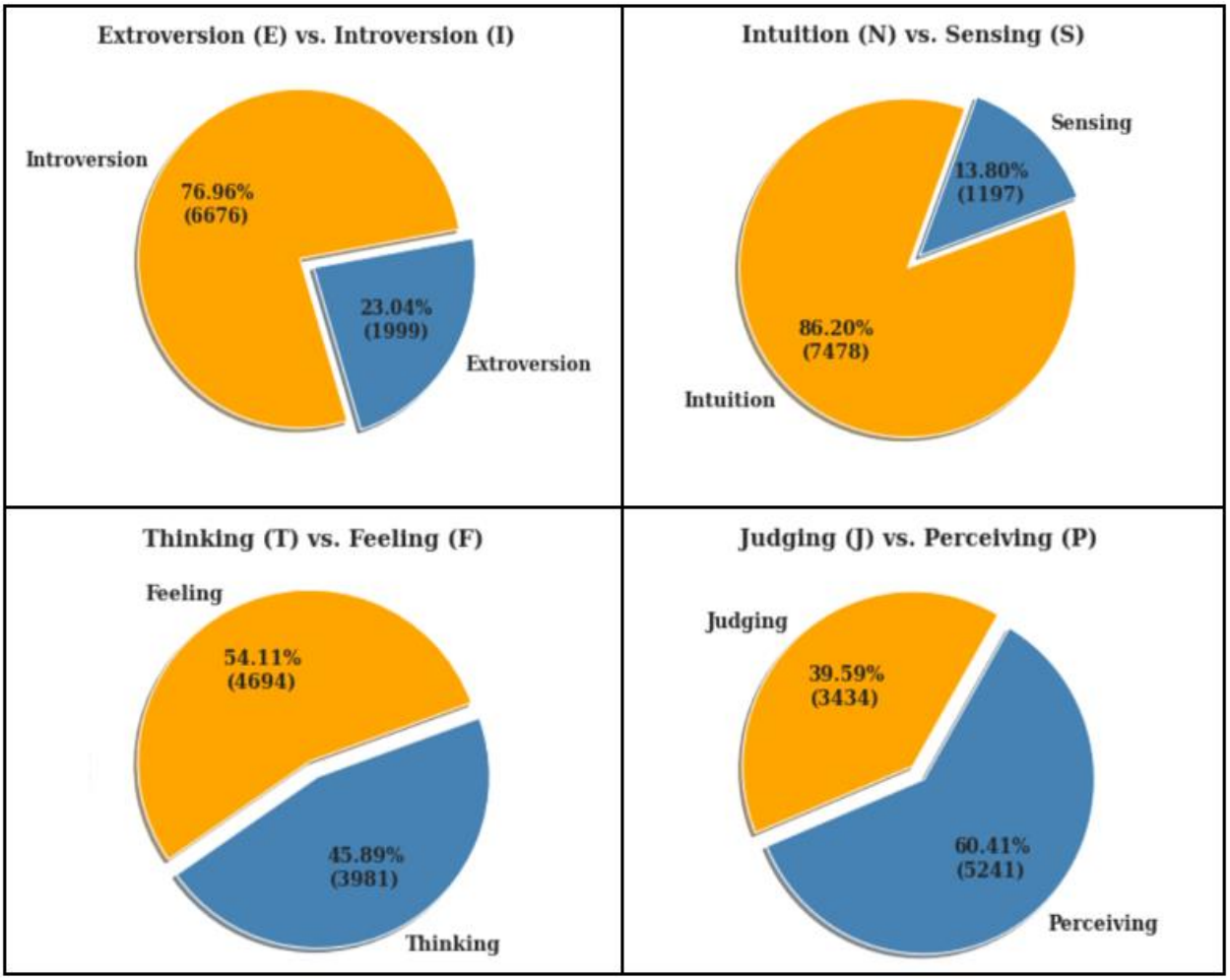


Figure 2.8: The distribution of the target classes

2.7 Methods of Optimizations

Two optimization methods are used to find the best parameters in this study namely, the grid search and the random search optimizers.

1) The grid search optimizer method is widely used for determining the appropriate hyperparameters of a classification model. It can reach the ideal solution if there are enough grid nodes as shown in Figure 2.9 [69]. By using cross-validation, the dataset is randomly split into test and training sets in the grid search optimizer method [69]. Finding the best hyperparameters can significantly affect prediction accuracy.

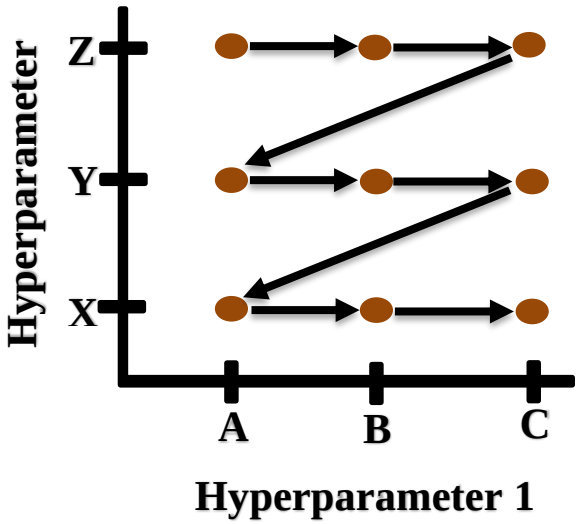


Figure 2.9: The grid search optimization technology [69]

2) The random search optimizer is a technique in which the best random collections of hyperparameters are chosen and utilized to train the model. Although the random search optimizer is similar to some extent to the grid search optimizer as shown in Figure 2.10, it overcomes the drawback of the grid search. The grid search optimizer requires high execution cost, whereas the random search optimizer needs less iteration in order to find the best set of hyperparameters. This means that the random search optimizer requires less amount of the execution time in comparison with the grid search optimizer [69]. This is more especially, if only a few hyperparameters impact the performance of the algorithm being used [68].

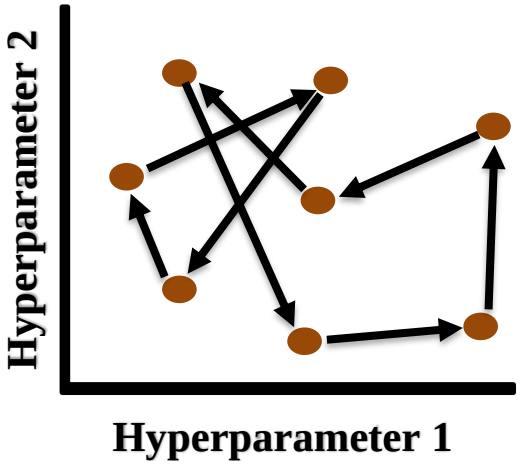


Figure 2.10: The random search optimization technology [69]

This thesis integrates the grid search optimization into three algorithms used to select the optimal hyperparameters namely, SVM, LR, and LightGBM, whereas the random search optimization is used to select the optimal hyperparameters of the LSTM algorithm because using the grid search optimization requires high execution time.

2.8 Evaluation and Performance Metric

2.8.1 The Cross-Validation Technique

In this technique, each example utilizes K once for testing and several times for training in which the value of K is higher than or equal to one. There are several ways to cross-validation, which are:

- Two-fold cross-validation: In this way, the dataset is into two equal parts. The first partition is for training, while the second is for testing. The partitions' roles are then reversed so that the previous training partition becomes the testing partition and vice versa. Each example is used precisely for both testing and training in this manner [56].

- K-fold cross-validation: This method divides the dataset into K subsets of equal-sized. This approach consists of K steps. In each step, all subsets are utilized for training except for one subset, which is used for testing. This technique is utilized, where each partition is tested exactly once [56].

- Leave-one-out cross-validation: This method is a special case of k-fold cross-validation in which the dataset is divided into sections of k these sections are equal in size and in addition to being equal to the number of cases N, the algorithms used, are applied once for each time. The selected instance is used as a one-item test set, while the rest of the cases are used as a training set [56].

2.8.2 Performance Metrics

Performance measures are calculated to compare the prediction accuracy. Evaluation is the stage of determining if the model is performing well, according to every possible evaluation method, as a confusion matrix is a table that can be created for a classifier on a binary dataset and is used to describe the classifier's performance. The confusion matrix includes four measures which are false negatives (FN), false positives (FP), true negatives (TN), and true positives (TP) (see Figure 2.11). In this matrix:

(FN) – The actual is yes and the prediction is no.

(FP) – The actual is no and the prediction is yes.

(TN) – The actual is no and the prediction is no.

(TP) – The actual and the prediction are yes.

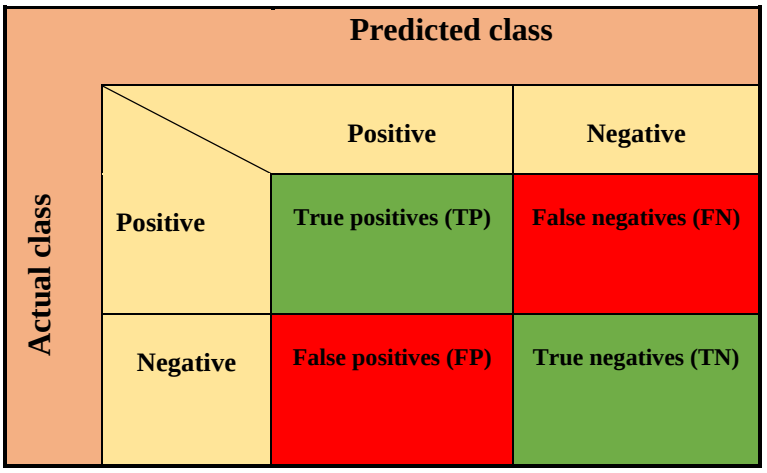


Figure 2.11: The confusion matrix

There are several various metrics as shown below. The computation of these metrics is based on the calculation of the confusion matrix.

- Accuracy is the most commonly used metric for model evaluation in classification. It is calculated by dividing the sum of correct predictions by the total number of predictions based on Equation 2.15 [66].

$$\text{Accuracy} = (TP+TN) / (TP+TN+FP+FN) \qquad (2.15)$$

- Recall is calculated by dividing the value of TP by the sum value of FN and TP based on Equation 2.16 [66].

$$\text{Recall} = TP/(FN + TP) \qquad (2.16)$$

- Precision is calculated by dividing the value of TP by the sum value of FP and TP based on Equation 2.17 [66].

$$\text{Precision} = TP / (FP + TP) \qquad (2.17)$$

- F1-measure is the harmonic method of recall and precision. It is calculated by the 2*((precision*recall)/ (precision + recall)) or based on Equation 2.18 [66].

$$\text{F1-measure} = (2*TP) / (2*TP+FN+FP) \qquad (2.18)$$

CHAPTER THREE

THE PROPOSED SYSTEM

CHAPTER THREE
THE PROPOSED SYSTEM

3.1 Overview

In this Chapter, the major steps of the proposed system in this thesis are discussed. It first presents the proposed system's architecture. Then, this is followed by explaining the preprocessing techniques. The feature extraction is presented alongside the feature selection algorithm. Finally, the used classifiers and their application in the proposed system are demonstrated.

3.2 The Proposed System Architecture

The proposed system architecture contains three stages. Each stage involves sub-steps to reach the study objectives and meet its major aim. These phases are: data preprocessing, feature extraction and selection, and classification and evaluation as depicted in Figure 3.1.

A dataset from the Personality Cafe Forum is utilized to predict individuals' personalities based on their posts. The first phase in the proposed system is data preprocessing, which includes a variety of steps to prepare the proposed system inputs. The second phase is feature extraction and selection which includes the creation of vocabulary from all terms that are extracted from the posts. This also comprises the construction of a vector of features and computes the weight of each feature using TF-IDF. Furthermore, it includes feature selection, which helps reduce the number of features utilized in the classification step. In the final phase, classification techniques are implemented. This comprises the application of Logistic Regression (LR), Support Vector Machine (SVM), LightGBM, and Long Short-Term Memory (LSTM).

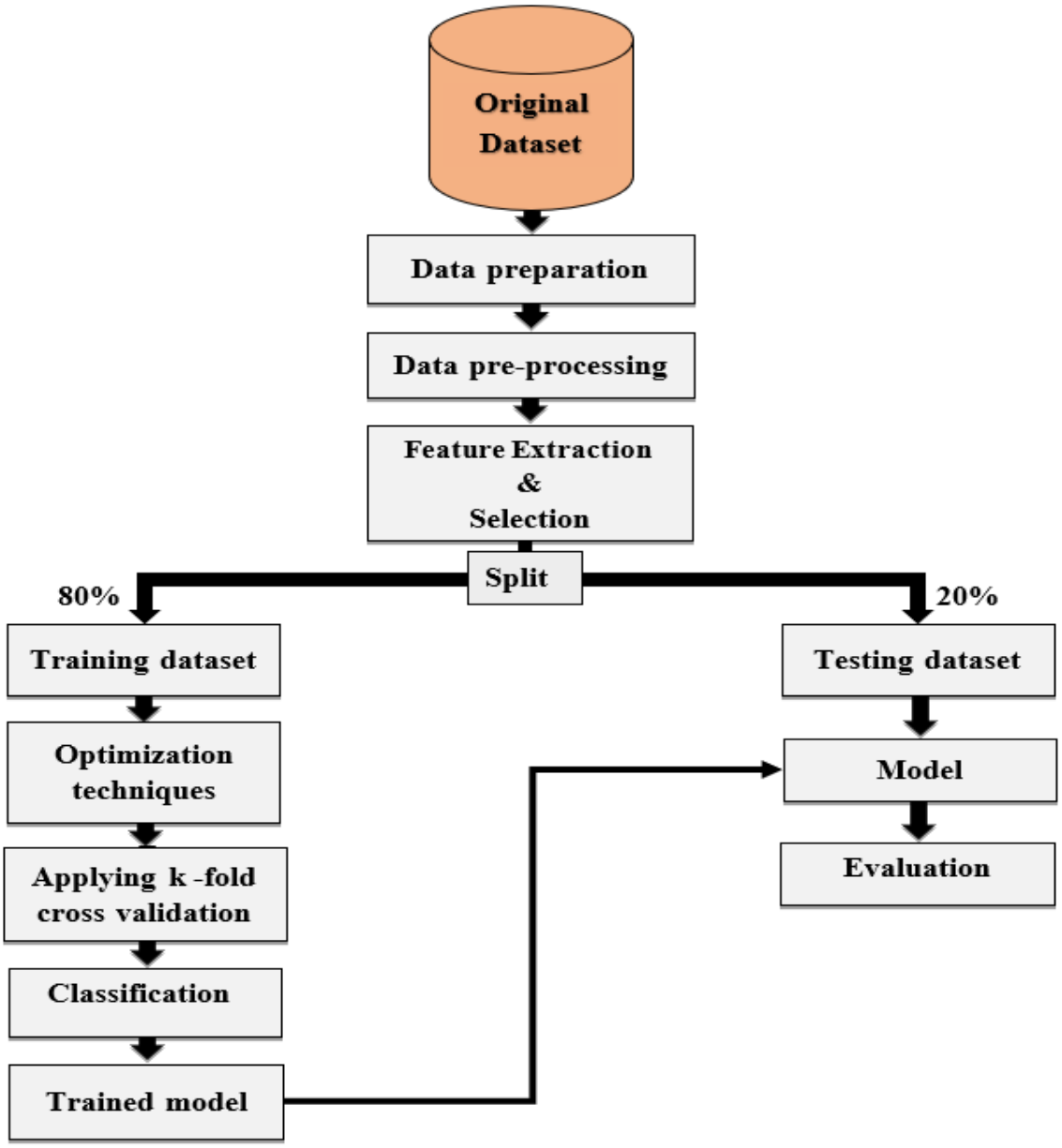


Figure 3.1: The proposed system

3.2.1 The Research Dataset

This study uses the Personality Cafe Forum dataset released in 2017. This dataset can be found on Kaggle at https://www.kaggle.com /datasets/datasnaek/mbti-type. The Personality Cafe Forum is a platform for people to converse and discuss their personalities. In the dataset, there are 8668 rows of data. Each row has a person's posts in which they include a mean of 1220 words [22]. The dataset had only two columns. The first is the MBTI users' type, whereas the second is what people said on the Personality Cafe Forum. The reasons of using this dataset are

(1) it is a public dataset with a large size, and (2) it is not based on microblogging. Figure 3.2 shows a sample of the posts with the MBTI class.

	type	posts
0	INFJ	'http://www.youtube.com/watch?v=qsXHcwe3krw|||...
1	ENTP	'I'm finding the lack of me in these posts ver...
2	INTP	'Good one _____ https://www.youtube.com/wat...
3	INTJ	'Dear INTP, I enjoyed our conversation the o...
4	ENTJ	'You're fired.|||That's another silly misconce...

