A Comparison of Machine Learning Techniques to Classify Tweets relevant to People impacted by Dementia and COVID-19

A Thesis Submitted to the
College of Graduate and Postdoctoral Studies
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Mehrnoosh Azizi

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Or

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9
Canada

Abstract

Dementia has emerged as one of today's biggest healthcare challenges due to the increasing demand for medical, social, and institutional care. Moreover, the COVID-19 pandemic has had a unique impact on people with dementia. Those with dementia are also at an increased risk of contracting COVID-19, as well as having more severe symptoms and disease consequences. This highlights the importance of focusing on the issues of people living with dementia.

Modern technologies including social media can help psychologists to analyze people's experiences and take necessary measures. However, one of the principal problems for psychologists is that they must process huge amounts of data, but not all of the data can be analyzed due to a lot of irrelevant information in the data. Therefore, the data need to be labeled manually either by one or several researchers, which is a tedious and time-consuming task and may be costly due to the human effort involved. Thus, improvements to existing methodologies are needed to enable psychologists to make better use of the data and understand the impacts of COVID-19 on people with dementia.

Nowadays, one of the modern and reasonable ways perform a task (e.g., automatic labeling) is to use Machine Learning (ML) algorithms to save time and energy. To this end, this study compares various ML algorithms to classify tweets relevant to dementia and COVID-19 in order to help psychologist examine the COVID-19 impacts on people living with dementia.

In this case, three different datasets are used: (i) a dataset comprised of 5,058 tweets extracted from Twitter on COVID-19 and dementia from February 15 to September 7, 2020 to train, evaluate, and compare different models, (ii) a dataset comprised of 6,240 tweets from September 8, 2020 to December 8, 2021 to test the best model, and (iii) a dataset comprised of 1,289 tweets related to Canada's Alzheimer's Awareness Month from January 1 to January 31, 2022 to retrain and test the best model.

In the first step, to choose the best machine learning model, several classification models, including logistic regression, Gaussian naïve Bayes classifier, multinomial naïve Bayes classifier, support vector classifier, decision tree classifier, K-nearest neighbor classifier, random forest classifier, AdaBoost classifier, XGBoost classifier, BERT classifier, and ALBERT classifier are trained and compared in terms of classification performance. According to the classification results, the ALBERT model outperformed all other models in the comparison and achieved the least over-fitting problem and the highest accuracy, AUC, and F1-score compared to the other explored models. In the second step, the ALBERT model is tested on the second dataset (a completely unseen dataset) and achieved an accuracy of 80% in classifying relevant and irrelevant tweets for people impacted by dementia and COVID-19. Finally, to show that the ALBERT model can be used for future studies in the context of people impacted by dementia and COVID-19 in an efficient way, the model is trained on 10% of the third dataset and tested using 90% of the rest and reached an accuracy of 88%.

Acknowledgements

First and foremost, I deeply thank God Almighty for the blessings He has bestowed upon me and for giving me the strength and wisdom to achieve this dream. This master thesis is due to the support and encouragement of many people. It is a pleasure to express my sincere thanks to all those who helped me for the success of this study.

I would like to express my sincere gratitude to my supervisor Prof. Raymond J. Spiteri for the continuous support of my M.Sc. study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better supervisor and mentor for my M.Sc. study.

Also, I would like to thank my dear husband, Max, who supported me spiritually during my married life especially my M.Sc. study. In addition, I would like to thank my lovely mother, who supported and encouraged me a lot throughout my entire life to achieve this goal.

To my family and my beloved father who is sadly no more.

Contents

List of Tables

List of Figures

List of Abbreviations

AUC	Area Under the Curve
CSV	Comma-Separated Value
DT	Decision Tree
FN	False Negative
FNR	False Negative Rate
FP	False Positive
KNN	K-Nearest Neighbor
LR	Logistic Regression
ML	Machine Learning
MNB	Multinomial Naive Bayes
NB	Naive Bayes
NLP	Natural Language Processing
RF	Random Forest
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
TPR	True Positive Rate

1 Introduction

In the 21st century, dementia has become one of the biggest healthcare challenges due to the high demand for medical care, social care, and institutional care [46]. The term dementia refers to a variety of progressive syndromes that affect the brain, often impairing high-level cognitive functions such as memory, language, and thinking. It is also accompanied by a decline in everyday functioning and impairment in social interactions. There are an estimated 50 million people worldwide affected by dementia [48] [70]. Dementia support has thus become a key priority of national and international health policies. The COVID-19 pandemic has had a unique impact on people with dementia. As well as being at increased risk of contracting COVID-19, older adults with dementia are also more likely to have more severe symptoms and disease consequences than those without dementia. Almost two-thirds of all COVID-19 related deaths in Canada have been people with dementia. The majority of COVID-19 related deaths in care homes (49.5%) in the United Kingdom occurred in individuals living with dementia [4].

Due to the unprecedented situation and concerns regarding the impact of COVID-19 on people with dementia, the need to focus on people living with dementia has been heightened. As a result of COVID-19, universities have been unable to conduct timely or collaborative research, because recruitment and in-person studies have been suspended. In this case, modern technologies including social media can help health care researchers to analyze people's experiences and take necessary measures. Social media have rapidly shaped the social environment in recent years. The largest social media networks include Facebook, Instagram, Twitter, YouTube, and TikTok.

With the advent of Twitter in 2006, social networking has become popular, enabling its 330 million users to post comments and status updates. Tweets are often treated as an information source and cited in traditional news outlets [53]. However, despite the availability of this large dataset and some existing methodological tools for data scraping, the use of these data by health care researchers has tended to lag behind other disciplines — and this is especially true in social studies. One of the problems for the health care researchers is the unstructured nature of the data scraped from Twitter, from which extracting insights can be a challenging task. Accordingly, pre-processing methods should be applied on the data in order to analyze, understand, organize, and sort useful data.

Moreover, another problems is not only the huge amount of data to be processed but also the fact that not all of the data can be used for further analysis because they contain a lot of irrelevant information. Generally, the data need to be labeled manually. Manual labeling, which is a part of thematic analysis, is tedious, time-consuming, and daunting. Thus, improvements to existing methodologies are needed to enable

health care researchers to make better use of the data and understand the impacts of COVID-19 on people with dementia.

Nowadays, one of the modern and reasonable ways to perform a task (e.g., automatic labeling) is to use Machine Learning (ML) algorithms. *Supervised Learning* is one of the techniques in ML that is used for classification tasks based on labeled datasets. The goal