Figure 3.2: Some posts with the MBTI class

3.2.2 Data Preparing

The second phase of the proposed system is data preparation. Algorithm 3.1 was used to convert the class column of the original dataset into four dimensions, in which each dimension contains either 0 or 1, as shown in Figure 3.3. Algorithm 3.1 describes the process of converting one row, but its strategy was applied for all rows.

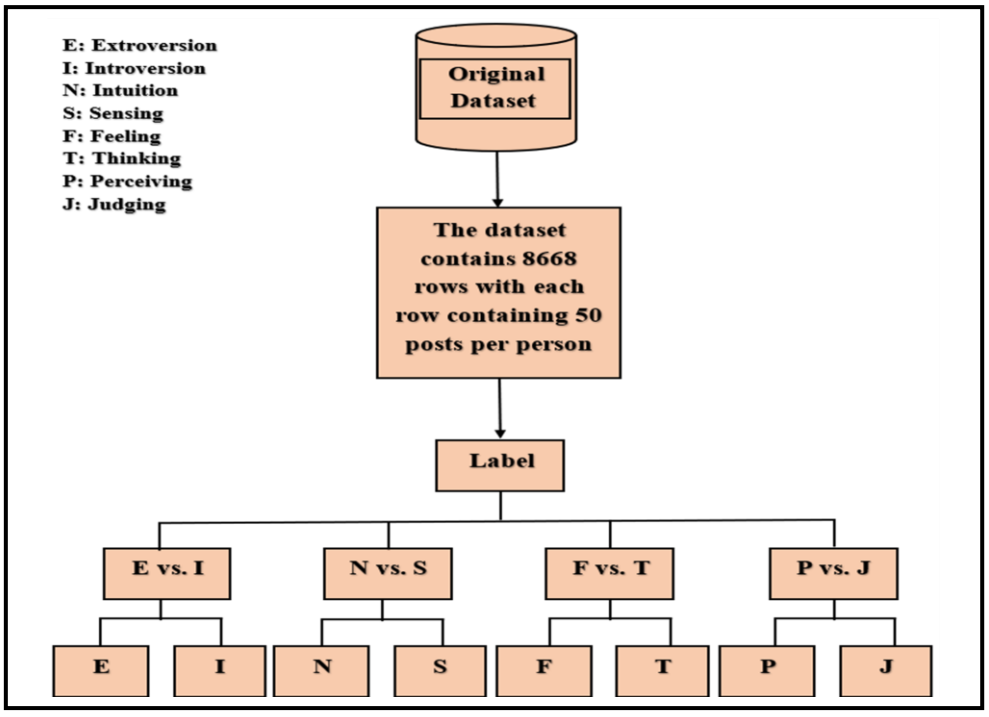


Figure 3.3: Converting the class column

Algorithm 3.1: Converting the Class Column into Four Dimensions

The definition of the algorithm

- Converting the sixteen types of personality into four dimensions, each dimension is assigned either 0 or 1.

Input: The sixteen types of personalities for the MBTI model as found in the dataset.

Output: The four dimensions of the MBTI in which each dimension is either 0 or 1.

Variables' definition

- Type: It is the name of the column in the dataset that contains the sixteen types of personality.
- I, N, F, and J are equal to zero, whereas E, S, T, and P are equal to one.
- row: This is a parameter inside the function.
- i: It is an index to reach all rows in the dataset.

Begin

A function that contains one parameter:(row)

1. **Step 1:** Converting the first dimension (Introversion (I), Extraversion (E)) into either 0 or 1 (binary classification).
2. **Step 1.1:** Type = row['type']
3. **Step 1.2**: if Type[i,0]='I' → 'I' = 0.
4. **Step 1.3:** else if Type[i,0]='E' → 'I' = 1.
5. **Step 1.4:** else:
6. **Step 1.5:** print ('I, E not found')
7. **Step 2:** Converting the second dimension (Intuition (N), Sensing (S)) into either 0 or 1 (binary classification).
8. **Step 2.1:** if Type[i,1]='N' →'N' = 0.
9. **Step 2.2:** Else if Type[i,1]='S' →'N' = 1.
10. **Step 2.3:** else:
11. **Step 2.4:** print ('N, S not found')
12. **Step 3:** Converting the third dimension (Feeling (F), Thinking (T)) into either 0 or 1 (binary classification).
13. **Step 3.1:** if Type[i,2]='F' →'F' = 0.
14. **Step 3.2:** Else if Type[i,2]='T' →'F' = 1.
15. **Step 3.3:** else:
16. **Step 3.4:** print ('T, F not found')
17. **Step 4:** Converting the fourth dimension (Judging (J), Perceiving (P)) into either 0 or 1 (binary classification).
18. **Step 4.1:** if Type[i,3]='J' → 'J' = 0.
19. **Step 4.2:** Else if Type[i,3]='P' → 'J' = 1.
20. **Step 4.3:** else:
21. **Step 4.4:** print ('J, P not found')
22. return ({'IE': I, 'NS': N, 'FT': F, 'JP': J})

End

3.2.3 Data Preprocessing

The third phase of the proposed system is preprocessing. Here, posts are preprocessed because the primary dataset or raw posts are not in the

best format. Thus, it must be preprocessed to clean the content of posts, from noise and uninformative information such as tags, advertisements, stop-words, and URLs. Many various preprocessing procedures are used in this study. Algorithm 3.2 depicts the steps involved in the preprocessing of posts.

Algorithm 3.2: Preprocessing of the Posts Texts

Input: Dataset (DS) // The dataset includes two columns

Output: List of words

Variables' definition

- **POSTS :** A set of posts texts.
- **EW:** Separation of words.
- **W:** Word.

Begin

```
1.  Step 1: Read DS
2.  Step 2: Removing unimportant symbols
3.          i = 1
4.          While i <= length of DS do
5.          Begin
6.              Step 2.1: Remove tags (POSTS[i])
7.              Step 2.2: Remove URLs (POSTS[i])
8.              Step 2.3: Remove punctuations (POSTS[i])
9.              Replace these steps with space
10.             i= i+1
11.         End while
12. Step 3: Removing stop-words
13. Step 4: Splitting the posts texts into words
14.         For POSTS in DS do
15.             Separation of words (EW) based on the space between them
16.         DS = EW
17.         End For
18. Step 5: Stemming        // A procedure that chops off the ends of words in the
                               hope of appropriately changing them into their base form.
19. Step 6: Lemmatization   // Eliminates inflections and maps a word to its root form.
20. Step 7: Removing very short words
21.         For i in POSTS do
22.             If length (w) < 3
23.                 Delete w
24.             End if
25.         End for
```

End

A) Word Tokenization

Tokenization is the process of splitting the text document into phrases or words called tokens. The major utilization of tokenization is identifying meaningful keywords by dividing a large quantity of text into smaller parts. NLP is utilized for constructing applications such as sentimental analysis, language translation, text classification, etc.

B) Posts Cleaning

In this stage, posts were cleaned from uninformative information. This involves removing irrelevant information such as tags, symbols, punctuation, special characters, numbers, URLs links, very short words, and stop-words.

- **Removing Tags**

Step 2.1 in Algorithm 3.2 displays the step of removing tags from posts by regular expressions.

- **Removing URLs Links**

Step 2.2 in Algorithm 3.2 clarifies how to remove links from posts.

- **Removing Punctuations**

Step 2.3 in Algorithm 3.2 depicts the steps of deleting punctuations from posts.

- **Removing Stop-Words**

Step 3 in Algorithm 3.2 clarifies how to remove stop-words from posts.

- **Word Tokenization**

Step 4 in Algorithm 3.2 clarifies the tokenization method for each row in the dataset.

- **Stemming**

Step 5 in Algorithm 3.2 highlights how to implement stemming.

- **Lemmatization**

Step 6 in Algorithm 3.2 illustrates how to use lemmatization.

- **Removing very short words**

Step 7 in Algorithm 3.2 shows how to remove very short words that include less than three letters, where the output of this step becomes the input for the feature extraction and selection stage.

3.2.4 Feature Extraction and Selection

The task of converting a specific text into a vector based on space is an important procedure in text processing. This helps extract the most significant features from the text. Feature extraction techniques are utilized to extract features because textual data contains formats that are not acceptable through machine learning methods. As a result, text features are extracted in a particular format that is acceptable through these techniques. One of the most important and popular methods is TF-IDF. This method can be used to figure out which words are the most important and this, in turn, can help determine keywords for a dataset. When calculating the values of TF and IDF, it will be possible to find the value of TF-IDF. In this study, three different steps of feature extraction were included, as described in the following subsections.

A) Creating Vocabularies

This step consists of several points. The first is to compose a list of unique words from the texts in the training set. These words are called vocabulary. Second, the words for each text are chosen without repetition. Third, the words generated from the original text are arranged alphabetically to account the frequency of each word. After creating the list of features in which each feature consists of a pair (key, token). The key symbolizes the index of features in the list, while the token represents the feature. The number of features in the vocabulary of the dataset that was adopted in this thesis is 1500 features.

B) Creating a Vector of Features

This step makes each text as a set of features by converting the text into a vector space of features, where the number of columns or

dimensions corresponds to all texts in terms of the total number of features. The vector length must be equal to all texts because it depends on the length of the feature. To avoid an unnecessarily large feature, some features occur more frequently than the rest, called tokens. Next, the features are used to be compared with the text and then count the number of times each feature appears. As a result, this determines the importance of each feature in the document by associating each feature with a weight. Hence, the text T was represented as (W1, W2, ..., Wn) where Wi is the weight of the feature i in the text T.

C) Computing Features Weight

In the previous step, the features are represented as a vector. In this step, each feature was calculated using the TF-IDF weight. TF represents term frequency, whereas IDF illustrates inverse document frequency as in Equation 2.3. This can be adopted in text classification to minimize the impact of features that appear frequently in a specific set (text group). Algorithm 3.3 shows the stages followed to extract features.

Algorithm 3.3: Features Extraction Using TF-IDF
Input: A set of posts
Output: Numeral feature vectors
Begin
1. ***Creating vocabulary***
2. **Step 1**: **For** all tokens from the full set of text posts **do** 3. ***Begin*** 4. **Step 1.1**: Assembling a vocabulary of tokens existing in the dataset 5. **Step 1.2**: Making a table of features from a group of vocabulary 6. ***End for***
7. ***Creating a feature vector***
8. **Step 2**: **For** all text posts from the dataset **do** 9. ***Begin*** 10. **Step 2.1**: Converting the text posts into feature vectors 11. **Step 2.2**: Calculating the weight of every feature using Equation (2.1) 12. ***End for***
13. **Step 3**: Return an array with one row for each text and one column for each feature, where each feature related to its weight (TF-IDF)
End

3.2.5 Optimization Techniques

During this step, two optimization techniques were used to determine the appropriate hyperparameters of a classification model. Although the random search optimizer is similar to some extent to the grid search optimizer in terms of functionality, they differ in terms of execution time and prediction accuracy. The grid search outperforms the random search with slight differences in the prediction accuracy. This thesis integrated the grid search optimizer with machine learning algorithms as it requires high execution time and machine learning algorithms require little training time. On the other hand, the random search optimizer was used with the LSTM technique because it needs less iteration to find the best set of hyperparameters and the LSTM algorithm requires a very long training time.

3.2.6 The Classification Methods

The Myers-Briggs Type Indicator (MBTI) is used to classify the personality of 8668 people into 16 personality types based on 50 posts for each user on a social media site. These sixteen types are obtained by four main dimensions. Each pair represents two traits, and only one of them is selected for each user. The total number of the classified posts was divided into 80% for training and 20% for testing .

Four classifiers were implemented to identify the highest accuracy that would be obtained. The applied classifiers are:

1. Logistic Regression (LR).
2. Support Vector Machine (SVM).
3. Light Gradient Boosting Machine (LightGBM).
4. Long Short-Term Memory (LSTM).

After applying the four algorithms, it was shown that LightGBM outperformed others. As a result, the proposed system was based on LightGBM for predicting users' personalities, whereas other classifiers

were used as a baseline.

Light Gradient Boosting Machine

Prediction models are designed to detect a pattern of a specific problem. In other words, prediction models are used to predict a target class for a given instance. If the values of the target class are discrete, classification techniques can be used. This thesis aims to predict users' personalities on social media. Therefore, classification techniques are appropriate for such a problem. Although many machine learning algorithms can be used to predict this class based on a set of features, the main concepts of the data mining algorithms used in this thesis are discussed. Algorithm 3.4 shows the LightGBM algorithm [70].

Algorithm 3.4: LightGBM algorithm [70]

Input:
(a) Training data as input, T={(x1, y1),(x2, y2),....,(xn, yn)} For each iteration I, Xi ∈ X; X ⊆ R; and yi ∈ {-1, +1};
(b) Iteration: I; Model: M;
(c) Loss functions: L(y, f(x)) = L(y, yHAT)2 where L(y, yHAT)2 ∈ quadratic loss;
(d) Iteration: P;
(e) Large gradient data sampling ratio: a; Small gradient data sampling ratio: b;

1. **Many features are mutually exclusive i.e., they never take nonzero values simultaneously, therefore Exclusive Feature Bundling (EFB) techniques can be used to produce a single feature and to increase speed for training of this algorithm.**
2. **To determine which features should he bundled: for each iteration q, $Bundling_q$.add($Features_p$); where bundling is the array.**
   Set of mutually exclusive features arc formed into a single feature namely an exclusive feature boodle.
3. **Define $f_o(x)=argmin_d \sum_{j=1}^{n} L(yi, d)$, where $f_o(x)$ = exclusive feature bundle.**
4. **For each iteration p from 1 to P:**
   For i^{th} examples from 1 to n:
   Prediction = models.prcdict(T);
   Absolute value of gradient,

$$G(x_i, y_i) = \text{Loss}(T, \text{Prediction}) = \left|\frac{\partial L\ (y_i,\ s)}{\partial s}\right|_{s=f_{p-1}\ (x_i)} = \Delta L(y_i, s)$$

Resample dataset using Gradient-based One Side Sampling (GOSS) approach:
topN = a x length(T);
randN = b x length(T);
sorted = GetSortedindices(abs(G));
TS = sorted[1: topN];
RS = RandomPick(sorted[topN: lenght(T)LrandN);
whereTS = Topset and RS = randomset;
New_DatasetT' = TS + RS;

Variance gain Vi(d) is estimated as follows:

$$V_j(d) = \frac{1}{n}\left(\frac{\left(\sum_{x_i \in TS_l} G_i + \left(\frac{1-a}{b}\right)\sum_{x_i \in RS_l} G_i\right)^2}{n_l^j(d)} + \frac{\left(\sum_{x_i \in TS_r} G_i + \left(\frac{1-a}{b}\right)\sum_{x_i \in RS_r} G_i\right)^2}{n_r^j(d)}\right)$$

where $TS_l = \{x_i \in TS: x_{ij} \leq d\}, TS_r = \{x_i \in TS: x_{ij} > d\}, RS_l = \{x_i \in RS: x_{ij} \leq d\}, RS_r = \{x_i \in RS: x_{ij} > d\}$;

Generate a newly formed decision tree F on the dataset T'.

$F = L(x_i, G_i)$;

update $f_p = f_p + F$;

Return F^p

3.2.7 Evaluating the Performance of the Proposed Model

Four different metrics were used to evaluate the efficiency of a particular classification algorithm. This includes accuracy, f1- score, precision, and recall. The calculation of these measures is based on computing confusion matrix which is a matrix that summarizes the number of examples that are either properly or wrongly predicted by a classification model. The results of this thesis are compared with previous studies according to the accuracy measure only.

In section 2.8.2, a two-dimensional confusion matrix was discussed. In this thesis, the predicted classes are four, so the form of the confusion matrix would be as follows (see Figure 3.4).

Actual \ Predicted		Class One	Class Two	Class Three	Class Four
Actual	Class One	Predicted Class One as Class One	Predicted Class One as Class Two	Predicted Class One as Class Three	Predicted Class One as Class Four
	Class Two	Predicted Class Two as Class One	Predicted Class Two as Class Two	Predicted Class Two as Class Three	Predicted Class Two as Class Four
	Class Three	Predicted Class Three as Class One	Predicted Class Three as Class Two	Predicted Class Three as Class Three	Predicted Class Three as Class Four
	Class Four	Predicted Class Four as Class One	Predicted Class Four as Class Two	Predicted Class Four as Class Three	Predicted Class Four as Class Four

Figure 3.4: The Confusion matrix of a four class categorization

CHAPTER FOUR

RESULTS AND DISCUSSION

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Overview

The proposed system explained in Chapter three is applied to achieve the thesis aims explained in Chapter one. The findings of all phases are ordered according to their appearance in Chapter three. However, this Chapter begins with the software and hardware requirements for implementing the proposed methodology.

4.2 Software and Hardware Specifications

The proposed methodology is implemented based on the following software and hardware specifications.

Hardware: Processor Intel i5- 10500H, RAM 16GB, Storage1 SSD 512GB, Storage2 SSD 256GB, Freq. 2.50**.**

GPU: NVIDIA GEFORCE GTX 1650 4GB.

Software: Operating System: Windows10 (64)bit.

Programming language: Python language

IDE: the system is implemented by Python 3.9.12, Jupyter Notebook.

4.3 Results of Data Preprocessing

The outcomes of the preprocessing stage are displayed after conducting word tokenization to divide words, and cleaning posts by deleting irrelevant information. The results of the preprocessing step are a set of tokens for each text.

- **Removing Tags**

This indicates words starting with @username. This is also deleted from the post's text. Table 4.1 presents an example of tags removal.

Table 4.1: Removing Tages

Before Removing Tags	"'18/37 @.@|||Science is not perfect. No scientist claims that it is, or that scientific information will not be revised as we discover new things. Rational thinking has been very useful to our society....|||INFP- Edgar Allen Poe was an INFP and he's in your siggy.|||People see the obvious Fi and are quick to put her as INFP. I agree that she has no Ne. I see her as an ISFP. Compare her to Haku (definite INFP). |||what you need to know as an INTJ fiction writer: http://phantomshine.blogspot.com/2012/05/writer-analysis-through-mbti.html.
After Removing Tags	"'18/37 .|||Science is not perfect. No scientist claims that it is, or that scientific information will not be revised as we discover new things. Rational thinking has been very useful to our society....|||INFP- Edgar Allen Poe was an INFP and he's in your siggy.|||People see the obvious Fi and are quick to put her as INFP. I agree that she has no Ne. I see her as an ISFP. Compare her to Haku (definite INFP). |||what you need to know as an INTJ fiction writer: http://phantomshine.blogspot.com/2012/05/writer-analysis-through-mbti.html.

- **Removing URLs Links**

URLs were regarded as unnecessary data in a post's text. As a result, such URLs are deleted and replaced with space. Table 4.2 clarifies the outcomes of removing URLs.

Table 4.2: Removing URLs Links

Before RemovingURLs	"'18/37 .|||Science is not perfect. No scientist claims that it is, or that scientific information will not be revised as we discover new things. Rational thinking has been very useful to our society....|||INFP- Edgar Allen Poe was an INFP and he's in your siggy.|||People see the obvious Fi and are quick to put her as INFP. I agree that she has no Ne. I see her as an ISFP. Compare her to Haku (definite INFP). |||what you need to know as an INTJ fiction writer: http://phantomshine.blogspot.com/2012/05/writer-analysis-through-mbti.html.
After Removing URLs	"'18/37 .|||Science is not perfect. No scientist claims that it is, or that scientific information will not be revised as we discover new things. Rational thinking has been very useful to our society....|||INFP- Edgar Allen Poe was an INFP and he's in your siggy.|||People see the obvious Fi and are quick to put her as INFP. I agree that she has no Ne. I see her as an ISFP. Compare her to Haku (definite INFP). |||what you need to know as an INTJ fiction writer:

- **Removing Punctuations**

Punctuations are deleted from all posts' texts. Table 4.3 shows a sample of the posts before and after deleting punctuations.

Table 4.3: Removing Punctuations

Before Removing Punctuation	"'18/37 .|||Science is not perfect. No scientist claims that it is, or that scientific information will not be revised as we discover new things. Rational thinking has been very useful to our society....|||INFP- Edgar Allen Poe was an INFP and he's in your siggy.|||People see the obvious Fi and are quick to put her as INFP. I agree that she has no Ne. I see her as an ISFP. Compare her to Haku (definite INFP). |||what you need to know as an INTJ fiction writer:
After Removing Punctuation	Science is not perfect No scientist claims that it is or that scientific information will not be revised as we discover new things Rational thinking has been very useful to our society INFP Edgar Allen Poe was an INFP and he s in your siggy People see the obvious Fi and are quick to put her as INFP I agree that she has no Ne I see her as an ISFP Compare her to Haku definite INFP what you need to know as an INTJ fiction writer

- **Removing Stop-words**

In text analysis, uninformative words which are called stop-words are deleted from the text of the original posts. Table 4.4 shows a sample of posts after deleting stop-words.

Table 4.4: Removing Stop-words

Before Removing Stop-words	Science is not perfect No scientist claims that it is or that scientific information will not be revised as we discover new things Rational thinking has been very useful to our society INFP Edgar Allen Poe was an INFP and he s in your siggy People see the obvious Fi and are quick to put her as INFP I agree that she has no Ne I see her as an ISFP Compare her to Haku definite INFP what you need to know as an INTJ fiction writer
After Removing Stop-words	science perfect scientist claims scientific information revised discover new things rational thinking useful society infp edgar allen poe infp siggy people see obvious fi quick put infp agree ne see isfp compare haku definite infp need know intj fiction writer

- **Word Tokenization**

Table 4.5 depicts the outcomes of splitting large posts into its constituent words.

Table 4.5: Word Tokenization

Before Tokenization	science perfect scientist claims scientific information revised discover new things rational thinking useful society infp edgar allen poe infp siggy people see obvious fi quick put infp agree ne see isfp compare haku definite infp need know intj fiction writer
After Tokenization	['science', 'perfect', 'scientist', 'claims', 'scientific', 'information', 'revised', 'discover', 'new', 'things', 'rational', 'thinking', 'useful', 'society', 'infp', 'edgar', 'allen', 'poe', 'infp', 'siggy', 'people', 'see', 'obvious', 'fi', 'quick', 'put', 'infp', 'agree', 'ne', 'see', 'isfp', 'compare', 'haku', 'definite', 'infp', 'need', 'know', 'intj', 'fiction', 'writer']

- **Stemming**

This means cutting off additions such as suffixes and prefixes from words. This helps may reduce the number of words before the feature extraction stage. Table 4.6 illustrates a sample of posts after using stemming.

Table 4.6: The use of stemming

Before Stemming	['science', 'perfect', 'scientist', 'claims', 'scientific', 'information', 'revised', 'discover', 'new', 'things', 'rational', 'thinking', 'useful', 'society', 'infp', 'edgar', 'allen', 'poe', 'infp', 'siggy', 'people', 'see', 'obvious', 'fi', 'quick', 'put', 'infp', 'agree', 'ne', 'see', 'isfp', 'compare', 'haku', 'definite', 'infp', 'need', 'know', 'intj', 'fiction', 'writer']
After Stemming	['science', 'perfect', 'scientist', 'claim', 'scientific', 'inform', 'revise', 'discover', 'new', 'thing', 'ration', 'think', 'use', 'society', 'infp', 'edgar', 'allen', 'poe', 'infp', 'siggi', 'people', 'see', 'obvious', 'fi', 'quick', 'put', 'infp', 'agree', 'ne', 'see', 'isfp', 'compare', 'haku', 'definite', 'infp', 'need', 'know', 'intj', 'fiction', 'writer']

- **Lemmatization**

This means returning the words to their origin by comparing the words with their own dictionary. Table 4.7 shows a sample of a post after using the lemmatization.

Table 4.7: The use of lemmatization

Before lemmatization	['science', 'perfect', 'scientist', 'claim', 'scientific', 'inform', 'revise', 'discover', 'new', 'thing', 'ration', 'think', 'use', 'society', 'infp', 'edgar', 'allen', 'poe', 'infp', 'siggi', 'people', 'see', 'obvious', 'fi', 'quick', 'put', 'infp', 'agree', 'ne', 'see', 'isfp', 'compare', 'haku', 'definite', 'infp', 'need', 'know', 'intj', 'fiction', 'writer', 'Children']
After lemmatization	['science', 'perfect', 'scientist', 'claim', 'scientific', 'inform', 'revise', 'discover', 'new', 'thing', 'ration', 'think', 'use', 'society', 'infp', 'edgar', 'allen', 'poe', 'infp', 'siggi', 'people', 'see', 'obvious', 'fi', 'quick', 'put', 'infp', 'agree', 'ne', 'see', 'isfp', 'compare', 'haku', 'definite', 'infp', 'need', 'know', 'intj', 'fiction', 'writer', 'Child']

- **Removing Very Short Words**

Very short words are deleted from all posts' texts. Table 4.8 presents a sample of a post before and after deleting very short words.

Table 4.8: Removing very short words

Before Removing Very Short Words	['science', 'perfect', 'scientist', 'claim', 'scientific', 'inform', 'revise', 'discover', 'new', 'thing', 'ration', 'think', 'use', 'society', 'infp', 'edgar', 'allen', 'poe', 'infp', 'siggi', 'people', 'see', 'obvious', 'fi', 'quick', 'put', 'infp', 'agree', 'ne', 'see', 'isfp', 'compare', 'haku', 'definite', 'infp', 'need', 'know', 'intj', 'fiction', 'writer']
After Removing Very Short Words	science perfect scientist claim scientific inform revise discover thing ration think society infp edgar allen infp siggi people obvious quick infp agree isfp compare haku definite infp need know intj fiction writer

After completing the preprocessing stages, the posts' texts became perfectly free of uninformative information that could affect the proposed

system's accuracy. Figure 4.1 illustrates obtaining the text of the posts devoid of useless information.

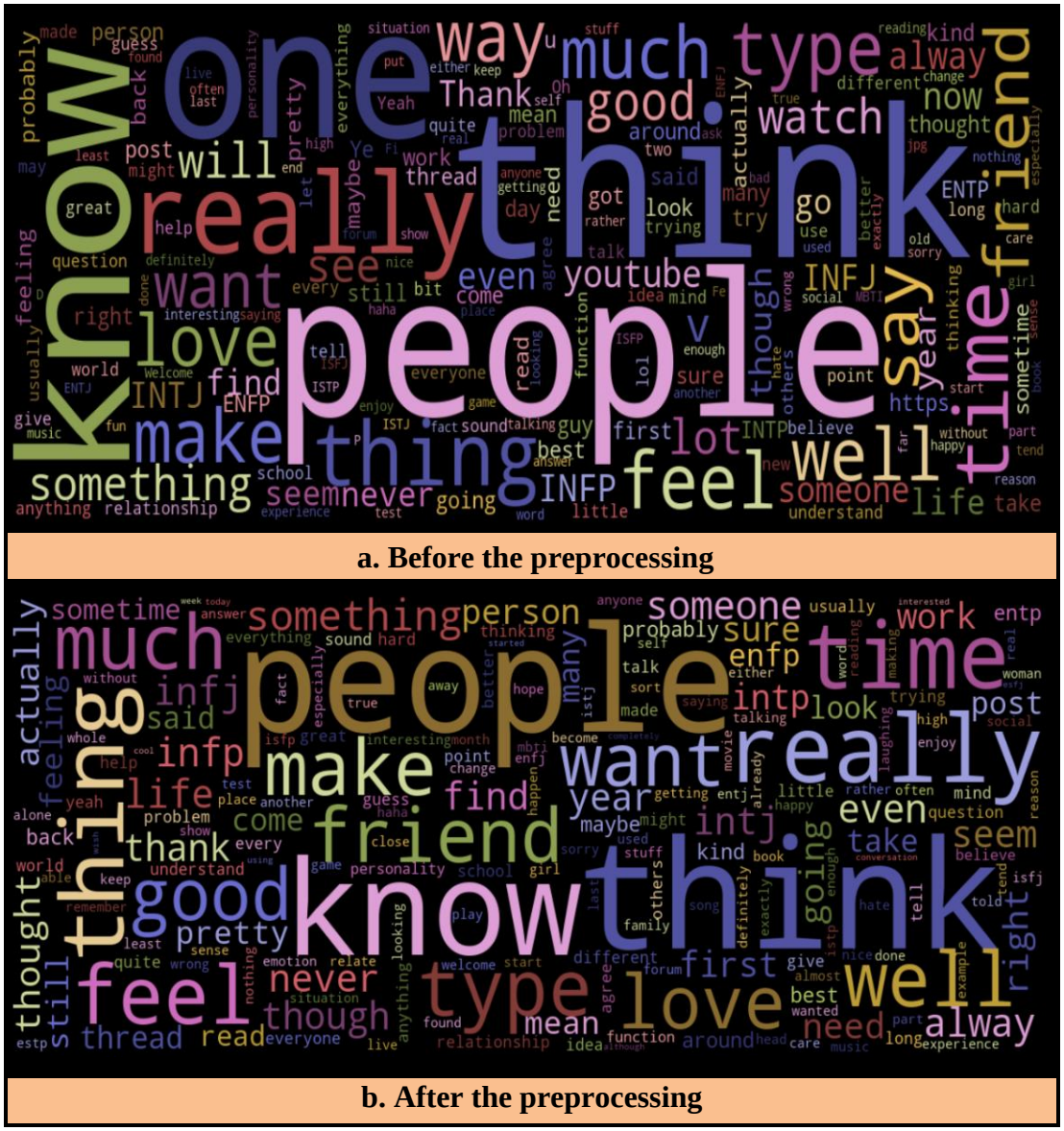


Figure 4.1: The word cloud visualization analysis of the posts texts.

(a) Before the preprocessing (b) After the preprocessing

4.4 Results of Features Extraction and Selection

Features are obtained based on the notion of BoW. Each text was converted into a vector of features and then the weight of these features was calculated utilizing TF-IDF. The outcome of this step was a vector of weights TF-IDF of the features. Table 4.9 demonstrates an example of the outcomes of BoW.

Table 4.9 : An example of the results of BoW

Features / No. Posts	Home	Weed	Video	...	Around	Followed	Convinced
Post 0	1	0	0	...	1	0	1
Post 1	1	1	1	...	1	1	1
Post 2	1	0	1	...	1	0	0
...	...	...	...	...	...	...	...
Post 8667	0	0	1	...	1	1	1
Post 8668	1	1	1	...	1	1	1

As shown in Table 4.9, it is clear that some tokens contain a zero value because the subset of tokens would be used from the collection of posts where these tokens may exist or not exist in the posts' text. If this feature exists, a value will be given to represent the number of times a token appears in the post, whereas if it is not present, the resulting matrix will contain many feature values that are zeros.

Table 4.10 shows a sample of the frequency of each word in posts.

Table 4.10: The frequency of each word in posts

Features / No. Posts	Home	Weed	Video	...	Around	Followed	Convinced
Post 0	2	0	0	...	1	0	1
Post 1	2	1	4	...	2	2	1
Post 2	1	0	1	...	2	0	0
...	...	...	...	...	...	...	...
Post 8667	0	0	1	...	1	1	1
Post 8668	4	1	4	...	1	1	1

Figure 4.2 shows each feature in the posts' text that has a weight which was calculated by using TF-IDF.

```
X: Part of the first post in TF-IDF representation
[0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.08105478 0.07066064
 0.         0.         0.         0.         0.         0.
 0.         0.04516864 0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.05321691 0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.0871647  0.         0.         0.
 0.         0.         0.         0.05506308 0.0708757  0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.
 0.         0.16585935 0.         0.         0.09676192 0.
 0.         0.04970682 0.         0.         0.         0.
 0.07397056 0.         0.         0.         0.         0.
 0.         0.0748045  0.07639898 0.09185775 0.         0.
```

Figure 4.2: Outcomes of Features Using TF-IDF

4.5 Results of the Classification Methods

The total number of classified posts is 8668 which are split into 6934.4 as a training set (80%) and 1733.6 as a testing set (20%). The accuracy of the prediction performance of the four classifiers is evaluated. The accuracy measure is more sensitive to the distribution of the target variable, as well as the performance of the classifier on an unbalanced dataset.

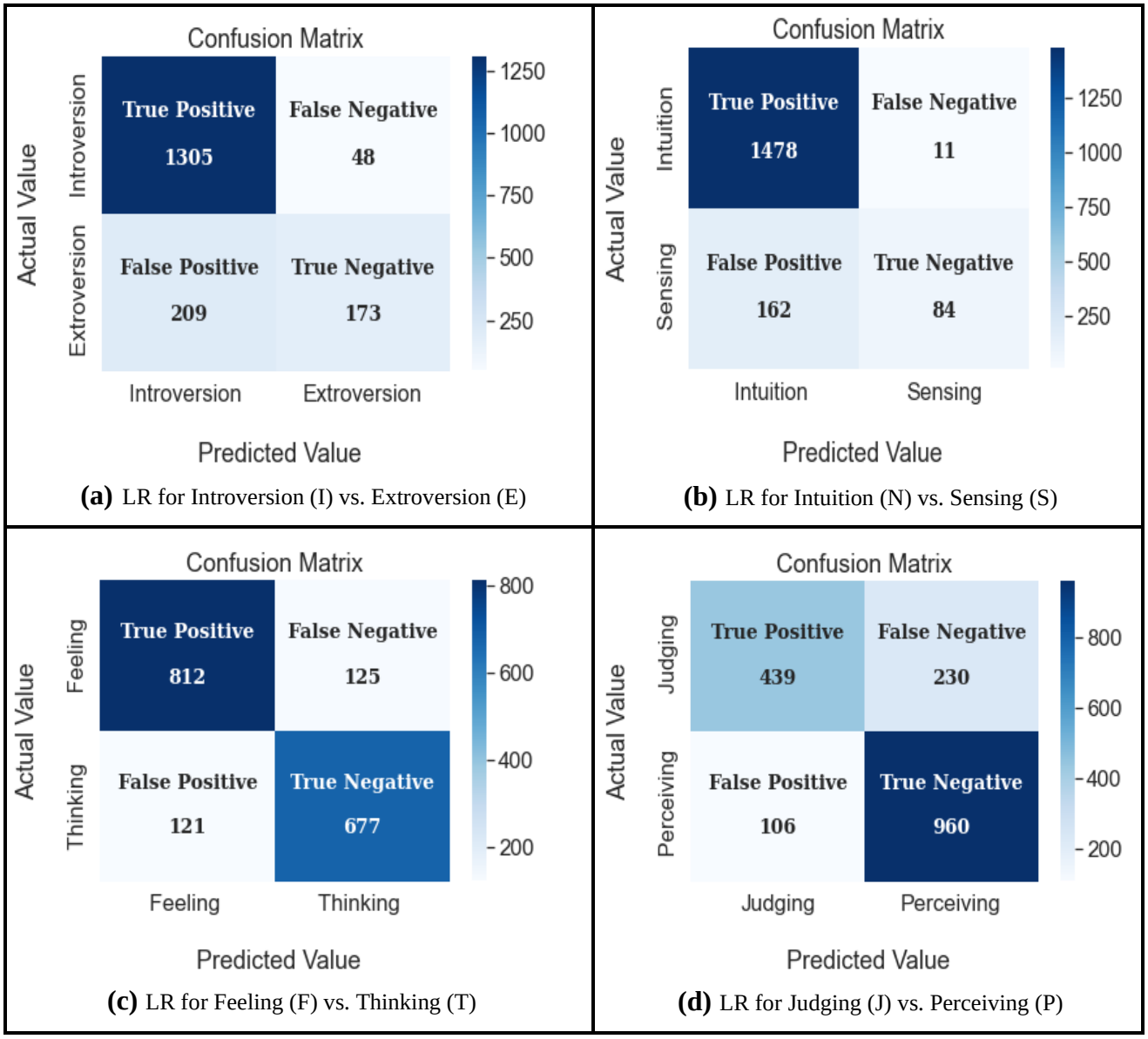
4.5.1 Machine Learning

After the preprocessing steps, feature extraction, classification, and performance evaluation stages for all three classifiers are performed. The accuracy of the obtained machine learning algorithms for each dimension is summarized in Tables 4.11, 4.12, 4.13, 4.14, 4.15, and 4.16. Figures 4.3, 4.4, 4.5, 4.6, 4.7, and 4.8 shows the confusion matrix of the MBTI four dimensions. Figure 4.9 shows the prediction accuracy of the MBTI four dimensions based on the three classifiers namely, LR, SVM, and LightGBM.

Table 4.11 shows that using LR with stop-words, stemming, and lemmatization can achieve the highest accuracy. Many reasons can be drawn behind such results. First, stemming decreases the number of words in the corpus and correlates with the words that have similar meanings. Second, lemmatization helps the classifiers better predict the labels. Third, stop-words reduced the size of the dataset. Based on Figure 4.3, the prediction of Intuition (N) vs. Sensing (S) is better than other dimensions. The accuracy of Intuition (N) vs. Sensing (S) is 90.03%. This result is in agreement with another research study [36].

Table 4.11: The accuracy after different preprocessing techniques for the Logistic Regression algorithm

Kaggle dataset from Personality Café Forum in 2017	Logistic Regression					The average accuracy of the four dimensions
	Metrics	IE	NS	FT	JP	
WO stop-words, **W** stemming, **W** lemmatizing.	Accuracy	85.01	89.28	85.88	80.29	85.11
	Precision	82.0	88.0	86.0	80 .0	85.33
	Recall	70.0	64.0	86.0	77.0	74.25
	F1-score	74.0	69.0	86.0	78.0	76.75
W stop-words, **W** stemming, **W** lemmatizing.	Accuracy	85.19	90.03	86.00	81.00	85.55
	Precision	82.0	89.0	86.0	81.0	84.5
	Recall	71.0	67.0	86.0	78.0	75.5
	F1-score	74.0	72.0	86.0	79.0	77.75
W stop-words, **WO** stemming, **WO** lemmatizing.	Accuracy	85.59	89.39	84.55	79.48	84.75
	Precision	83.0	88.0	84.0	79.0	83.5
	Recall	72.0	64.0	84.0	77.0	74.25
	F1-score	75.0	69.0	84.0	77.0	76.25
W stop-words, **WO** stemming, **W** lemmatizing.	Accuracy	85.24	89.22	85.24	80.06	84.94
	Precision	82.0	88.0	85.0	80.0	83.75
	Recall	71.0	64.0	85.0	77.0	74.25
	F1-score	74.0	68.0	85.0	78.0	76.25
W stop-words, **W** stemming, **WO** lemmatizing.	Accuracy	85.13	89.97	85.53	80.52	85.28
	Precision	82.0	89.0	85.0	80.0	84.0
	Recall	71.0	67.0	85.0	78.0	75.25
	F1-score	74.0	72.0	85.0	79.0	77.5
WO stop-words, **WO** stemming, **WO** lemmatizing**.**	Accuracy	83.75	87.32	83.69	79.08	83.46
	Precision	83.0	88.0	84.0	80.0	83.75
	Recall	66.0	56.0	84.0	75.0	70.25
	F1-score	69.0	57.0	84.0	76.0	71.5
WO stop-words, **W** stemming, **WO** lemmatizing**.**	Accuracy	85.07	89.34	85.71	80.23	85.08
	Precision	83.0	88.0	86.0	80.0	84.25
	Recall	70.0	64.0	86.0	77.0	74.25
	F1-score	74.0	69.0	86.0	78.0	76.75
WO stop-words, **WO** stemming, **W** lemmatizing**.**	Accuracy	84.78	88.88	84.61	79.88	84.53
	Precision	82.0	89.0	85.0	80.0	84.0
	Recall	69.0	62.0	84.0	77.0	73.0
	F1-score	73.0	66.0	85.0	78.0	75.5
Note: W=With, WO=Without, S= Seconds						



(a) LR for Introversion (I) vs. Extroversion (E)

(b) LR for Intuition (N) vs. Sensing (S)

(c) LR for Feeling (F) vs. Thinking (T)

(d) LR for Judging (J) vs. Perceiving (P)

Figure 4.3: The confusion matrix with: stop-words, stemming, lemmatization, and without grid search optimizer for the Logistic Regression algorithm

Table 4.12 demonstrates that the use of SVM with stop-words, stemming, and lemmatization has achieved the highest accuracy. According to Figure 4.4, the prediction of Intuition (N) vs. Sensing (S) is superior to other dimensions where its accuracy is 90.72%. This finding is in agreement with many studies [22, 36].

Table 4.12: The accuracy after different preprocessing techniques for the Support Vector Machine algorithm

Kaggle dataset from Personality Café Forum in 2017	Support Vector Classifier					The average accuracy of the four dimensions
	Metrics	IE	NS	FT	JP	
WO stop-words, **W** stemming, **W** lemmatizing.	Accuracy	86.46	90.49	85.48	80.58	85.75
	Precision	84.0	89.0	85.0	80.0	84.5
	Recall	74.0	69.0	85.0	78.0	76.5
	F1-score	77.0	74.0	85.0	79.0	78.75
W stop-words, **W** stemming, **W** lemmatizing.	Accuracy	86.05	90.72	85.53	80.98	85.82
	Precision	83.0	89.0	85.0	81.0	84.5
	Recall	73.0	70.0	85.0	79.0	76.75
	F1-score	76.0	75.0	85.0	79.0	78.75
W stop-words, **WO** stemming, **WO** lemmatizing.	Accuracy	86.11	89.91	83.98	79.88	84.97
	Precision	83.0	89.0	84.0	80.0	84.0
	Recall	73.0	66.0	84.0	77.0	75.0
	F1-score	76.0	72.0	84.0	78.0	77.5
W stop-words, **WO** stemming, **W** lemmatizing.	Accuracy	85.88	90.20	84.50	80.00	85.14
	Precision	83.0	89.0	84.0	0.80	64.2
	Recall	72.0	68.0	84.0	0.77	56.19
	F1-score	76.0	73.0	84.0	0.78	58.44
W stop-words, **W** stemming, **WO** lemmatizing.	Accuracy	85.99	90.72	85.48	81.04	85.80
	Precision	83.0	89.0	85.0	81.0	84.5
	Recall	73.0	70.0	85.0	79.0	76.75
	F1-score	76.0	75.0	85.0	79.0	78.75
WO stop-words, **WO** stemming, **WO** lemmatizing**.**	Accuracy	85.59	88.65	83.80	80.00	84.51
	Precision	83.0	88.0	84.0	0.80	63.95
	Recall	72.0	61.0	84.0	0.77	54.44
	F1-score	75.0	65.0	84.0	0.78	56.19
WO stop-words, **W** stemming, **WO** lemmatizing**.**	Accuracy	86.40	90.49	85.19	80.46	85.63
	Precision	84.0	89.0	85.0	80.0	84.5
	Recall	74.0	69.0	85.0	78.0	76.5
	F1-score	77.0	74.0	85.0	79.0	78.75
WO stop-words, **WO** stemming, **W** lemmatizing**.**	Accuracy	85.53	89.68	84.15	79.83	84.79
	Precision	83.0	89.0	84.0	80.0	84.0
	Recall	71.0	65.0	84.0	77.0	74.25
	F1-score	75.0	70.0	84.0	78.0	76.75

(a) SVM for Introversion (I) vs. Extroversion (E)

(b) SVM for Intuition (N) vs. Sensing (S)

(c) SVM for Feeling (F) vs. Thinking (T)

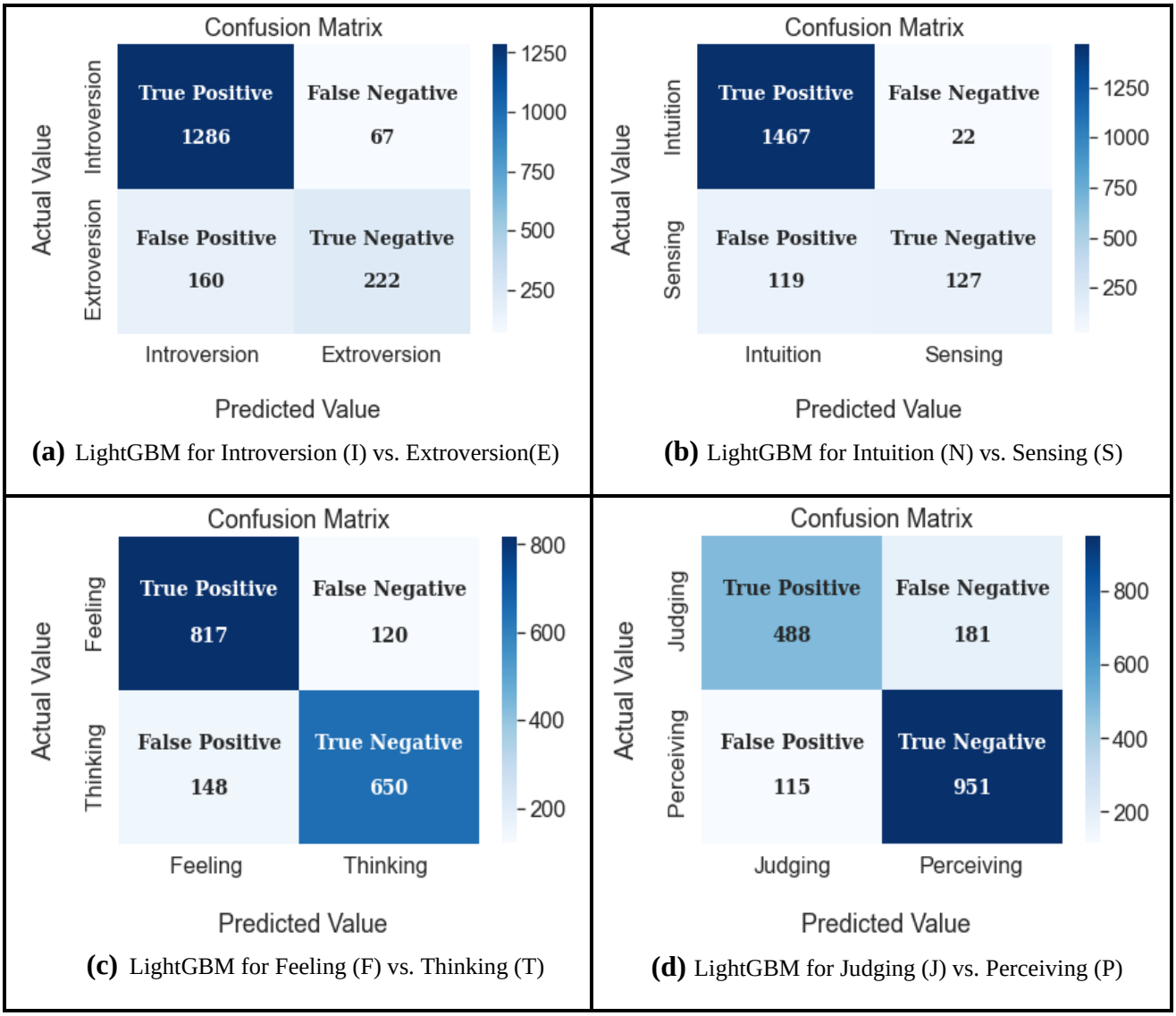
(d) SVM for Judging (J) vs. Perceiving (P)

Figure 4.4: The confusion matrix with: stop-words, stemming, lemmatization, and without grid search optimizer for the Support Vector classifier algorithm

Table 4.13 presents the use of LightGBM with stop-words, stemming, and lemmatization that have obtained the best accuracy. As shown in Figure 4.5, the prediction of Intuition (N) vs. Sensing (S) is better than other dimensions where it is 91.87%. This outcome is consistent with a previous study [22].

Table 4.13: The accuracy after different preprocessing techniques for the LightGBM algorithm

Kaggle dataset from Personality Café Forum in 2017	LightGBM					The average accuracy of the four dimensions
	Metrics	IE	NS	FT	JP	
WO stop-words, **W** stemming, **W** lemmatizing.	Accuracy	86.97	91.82	84.09	82.59	86.36
	Precision	83.0	89.0	84.0	82.0	84.5
	Recall	77.0	75.0	84.0	81.0	79.25
	F1-score	79.0	80.0	84.0	81.0	81.0
W stop-words, **W** stemming, **W** lemmatizing.	Accuracy	86.92	91.87	84.55	82.94	86.57
	Precision	83.0	89.0	85.0	82.0	84.75
	Recall	77.0	75.0	84.0	81.0	79.25
	F1-score	79.0	80.0	84.0	82.0	81.25
W stop-words, **WO** stemming, **WO** lemmatizing.	Accuracy	86.05	91.18	83.98	80.81	85.50
	Precision	81.0	87.0	84.0	80.0	83.0
	Recall	75.0	73.0	84.0	79.0	77.75
	F1-score	78.0	78.0	84.0	79.0	79.75
W stop-words, **WO** stemming, **W** lemmatizing.	Accuracy	86.40	91.01	84.67	81.79	85.96
	Precision	82.0	86.0	85.0	81.0	83.5
	Recall	76.0	74.0	85.0	80.0	78.75
	F1-score	78.0	78.0	85.0	80.0	80.25
W stop-words, **W** stemming, **WO** lemmatizing.	Accuracy	86.74	91.70	84.50	82.77	86.42
	Precision	83.0	89.0	84.0	82.0	84.5
	Recall	76.0	74.0	84.0	81.0	78.75
	F1-score	79.0	79.0	84.0	81.0	80.75
WO stop-words, **WO** stemming, **WO** lemmatizing.	Accuracy	85.59	91.07	83.63	81.44	85.43
	Precision	81.0	88.0	84.0	81.0	83.5
	Recall	74.0	72.0	84.0	79.0	77.25
	F1-score	77.0	77.0	84.0	80.0	79.5
WO stop-words, **W** stemming, **WO** lemmatizing.	Accuracy	85.07	89.34	85.71	80.23	85.08
	Precision	83.0	88.0	86.0	80.0	84.25
	Recall	70.0	64.0	86.0	77.0	74.25
	F1-score	74.0	69.0	86.0	78.0	76.75
WO stop-words, **WO** stemming, **W** lemmatizing.	Accuracy	85.24	90.55	83.69	80.81	85.07
	Precision	80.0	87.0	84.0	80.0	82.75
	Recall	74.0	70.0	84.0	79.0	76.75
	F1-score	76.0	75.0	84.0	79.0	78.5



(a) LightGBM for Introversion (I) vs. Extroversion(E)

(b) LightGBM for Intuition (N) vs. Sensing (S)

(c) LightGBM for Feeling (F) vs. Thinking (T)

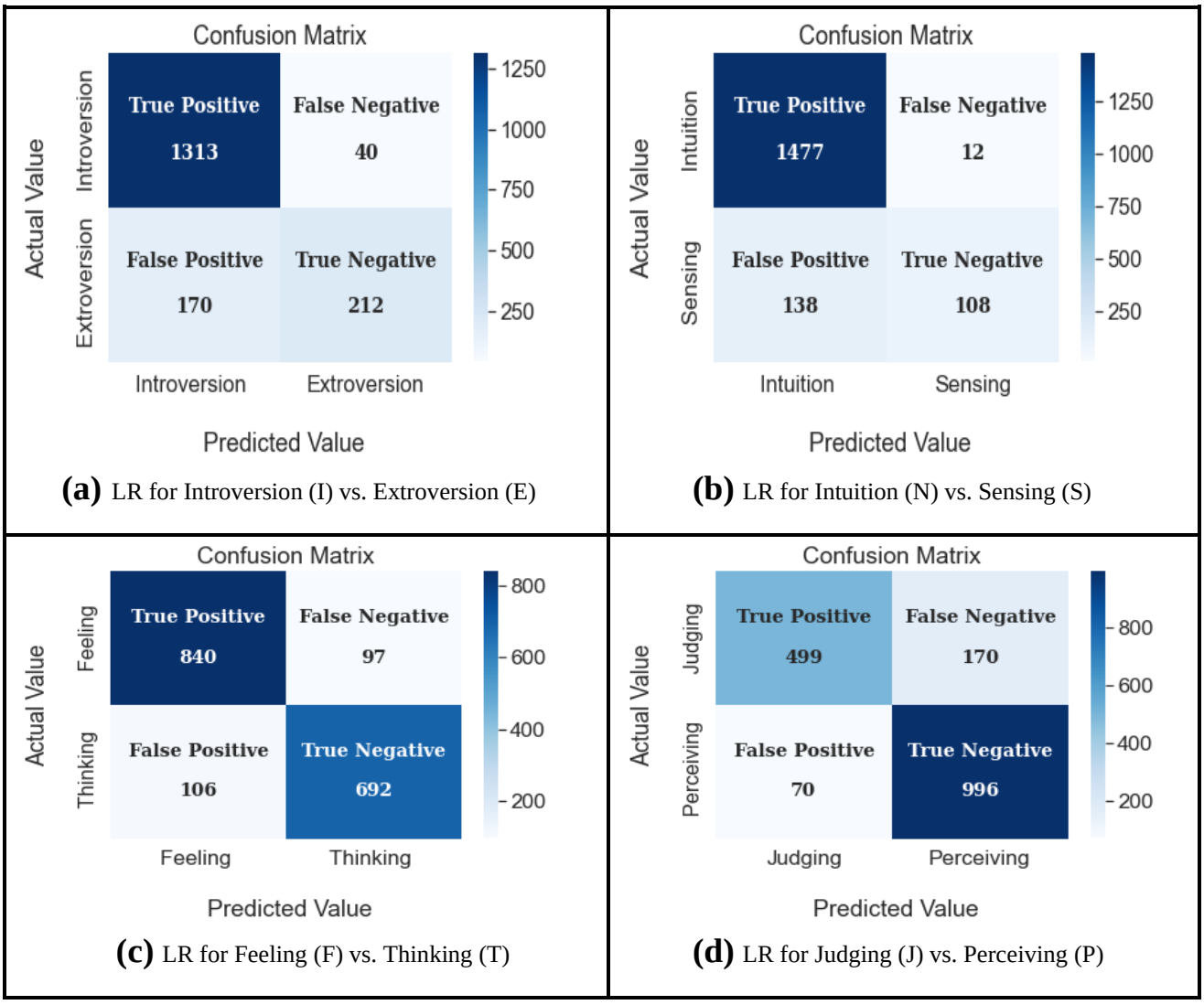
(d) LightGBM for Judging (J) vs. Perceiving (P)

Figure 4.5: The confusion matrix with: stop-words, stemming, lemmatization, and without grid search optimizer for the LightGBM algorithm

Table 4.14 shows that using LR with stop-words, stemming, grid search optimizer, and lemmatization have obtained the highest accuracy. This because the grid search optimization has identified the best hyperparameters. Based on Figure 4.6, the prediction of Intuition (N) vs. Sensing (S) is better than other dimensions in which it is 91.35%.

Table 4.14: The accuracy after different preprocessing techniques and with the integration of the grid search optimizer for the Logistic Regression algorithm

Kaggle dataset from Personality Café Forum in 2017	Logistic Regression					The average accuracy of the four dimensions
	Metrics	IE	NS	FT	JP	
W stop-words, W stemming, W lemmatizing.	Accuracy	85.19	90.03	86.00	81.00	85.55
	Precision	82.0	89.0	86.0	81.0	84.5
	Recall	71.0	67.0	86.0	78.0	75.5
	F1-score	74.0	72.0	86.0	79.0	77.75
W stop-words, W stemming, W lemmatizing, W grid search optimizer	Accuracy	87.89	91.35	88.30	86.16	88.42
	Precision	86.33	90.72	88.25	86.55	87.96
	Recall	76.27	71.54	88.18	84.01	80.0
	F1-score	79.73	77.09	88.21	84.93	82.49
Best parameters	C = 1.0, penalty = l2					



(a) LR for Introversion (I) vs. Extroversion (E)

(b) LR for Intuition (N) vs. Sensing (S)

(c) LR for Feeling (F) vs. Thinking (T)

(d) LR for Judging (J) vs. Perceiving (P)

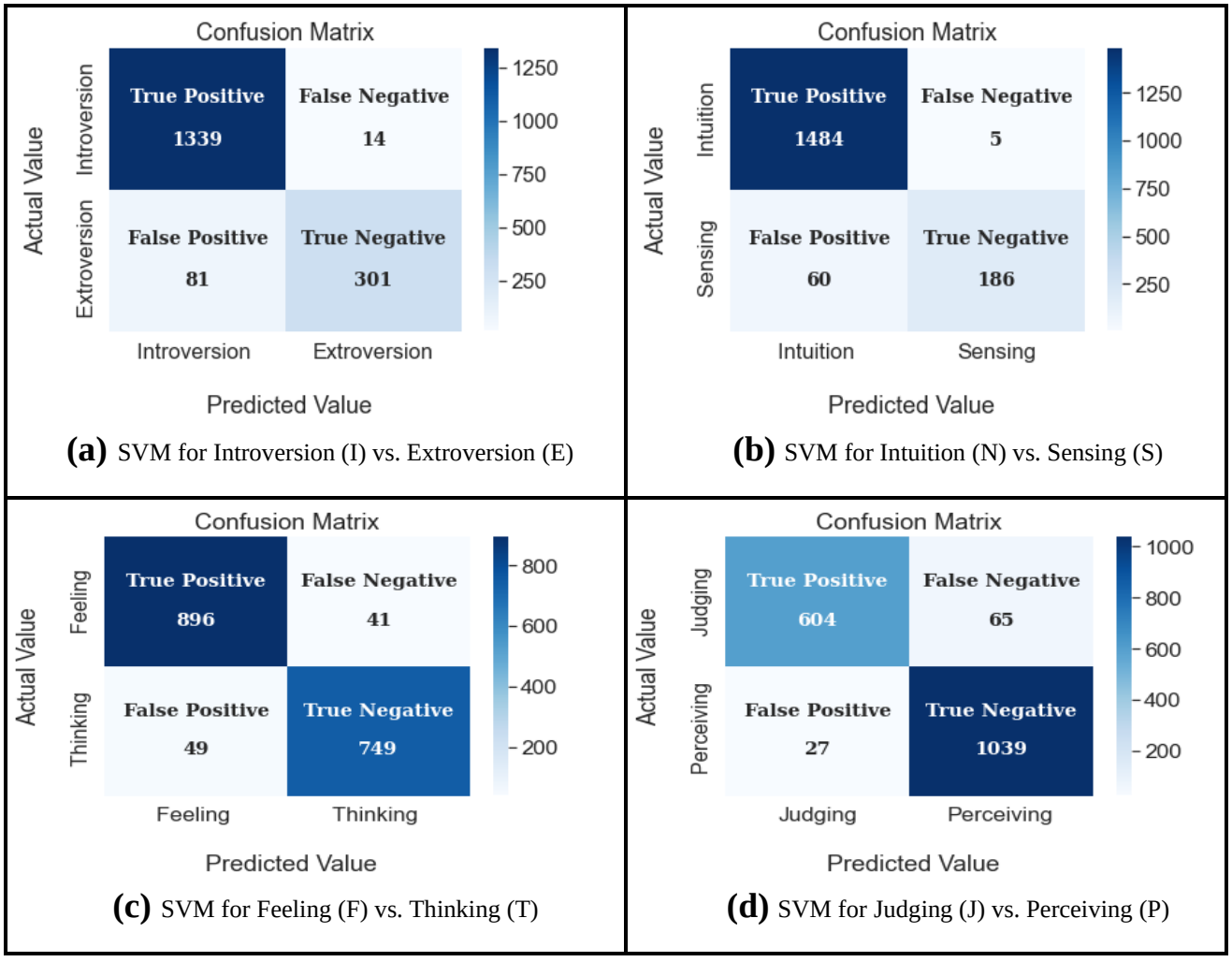
Figure 4.6: The confusion matrix with: stop-words, stemming, lemmatization, and with grid search optimizer for the Logistic Regression algorithm

Table 4.15 demonstrates that the use of SVM with stop-words, stemming, grid search optimizer, and lemmatization has achieved the

highest accuracy. According to Figure 4.7, the prediction of Intuition (N) vs. Sensing (S) is also better than other dimensions where the accuracy is 96.25%. Other studies also found similar findings [22, 36].

Table 4.15: The accuracy after different preprocessing techniques and with the integration of the grid search optimizer for the Support Vector Machine algorithm

Kaggle dataset from Personality Café Forum in 2017	Support Vector Classifier					The average accuracy of the four dimensions
	Metrics	IE	NS	FT	JP	
W stop-words, W stemming, W lemmatizing.	Accuracy	86.05	90.72	85.53	80.98	85.82
	Precision	83.0	89.0	85.0	81.0	84.5
	Recall	73.0	70.0	85.0	79.0	76.75
	F1-score	76.0	75.0	85.0	79.0	78.75
W stop-words, W stemming, W lemmatizing, W grid search optimizer	Accuracy	94.52	96.25	94.81	94.69	95.06
	Precision	94.92	96.74	94.81	94.91	95.34
	Recall	88.88	87.63	94.74	93.87	91.28
	F1-score	91.47	91.49	94.77	94.34	93.01
Best parameters	C = 1, gamma = 1, kernel = rbf					



(a) SVM for Introversion (I) vs. Extroversion (E)

(b) SVM for Intuition (N) vs. Sensing (S)

(c) SVM for Feeling (F) vs. Thinking (T)

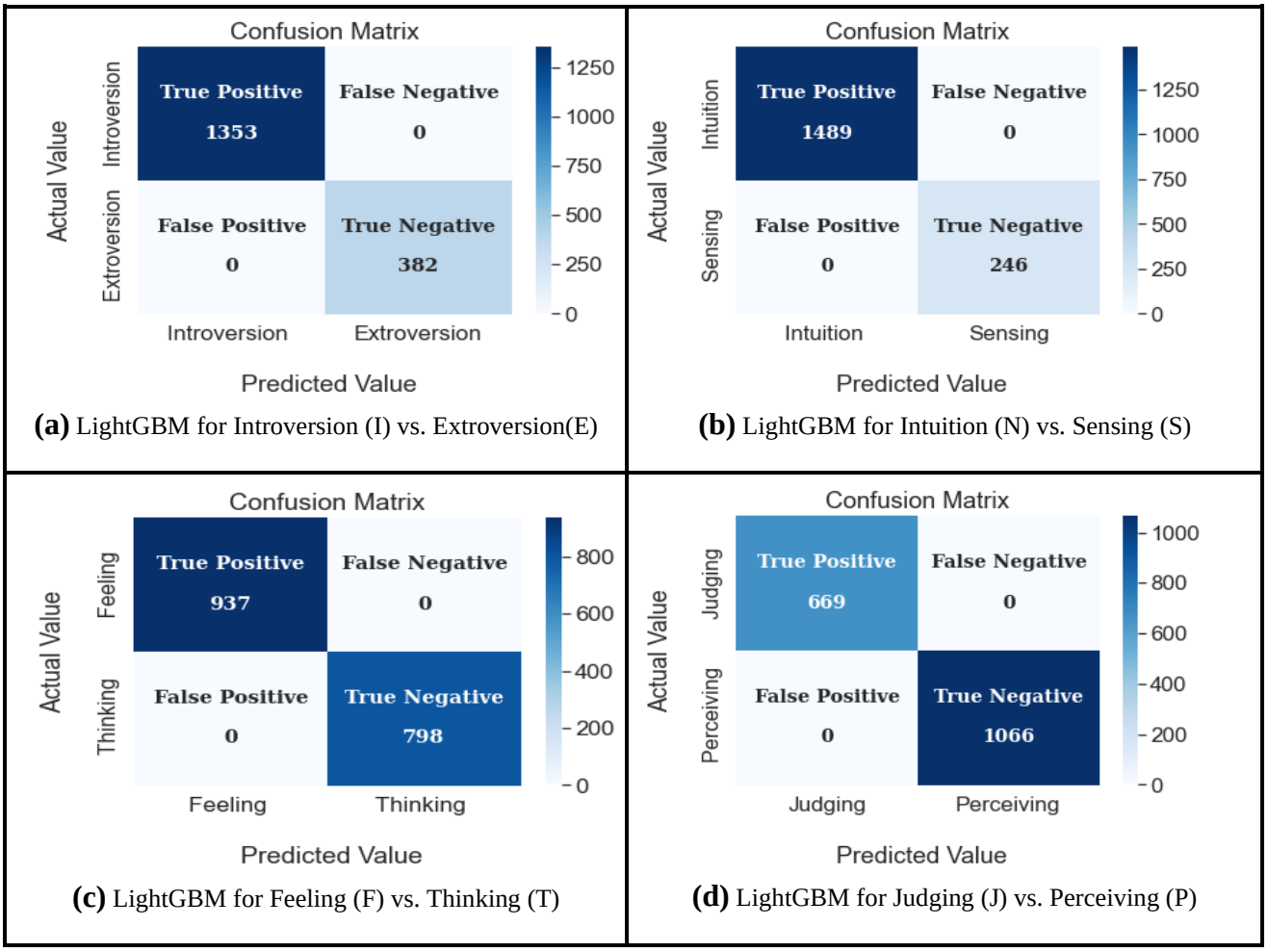
(d) SVM for Judging (J) vs. Perceiving (P)

Figure 4.7: The confusion matrix with: stop-words, stemming, lemmatization, and with grid search optimizer for the Support Vector classifier algorithm

Table 4.16 presents the use of LightGBM with stop-words, stemming, grid search optimizer, and lemmatization that has obtained the best accuracy. As depicted in Figure 4.8, the prediction of Intuition (N) vs. Sensing (S) is better than other dimensions (1.00%).

Table 4.16: The accuracy after different preprocessing techniques and with the integration of the grid search optimizer for the LightGBM algorithm

Kaggle dataset from Personality Café Forum in 2017	LightGBM					The average accuracy of the four dimensions
	Metrics	IE	NS	FT	JP	
W stop-words, **W** stemming, **W** lemmatizing.	Accuracy	86.92	91.87	84.55	82.94	86.57
	Precision	83.0	89.0	85.0	82.0	84.75
	Recall	77.0	75.0	84.0	81.0	79.25
	F1-score	79.0	80.0	84.0	82.0	81.25
W stop-words, **W** stemming, **W** lemmatizing, **W** grid search optimizer	Accuracy	1.00	1.00	1.00	1.00	1.00
	Precision	1.00	1.00	1.00	1.00	1.00
	Recall	1.00	1.00	1.00	1.00	1.00
	F1-score	1.00	1.00	1.00	1.00	1.00
Best parameters	learning_rate = 0.3, max_depth = 20, min_child_samples = 10, num_leaves = 40, reg_alpha = 0					



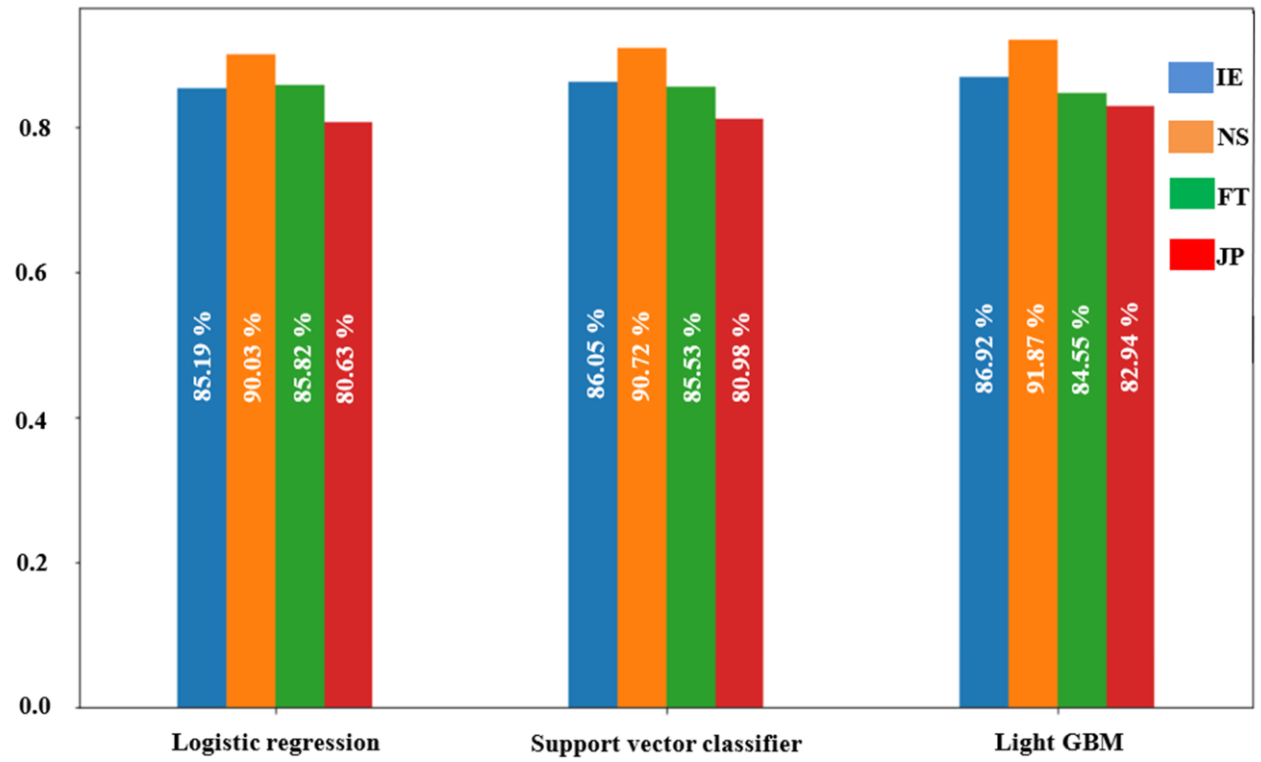
(a) LightGBM for Introversion (I) vs. Extroversion(E)

(b) LightGBM for Intuition (N) vs. Sensing (S)

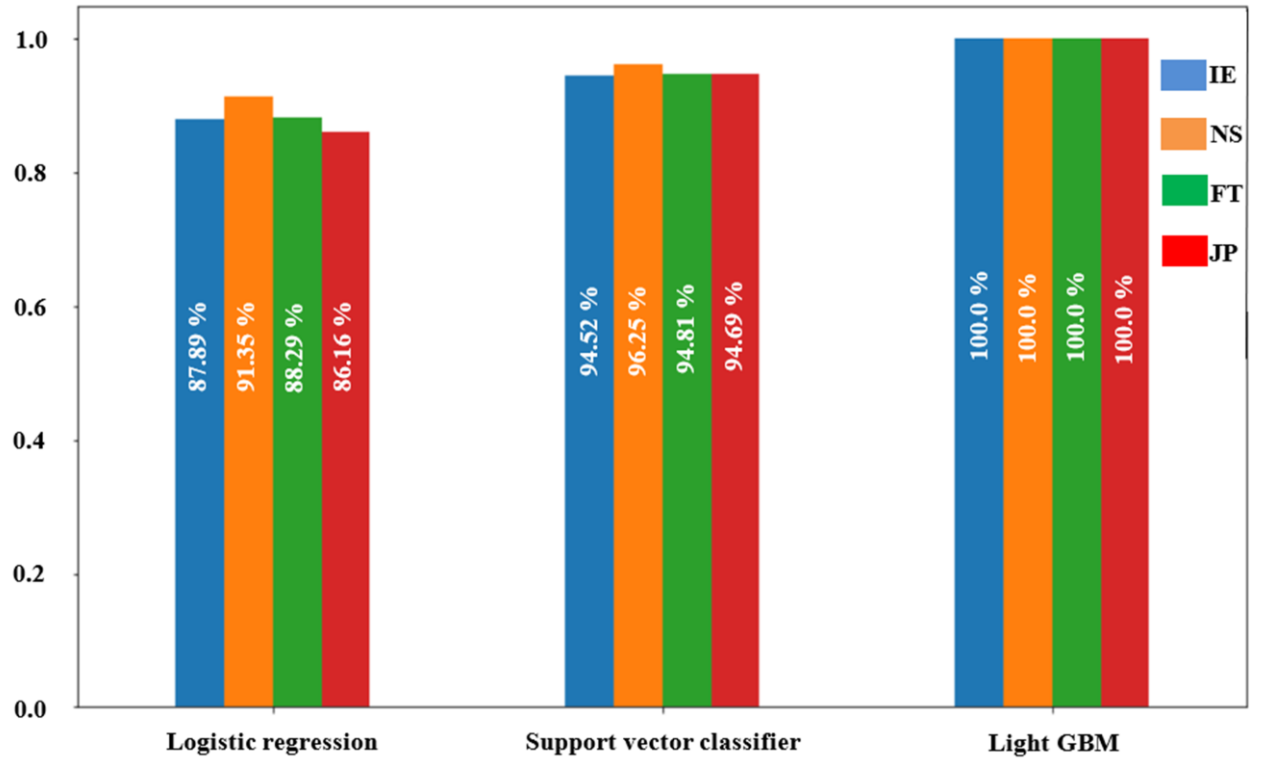
(c) LightGBM for Feeling (F) vs. Thinking (T)

(d) LightGBM for Judging (J) vs. Perceiving (P)

Figure 4.8: The confusion matrix with: stop-words, stemming, lemmatization, and with grid search optimizer for the LightGBM algorithm



a. The accuracy of the classifier s with: stop-words, stemming, lemmatization, and without grid search optimizer



b. The accuracy of the classifiers with: stop-words, stemming, lemmatization, and grid search optimizer

Figure 4.9: The accuracy of the classifiers with: stop-words, stemming, lemmatization, and with and without the grid search optimizer

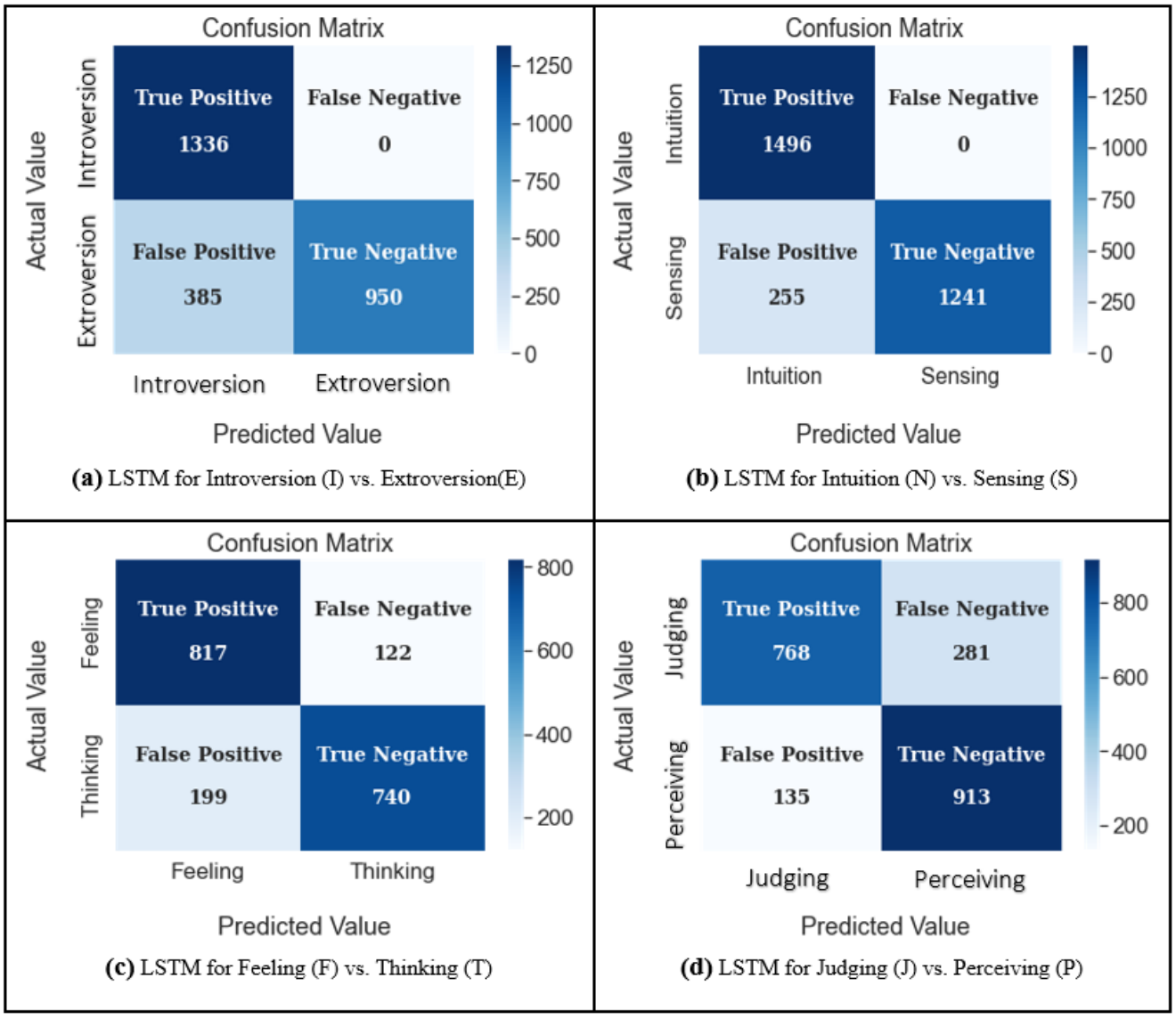
4.5.2 Long Short-Term Memory

After the preprocessing steps, classification, and performance evaluation stages for the classifier are performed. The accuracy of the obtained deep learning algorithm for each dimension is summarized in Table 4.17. Figure 4.10 shows the confusion matrix of the MBTI four dimensions. Parameters after different preprocessing techniques for Long Short-Term Memory algorithm are outlined in Table 4.18. It shows the baseline of the technique and then the development made in order to improve the prediction accuracy.

Table 4.17: The accuracy after different preprocessing techniques for the Long Short-Term Memory algorithm

Personality Café Forum in 2017	Long Short-Term Memory						The average accuracy of the four dimensions
	Epoch	Metrics	IE	NS	FT	JP	
Baseline-LSTM without preprocessing	25	Accuracy	70.43	76.94	63.80	57.69	67.21
	25	Precision	57.70	51.18	63.51	55.11	56.87
	25	Recall	58.02	51.10	63.39	55.05	56.89
	25	F1-score	57.84	51.13	63.42	55.07	56.86
LSTM with preprocessing	25	Accuracy	82.76	84.78	80.92	73.94	80.60
	25	Precision	76.92	65.03	80.84	72.64	73.85
	25	Recall	67.15	57.71	80.69	73.17	69.68
	25	F1-score	69.83	59.28	80.75	72.85	70.67
LSTM with SMOTE and preprocessing	25	Accuracy	80.40	89.51	80.74	74.23	81.22
	25	Precision	71.17	80.55	80.73	72.97	76.35
	25	Recall	68.46	71.49	80.93	73.57	73.61
	25	F1-score	69.57	74.83	80.71	73.19	74.57
LSTM with SMOTE and preprocessing and grid search optimizer	25	Accuracy	88.74	90.84	81.21	79.11	84.97
	25	Precision	89.03	91.67	81.34	79.57	85.40
	25	Recall	88.74	90.84	81.21	79.11	84.97
	25	F1-score	88.72	90.80	81.19	79.04	84.93
LSTM with SMOTE and preprocessing and grid search optimizer for Two layers	25	Accuracy	87.35	91.74	79.08	79.83	84.50
	25	Precision	88.93	92.48	80.93	79.91	85.56
	25	Recall	87.35	91.74	79.08	79.83	84.50
	25	F1-score	87.24	91.71	78.79	79.81	84.38

LSTM with SMOTE and preprocessing and grid search optimizer for Three layers	**25**	**Accuracy**	**85.58**	**91.47**	**82.90**	**80.16**	**85.02**
	25	**Precision**	**88.81**	**92.71**	**83.13**	**80.75**	**86.35**
	25	**Recall**	**85.58**	**91.47**	**82.90**	**80.16**	**85.02**
	25	**F1-score**	**85.27**	**91.41**	**82.87**	**80.06**	**84.90**
LSTM with SMOTE and preprocessing and grid search optimizer for Four layers	**25**	**Accuracy**	**85.62**	**92.14**	**83.81**	**67.85**	**82.35**
	25	**Precision**	**88.80**	**93.10**	**83.82**	**80.17**	**86.47**
	25	**Recall**	**85.61**	**92.14**	**83.81**	**67.87**	**82.35**
	25	**F1-score**	**85.32**	**92.10**	**83.81**	**64.21**	**81.36**



(a) LSTM for Introversion (I) vs. Extroversion(E)

(b) LSTM for Intuition (N) vs. Sensing (S)

(c) LSTM for Feeling (F) vs. Thinking (T)

(d) LSTM for Judging (J) vs. Perceiving (P)

Figure 4.10: The confusion matrix with: stop-words, stemming, lemmatization, and with grid search optimizer for the LSTM algorithm

Table 4.18: The parameters after different preprocessing techniques for the Long Short-Term Memory algorithm

Personality Café Forum in 2017	Epochs	Metrics	Activation	Optimizer	Number of layers
Baseline-LSTM without preprocessing	25	Accuracy	sigmoid	adam	1 = 24 Units
	Loss		Batch_size	The average accuracy of the four dimensions	
	binary_crossentropy		64	67.21	
LSTM with SMOTE, preprocessing and grid search optimizer	Epochs	Metrics	Activation	Optimizer	Number of layers
	25	Accuracy	sigmoid	adam	1 = 24 Units
	Loss		Batch_size	The average accuracy of the four dimensions	
	binary_crossentropy		64	84.97	
LSTM with SMOTE, preprocessing and grid search optimizer for Two layers	Epochs	Metrics	Activation	Optimizer	Number of layers
	25	Accuracy	sigmoid	adam	1 = 24 Units 2 = 64 Units
	Loss		Batch_size	The average accuracy of the four dimensions	
	binary_crossentropy		64	84.50	
LSTM with SMOTE, preprocessing and grid search optimizer for Three layers	Epochs	Metrics	Activation	Optimizer	Number of layers
	25	Accuracy	sigmoid	adam	1 = 24 Units 2 = 64 Units 3 = 64 Units
	Loss		Batch_size	The average accuracy of the four dimensions	
	binary_crossentropy		64	85.02	
LSTM with SMOTE, preprocessing and grid search optimizer for Four layers	Epochs	Metrics	Activation	Optimizer	Number of layers
	25	Accuracy	sigmoid	adam	1 = 24 Units 2 = 64 Units 3 = 64 Units 4 = 20 Units
	Loss		Batch_size	The average accuracy of the four dimensions	
	binary_crossentropy		64	82.35	

The results indicate that the LightGBM classifier performs significantly better than the other classifiers in terms of accuracy. Figure 4.5 and Figure 4.8 illustrate the confusion matrix of LightGBM without and with the implementation of the grid search optimizer, respectively. The use of the optimizer helps improve prediction accuracy. However, LightGBM takes a longer time to be executed than other techniques in machine learning algorithms while deep learning took several days (see Table 4.19). The

preprocessing stage is used to clean data. It helps remove unimportant words and symbols, return a word to its root form, obtain word stems, omit words that include two letters or less, and delete tags from comments and posts. The grid search optimizer is also applied, which helped select the best hyperparameters to improve the prediction accuracy. The cross-validation also led to balancing the dataset. Such operations contributed to a significant improvement in forecasting accuracy. Tables 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, and 4.17 display all findings of the algorithms with and without different steps of the preprocessing phase, whereas Table 4.19 shows the execution time in seconds of all classifiers.

Table 4.19: The time execution of all cases

Kaggle dataset from Personality Cafe Forum in 2017	Logistic Regression	Support Vector classifier	LightGBM	LSTM
	Execution time	Execution time	Execution time	Execution time
WO stop-words, W stemming, W lemmatizing.	2.5210 Seconds	140.3919 Seconds	31.0166 Seconds	Several days
W stop-words, W stemming, W lemmatizing.	2.4963 Seconds	146.0681 Seconds	30.35016 Seconds	
W stop-words, WO stemming, WO lemmatizing.	2.5544 Seconds	153.4194 Seconds	30.5234 Seconds	
W stop-words, WO stemming, W lemmatizing.	2.3739 Seconds	150.5366 Seconds	29.7117 Seconds	
W stop-words, W stemming, WO lemmatizing.	2.4486 Seconds	147.0819 Seconds	30.06909 Seconds	
WO stop-words, WO stemming, WO lemmatizing.	3.6067 Seconds	139.1776 Seconds	36.4450 Seconds	
WO stop-words, W stemming, WO lemmatizing.	2.38371 Seconds	140.1468 Seconds	31.3728 Seconds	
WO stop-words, WO stemming, W lemmatizing.	2.5342 Seconds	148.5092 Seconds	31.3742 Seconds	
W stop-words, W stemming, W lemmatizing, W grid search optimizer	59.2681 Seconds	2374.91606 Seconds	19939.0398 Seconds	

This thesis aimed at predicting users' personality types on SMSs by comparing the performance accuracy of four well-known classifiers. It also attempted to improve the prediction accuracy by generating many datasets from the original data using different preprocessing steps. Finally, the thesis integrated the grid search optimizer with the three classifiers while, random search optimizer with LSTM to further enhance the prediction accuracy. Table 4.20 compares the performance accuracy of this thesis with earlier literature. Although previous studies implemented many classifiers, Table 4.20 includes the findings of a classifier with the highest performance accuracy. Naïve Bayes, Gradient Boosting, LSTM, CNN, and SVM achieved the highest accuracy in earlier research. However, the overall findings of this study outperformed the findings of such literature on the same dataset as shown in Table 1.1. The overall accuracy obtained in this thesis is 100% based on LightGBM.

The rationale behind this may be the implementation of different preprocessing steps before building the classification model. Using SMOTE also helped solve the problem of the imbalanced dataset. Moreover, the use of the grid search optimizer improved the overall accuracy from about 86% to 100%. The findings suggest that LightGBM with the implementation of stemming, lemmatization, and removing stop-words as well as integrating the grid search optimizer can produce the best accuracy. Many reasons can be drawn behind such results. First, stemming helps reduce the derived words to their word stem [52], so it decreases the number of words in the corpus and correlates with the word with similar meanings. Second, another approach used which is similar to stemming is lemmatization, but it compares the words with a language dictionary [52], so this helps the classifiers better predict the labels. Third, the implementation of stop-words removal leads to eliminating low-information terms from the text [57] and this, in turn, reduced the size of the dataset. Finally, the integration of the grid search optimization identified the best hyperparameters [69]. Moreover, the K-fold cross-

validation is used with the grid search optimizer which splits the data into k parts and ensures that each part is used as a test set to obtain rid of imbalanced data and reduce overfitting.

Following such rigorous procedures in this thesis resulted in high prediction accuracy. The overall model performance is 100% for the four dimensions, whereas the classifier achieved 86.92%, 91.87%, 84.55%, and 82.94% for (IE), (NS), (FT), and (JP) respectively without the grid search optimization.

Although previous literature [25, 35, 36] used the same feature extraction methods, they followed different preprocessing steps and data split methods. In another research [22], the LightGBM also showed higher accuracy performance than other techniques, but the grid search optimizer is not used, resulting in lower prediction accuracy than in this study.

Table 4.20: A comparison between the findings of previous research and this study

Reference	Method	Results			
		IE	FT	SN	JP
[35]	SVM	84.9	88.4	87.0	78.8
	Neural Network	77.0	86.3	54.1	61.8
[36]	Random Forest	94.95	71.19	98.93	74.06
	KNN	93.79	96.82	53.67	76.16
[23]	Naïve Bayes	52.0	62.0	57.0	57.0
	BERT	60.0	67.0	63.0	59.0
[25]	XGBoost	79.0	85.9	74.1	65.4
[37]	BI-LSTM	83.59	93.22	80.00	77.40
	SVM	82.15	87.32	80.49	72.70
[38]	BERT + MLP	78.8	86.3	76.1	67.2
[39]	BERT	75.83	74.41	75.75	71.90
[33]	LSTM	89.51	89.84	69.09	67.65
This study with grid search optimizer	LSTM	85.58	91.47	82.90	80.16
This study without grid search optimizer	LightGBM	86.92	91.87	84.55	82.94
This study with grid search optimizer	LightGBM	100.0	100.0	100.0	100.0

4.5.3 Comparing the Results of Machine Learning and Deep Learning Techniques

The three machine learning models implemented in this thesis have better predictive performance than deep learning. Such findings are in agreement with previous literature that showed the application of machine learning algorithms outperformed the performance accuracy of deep learning [22, 25, 36] whereas other studies that implemented deep learning achieved an overall accuracy between 59% and 88% [71, 72]. On the other hand, the application of machine learning algorithms outperformed the performance accuracy of deep learning [22, 25, 36]. This could be attributed to two possible reasons. The first is that deep learning techniques need huge data to learn adequately [73].

On the other hand, the time for (training - testing) the machine learning models makes them more useful, according to the specifications mentioned in section 4.2. The average time is taken for all models to (train - test) based on the dataset inputted (see Table 4.19). These values (time) will either increase or decrease depending on the specification of the device utilized if other researchers want to work on this topic.

CHAPTER FIVE

CONCLUSION AND FUTURE WORKS

CHAPTER FIVE

CONCLUSION AND FUTURE WORKS

5.1 Thesis Summary

- Several preprocessing methods are used before passing the study data to the classifier.
- Personalities behavior analysis is performed to categorize users' personalities on SMSs.
- The features extraction technique is utilized to extract significant features from raw text, whereas feature selection is utilized to avoid uninformative features and choose the most important.
- Light Gradient Boosting Machine (LightGBM), Logistic Regression (LR), Long Short-Term Memory (LSTM), and Support Vector Machine (SVM) classifiers are found to be the most effective algorithms for analyzing personality behavior on SMSs.

5.2 Conclusions

1- This study aims at implementing machine learning methods to automate personality type prediction based on one of the most used personality models which is MBTI.

2- Natural language processing is used to achieve this aim.

3- The accuracy, time, and performance of the four algorithms are evaluated.

4- The grid and random search optimizers show a significant enhancement in the performance accuracy of the four dimensions.

5- In order to achieve better accuracy and reliability, the thesis presented a methodology that greatly improved the accuracy of predicting the

four personality dimensions of the MBTI model.

6- The accuracy obtained is 100% based on the LightGBM algorithm for the four dimensions.

7- This can actively assist NLP practitioners and psychologists in the identification of personality types and associated cognitive processes on SMSs.

8- The results of this thesis proved that personality analysis through text on SMSs is an important factor in predicting the users' personalities.

5.3 Future Work

Several directions could be highlighted based on the findings of the present thesis. A recommendation system can be built to predict users' personalities on SMSs through their profiles, images, and likes. Regardless of such important outcomes, this thesis is not without limitations.

- The thesis is based on one dataset, so it is necessary to apply the proposed model to other SMSs. This can confirm the validity of the existing results.
- The proposed model is implemented with traditional machine learning and deep learning techniques, whereas implementing other methods such as Monarch Butterfly Optimization (MBO) [74], Earth-worm Optimization algorithm (EWA) [75], Elephant Herding Optimization (EHO) [76], and Moth Search (MS) algorithm [75] may help provide further research directions.
- Implementing deep learning to predict the personality's dimensions requires a long time, so researchers are invited to work on reducing the execution time for such algorithms.

REFERENCES

[1] C. Chan and M. J. Holosko, “The utilization of social media for youth outreach engagement: A case study,” *Qualitative Social Work*, vol. 16, no. 5, pp. 680–697, Sep. 2017, doi: 10.1177/1473325016638917.

[2] S. Weshah, E. Alazzam, Q. Aldabbas, Z. Obeidat, M. Humeedat, and Y. AlQudah, “The impact of social media applications on predicting stock ’ s p rices and exchange volume : the case of Jordan,” *Acad. Strateg. Manag. J.*, vol. 20, no. 6, pp. 1–10, 2021.

[3] X. Luo, “Efficient English text classification using selected Machine Learning Techniques,” *Alexandria Engineering Journal*, vol. 60, no. 3, pp. 3401–3409, Jun. 2021, doi: 10.1016/J.AEJ.2021.02.009.

[4] B. Caci, M. Cardaci, M. E. Tabacchi, and F. Scrima, “Personality variables as predictors of facebook usage,” *Psychological Reports*, vol. 114, no. 2, pp. 528–539, Apr. 2014, doi: 10.2466/21.09.PR0.114k23w6.

[5] J. Marie Condie, I. Ayodele, S. Chowdhury, S. Powe, and A. M. Cooper, “Personalizing twitter communication: an evaluation of ‘rotation-curation’ for enhancing social media engagement within higher education,” *Journal of Marketing for Higher Education*, vol. 28, no. 2, pp. 192–209, Jul. 2018, doi: 10.1080/08841241.2018.1453910.

[6] J. A. Golbeck, “Predicting Personality from Social Media Text,” *AIS Transactions on Replication Research*, vol. 2, no. 1, p. 2, Sep. 2016, doi: 10.17705/1atrr.00009.

[7] G. Drakopoulos, A. Kanavos, P. Mylonas, and P. Pintelas, “Extending Fuzzy Cognitive Maps with Tensor-Based Distance Metrics,” *Mathematics 2020*, Vol. 8, Page 1898, vol. 8, no. 11, p. 1898, Oct. 2020, doi: 10.3390/MATH8111898.

[8] M. Y. Mun and S. Y. Hwang, “Impact of DISC Behavioral Styles on Job Satisfaction and Clinical Competencies among Newly Hired Nurses,” *Journal of Korean Academy of Nursing Administration*, vol. 21, no. 1, p. 43, 2015, doi: 10.11111/jkana.2015.21.1.43.

[9] N. Ahmad and J. Siddique, “Personality Assessment using Twitter Tweets,” *in Procedia Computer Science*, 2017, vol. 112, pp. 1964–1973, doi:

10.1016/j.procs.2017.08.067.

[10] K. Akbar, Y. Jin, M. Mahsud, M. Akbar, A. Waheed, and R. Amin, “Role of Big Five Personality Traits in Sustainable Consumption Behavior,” *in ACM International Conference Proceeding Series*, Sep. 2020, pp. 222–226, doi: 10.1145/3422713.3422750.

[11] A. Behaz and M. Djoudi, “Adaptation of learning resources based on the MBTI theory of psychological types,” *International Journal of Computer Science Issues (IJCSI)*, ISSN : 1694-0814, vol. 9, no. 2, p. pp 135—-141, 2012.

[12] A. Eryilmaz, “Perceived Personality Traits and Types of Teachers and Their Relationship to the Subjective Well-being and Academic Achievements of Adolescents,” *Educational Sciences: Theory & Practice*, vol. 14, no. 6, pp. 2049–2063, 2014, doi: 10.12738/estp.2014.6.2187.

[13] R. F. Krueger and N. R. Eaton, “Personality traits and the classification of mental disorders: toward a more complete integration in DSM-5 and an empirical model of psychopathology,” *Personality disorders*, vol. 1, no. 2, pp. 97–118, 2010, doi: 10.1037/A0018990.

[14] K. Razavipour, “Review of The Palgrave Handbook of Applied Linguistics Research Methodology, by Aek, Phakiti; Peter De Costa; Luke, Plonsky; & Sue, Starfield,” *Asian-Pacific Journal of Second and Foreign Language Education*, vol. 5, no. 1, Dec. 2020, doi: 10.1186/S40862-020-00095-X.

[15] J. K. Boparai, S. Singh, and P. Kathuria, “How to Design and Validate A Questionnaire: A Guide,” *Current Clinical Pharmacology*, vol. 13, no. 4, pp. 210–215, Aug. 2018, doi: 10.2174/1574884713666180807151328.

[16] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernández, “Syntactic N-grams as machine learning features for natural language processing,” *Expert Systems with Applications*, vol. 41, no. 3, pp. 853–860, 2014, doi: 10.1016/j.eswa.2013.08.015.

[17] M. L. Kern, P. X. McCarthy, D. Chakrabarty, and M. A. Rizoiu, “Social media-predicted personality traits and values can help match people to their ideal jobs,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 116, no. 52, pp. 26459–26464, Dec. 2019, doi:

10.1073/PNAS.1917942116/-/DCSUPPLEMENTAL.

[18] A. G. Reece, C. M. D.-E. P. J. D. Science, and undefined 2017, "Instagram photos reveal predictive markers of depression," *Springer*, Available: https://link.springer.com/content/pdf/10.1140/epjds/s13688-017-0110-z.pdf.

[19] R. Safa, P. Bayat, and L. Moghtader, Automatic detection of depression symptoms in twitter using multimodal analysis, vol. 78, no. 4. *Springer US*, 2022, doi: 10.1007/s11227-021-04040-8.

[20] L. M. Braunstein, J. J. Gross, and K. N. Ochsner, "Explicit and implicit emotion regulation: A multi-level framework," *Social Cognitive and Affective Neuroscience*, vol. 12, no. 10, pp. 1545–1557, 2017, doi: 10.1093/scan/nsx096.

[21] A. Balahur, J. M. Hermida, and A. Montoyo, "Detecting implicit expressions of emotion in text: A comparative analysis," *in Decision Support Systems*, Nov. 2012, vol. 53, no. 4, pp. 742–753, doi: 10.1016/j.dss.2012.05.024.

[22] E. J. Choong and K. D. Varathan, "Predicting judging-perceiving of Myers-Briggs Type Indicator (MBTI) in online social forum," *PeerJ*. 2021, doi: 10.7717/peerj.11382.

[23] P. Kadamb, "Exploring Personality and Online Social Engagement : An Investigation of MBTI Users on Twitter," *arXiv Prepr*. arXiv2109.06402, 2022.

[24] S. Chaudhary, R. Singh, S. T. Hasan, and M. I. Kaur, "IRJET-A Comparative Study of Different Classifiers for Myers-Brigg Personality Prediction Model," *Int. Res. J. Eng. Technol.*, vol. 5, no. 5, pp. 1410–1413, 2018.

[25] M. H. Amirhosseini and H. Kazemian, "Machine learning approach to personality type prediction based on the Myers–Briggs type indicator®," *Multimodal Technologies and Interaction*, vol. 4, no. 1, Mar. 2020, doi: 10.3390/mti4010009.

[26] Q. Gong et al., "Cross-site prediction on social influence for cold-start users in online social networks," *ACM Transactions on the Web*, vol. 15, no. 2, 2021, doi: 10.1145/3409108.

[27] R. Bin Tareaf, P. Berger, P. Hennig, and C. Meinel, "Cross-platform

personality exploration system for online social networks: Facebook vs. Twitter," *Web Intelligence*, vol. 18, no. 1, pp. 35–51, 2020, doi: 10.3233/WEB-200427.

[28] S. Chang et al., "Dilated Recurrent Neural Networks," *Adv. neural Inf. Process. Syst.* 30, 2017, https://doi.org/10.48550/arXiv.1710.02224.

[29] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks," *2005 IEEE International Joint Conference on Neural Networks*, vol. 4, pp. 2047–2052, 2005, doi: 10.1109/IJCNN.2005.1556215.

[30] K. Cho et al., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *arXiv preprint arXiv:1406.1078*, 2014, doi: org/10.48550/arXiv.1406.1078.

[31] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," *Adv. neural Inf. Process. Syst.* 31, 2018, https://doi.org/10.48550/arXiv.1706.09516.

[32] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," *in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, vol. 13-17-August-2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[33] B. Cui and C. Qi, "Survey Analysis of Machine Learning Methods for Natural Language Processing for MBTI Personality Type Prediction*," Lel. stanford Jr. Univ.*, 2017.

[34] L. Liu, D. Preoţiuc-Pietro, Z. R. Samani, M. E. Moghaddam, and L. Ungar, "Analyzing personality through social media profile picture choice," *in Proceedings of the 10th International Conference on Web and Social Media, ICWSM 2016*, 2016, no. Icwsm, pp. 211–220, Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14738.

[35] S. Bharadwaj, S. Sridhar, R. Choudhary, and R. Srinath, "Persona Traits Identification based on Myers-Briggs Type Indicator(MBTI) - A Text Classification Approach," *2018 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2018*, pp. 1076–1082, 2018, doi: 10.1109/ICACCI.2018.8554828.

[36] M. Maulidah, and H. F. Pardede, "Prediction Of Myers-Briggs Type Indicator Personality Using Long Short-Term Memory," *Jurnal Elektronika dan Telekomunikasi*, vol. 21, no. 2, pp. 104-111, Dec. 2021, doi: 10.14203/jet.v21.104-111.

[37] S. Ontoum and J. H. Chan, "Personality Type Based on Myers-Briggs Type Indicator with Text Posting Style by using Traditional and Deep Learning" *arXiv2201.08717 [cs]*, Jan. 2022, http://arxiv.org/abs/2201.08717.

[38] Y. Mehta, S. Fatehi, A. Kazameini, C. Stachl, E. Cambria, and S. Eetemadi, "Bottom-up and top-down: Predicting pesonality with psycholinguistic and language model features," *Proceedings - IEEE International Conference on Data Mining, ICDM*, vol. 2020-Novem, pp. 1184–1189, Nov. 2020, doi: 10.1109/ICDM50108.2020.00146.

[39] S. S. Keh and I.-T. Cheng, "Myers-Briggs Personality Classification and Personality-Specific Language Generation Using Pre-trained Language Models," *ArXiv*, 2019, doi: doi.org/10.48550/arXiv.1907.06333.

[40] C. Chan and S. S. yum Ngai, "Utilizing social media for social work: insights from clients in online youth services," *Journal of Social Work Practice*, vol. 33, no. 2, pp. 157–172, Apr. 2019, doi: 10.1080/02650533.2018.1504286.

[41] M. Tyagi, "Social Media and its Impact on Indian Society," *World Digit. Libr. - An Int. J.*, vol. 15, pp. 27–42, 2022, doi: 10.18329/09757597/2022/15103.

[42] J. A. Golbeck, "Predicting Personality from Social Media Text," *AIS Trans. Replication Res.*, vol. 2, no. 1, p. 2, Sep. 2016, doi: 10.17705/1atrr.00009.

[43] A. Furnham and J. Crump, "The Myers-Briggs Type Indicator (MBTI) and Promotion at Work," *Psychology*, vol. 06, no. 12, pp. 1510–1515, 2015, doi: 10.4236/PSYCH.2015.612147.

[44] M. Farrell, "Leadership Reflections: Extrovert and Introvert Leaders," *J. Libr. Adm.*, vol. 57, no. 4, pp. 436–443, 2017, doi: 10.1080/01930826.2017.1300455.

[45] R. Müller, M. Martinsuo, and T. Blomquist, "Project Portfolio Control and Portfolio Management Performance in Different Contexts," *Proj. Manag. J.*,

vol. 39, no. September, pp. 28–42, 2008, doi: 10.1002/pmj.

[46] K. Al Sharou, Z. Li, and L. Specia, “Towards a Better Understanding of Noise in Natural Language Processing,” *Int. Conf. Recent Adv. Nat. Lang. Process. RANLP*, pp. 53–62, 2021, doi: 10.26615/978-954-452-072-4_007.

[47] A. Zia, M. Aziz, I. Popa, S. A. Khan, A. F. Hamedani, and A. R. Asif, “Artificial Intelligence-Based Medical Data Mining,” *Journal of Personalized Medicine*, vol. 12, no. 9. MDPI, Sep. 01, 2022, doi: 10.3390/jpm12091359.

[48] C. A. Palacios, J. A. Reyes-Suárez, L. A. Bearzotti, V. Leiva, and C. Marchant, “Knowledge Discovery for Higher Education Student Retention Based on Data Mining: Machine Learning Algorithms and Case Study in Chile,” Entropy, vol. 23, no. 4, p. 485, 2021, doi: https://doi.org/10.3390/e23040485.

[49] P. J. Grace and N. N. Banu, “Machine Learning on Emotional Intelligence and Work Life Balance,” *International Journal of Computer Applications*, vol. 116, no. 10, pp. 975–8887, 2015.

[50] V. Gupta and G. S. Lehal, “A survey of text mining techniques and applications,” *J. Emerg. Technol. Web Intell.*, vol. 1, no. 1, pp. 60–76, 2009, doi: 10.4304/jetwi.1.1.60-76.

[51] F. S. Gharehchopogh and Z. A. Khalifelu, “Analysis and evaluation of unstructured data: Text mining versus natural language processing,” *2011 5th International Conference on Application of Information and Communication Technologies (AICT)*, doi: 10.1109/ICAICT.2011.6111017.

[52] A. S. Khan, H. Ahmad, M. Z. Asghar, F. K. Saddozai, A. Arif, and H. A. Khalid, “Personality classification from online text using machine learning approach,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 3, pp. 460–476, 2020, doi: 10.14569/ijacsa.2020.0110358.

[53] L. Kotthoff, I. P. Gent, and I. Miguel, “An evaluation of machine learning in algorithm selection for search problems,” *AI Commun.*, vol. 25, no. 3, pp. 257–270, 2012, doi: 10.3233/AIC-2012-0533.

[54] S. Garg and A. Garg, “Comparison of machine learning algorithms for content

based personality resolution of tweets," *Soc. Sci. Humanit. Open*, vol. 4, no. 1, p. 100178, 2021, doi: 10.1016/j.ssaho.2021.100178.

[55] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, and S. Homayouni, "Support Vector Machine Versus Random Forest for Remote Sensing Image Classification: A Meta-Analysis and Systematic Review," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 13, pp. 6308–6325, 2020, doi: 10.1109/JSTARS.2020.3026724.

[56] S. D. Dorle and N. N. Pise, "Political Sentiment Assessment through Social Media," *1st International Conference on Data Science and Analytics, PuneCon 2018 - Proceedings, no. Iccmc*, pp. 869–873, 2018, doi: 10.1109/PUNECON.2018.8745416.

[57] W. Zhang, T. Yoshida, and X. Tang, "A comparative study of TF*IDF, LSI and multi-words for text classification," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2758–2765, 2011, doi: 10.1016/j.eswa.2010.08.066.

[58] D. E. Cahyani and I. Patasik, "Performance comparison of TF-IDF and Word2Vec models for emotion text classification," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 5, pp. 2780–2788, Oct. 2021, doi: https://doi.org/10.11591/eei.v10i5.3157.

[59] A. Subakti, H. Murfi, and N. Hariadi, "The performance of BERT as data representation of text clustering," *Journal of Big Data*, vol. 9, no. 1, Feb. 2022, doi: https://doi.org/10.1186/s40537-022-00564-9.

[60] E. M. Dharma, F. L. Gaol, H. Leslie, H. S. Warnars, and B. Soewito, "the Accuracy Comparison Among Word2Vec, Glove, and Fasttext Towards Convolution Neural Network (CNN) Text Classification," *Journal of Theoretical and Applied Information Technology*, vol. 31, no. 2, 2022.

[61] V. Srividhya and R. Anitha, "Evaluating Preprocessing Techniques in Text Categorization," *Int. J. Comput. Sci. Appl.*, vol. 47, no. 11, pp. 49–51, 2010, Available: http://sinhgad.edu/ijcsa-2012/pdfpapers/1_11.pdf .

[62] X. Wu et al., "Top 10 algorithms in data mining," *Knowledge and Information Systems, vol*. 14, no. 1, pp. 1–37, 2008, doi: 10.1007/s10115-007-0114-2.

[63] C. Esposito, G. A. Landrum, N. Schneider, N. Stiefl, and S. Riniker, “GHOST: Adjusting the Decision Threshold to Handle Imbalanced Data in Machine Learning,” *Journal of Chemical Information and Modeling*, vol. 61, no. 6, pp. 2623–2640, Jun. 2021, doi: https://doi.org/10.1021/acs.jcim.1c00160.

[64] N. Saranya, “Classification of Soil and Crop Suggestion using Machine Learning Techniques,” *Int. J. Eng. Res. Technol.*, vol. 9, no. 02, pp. 671–673, 2020.

[65] D. Zhang and Y. Gong, “The Comparison of LightGBM and XGBoost Coupling Factor Analysis and Prediagnosis of Acute Liver Failure,” *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3042848.

[66] C. Tang, N. Luktarhan, and Y. Zhao, “An efficient intrusion detection method based on LightGBM and autoencoder,” *Symmetry*, vol. 12, no. 9, p. 1458, 2020, doi: 10.3390/sym12091458.

[67] Y. Yu, X. Si, C. Hu, and J. Zhang, “A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures,” *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019, doi: https://doi.org/10.1162/neco_a_01199.

[68] T. Zhu, Y. Lin, and Y. Liu, “Synthetic minority oversampling technique for multiclass imbalance problems,” *Pattern Recognit.*, vol. 72, pp. 327–340, 2017, doi: 10.1016/j.patcog.2017.07.024.

[69] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.

[70] J. Dhar, “An adaptive intelligent diagnostic system to predict early stage of parkinson’s disease using two-stage dimension reduction with genetically optimized lightgbm algorithm,” *Neural Computing and Applications*, vol. 34, pp. 4567–4593, 2022, doi: https://doi.org/10.1007/s00521-021-06612-4.

[71] D. Xue et al., “Deep learning-based personality recognition from text posts of online social networks*,” in Applied Intelligence (2018),* 2018, vol. 48, no. 11, pp. 4232–4246, doi: 10.1007/s10489-018-1212-4.

[72] A. Bhoi, S. P. Pujari, and R. C. Balabantaray, “A deep learning-based social media text analysis framework for disaster resource management,” *Soc.*

Netw. Anal. Min., vol. 10, no. 1, 2020, doi: 10.1007/s13278-020-00692-1.

[73] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *J. Big Data*, vol. 2, no. 1, pp. 1–21, 2015, doi: 10.1186/s40537-014-0007-7.

[74] G. G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 1995–2014, 2019, doi: 10.1007/s00521-015-1923-y.

[75] G. Wang, "Moth search algorithm : a bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Comput.*, vol. 10, pp. 151–164, 2016, doi: 10.1007/s12293-016-0212-3.

[76] G. Wang, S. Deb, and L. dos. . Coelho, "Elephant Herding Optimization," *in 3rd International Symposium on Computational and Business Intelligence*, 2015, pp. 1–5, doi: 10.1109/ISCBI.2015.8.

Appendix A The Published Paper

Karbala International Journal of Modern Science

Volume 8 | Issue 4 Article 5

Predicting Users' Personality on Social Media: A Comparative Study of Different Machine Learning Techniques

Ali Saadi Al-Fallooji
Department of Software/ College of Information Technology/ University of Babylon,
alisadee.sw.msc@student.uobabylon.edu.iq

Ahmed Al-Azawei
Department of Software/ College of Information Technology/ University of Babylon

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعـــة بابـــــل
كلية تكنولوجيا المعلومات
قسم البرمجيات

تصنيف شخصيات المستخدمين على وسائل التواصل الاجتماعي باستخدام آلة تعزيز التدرج الخفيف وتقنيات التحسين

رسالة مقدمة الى
مجلس كلية تكنولوجيا المعلومات – جامعة بابل كجزء من متطلبات نيل درجة الماجستير في تكنولوجيا المعلومات / البرمجيات

من قبل
علي سعدي عباس علي
بإشراف
أ. م. د. أحمد حبيب سعيـد العزاوي

١٤٤٤هـ ٢٠٢٣م

تصنيف شخصيات المستخدمين على وسائل التواصل الاجتماعي باستخدام آلة تعزيز التدرج الخفيف وتقنيات التحسين

أصبح استخدام مواقع التواصل الاجتماعي (Social Media Sites) في جميع أنحاء العالم ضرورة لابد منها، ويتضح ذلك من خلال تزايد عدد المستخدمين بشكل ملحوظ. وقد أدى ذلك إلى توظيف هذه المواقع من قبل الشركات السوقية والتجارية والتعليمية لتقديم محتوى يلبي احتياجات المستخدمين الشخصية، وهذا الأمر يتطلب تحديد شخصيات المستخدمين للاستجابة لتفضيلاتهم الفردية.

يهدف هذا البحث إلى: أولاً، تحليل مشاركات المستخدمين على مواقع التواصل الاجتماعي للتنبؤ بشخصيتهم بناءً على نموذج مؤشر نوع مايرز بريجز(Myers Briggs Type Indicator) . مقارنة دقة أداء تقنيات المعالجة المسبقة المختلفة واستخراج البيانات. أخيرًا، تحسين دقة التنبؤ بأنواع شخصيات المستخدمين. تتضمن مجموعة البيانات المستخدمة 8668 مستخدم حيث يحتوي كل صف على خمسين منشوراً لكل مستخدم.

يتم تطبيق أربع تقنيات لاستخراج البيانات. يتضمن ذلك آلة المتجهات الداعمة (SVM) والانحدار اللوجستي (LR) وآلة تعزيز التدرج الخفيف (LightGBM) والذاكرة طويلة المدى (LSTM). تم دمج طريقتين للتحسين في النظام المقترح.

تشير النتائج إلى أن تقنية LightGBM بأستخدام lemmatization، تحسين البحث الشبكي (Grid Search Optimizer) وإزالة كلمات التوقف قد تفوقت على التقنيات الأخرى. وتبلغ دقة التنبؤ لأبعاد الشخصيات الأربعة وهي: الانطواء - الانبساط (Introversion-Extroversion)، الحدس - الاستشعار (Intuition-Sensing)، الشعور - التفكير (Feeling-Thinking)، الحكم - الإدراك (Judgment-Perception) هي 100٪. أما بأستخدام تقنية الذاكرة طويلة المدى (Long Short-Term Memory) فقد كانت الدقة 85.02٪. أن نتائج البحث واعدة، حيث تم تحديد الأبعاد الأربعة لـ MBTI بشكل فعال، ومقارنة هذه النتائج مع الأبحاث السابقة حول التنبؤ الشخصي، وبالتالي يمكن أن تساعد نتائج الأطروحة مزودي مواقع التواصل الاجتماعي والشركات والمؤسسات التعليمية على تكييف مواقعهم على الإنترنت بناءً على مشاركات المستخدمين وتغريداتهم وتعليقاتهم التي يمكن استخدامها للتنبؤ بسلوك شخصيتهم.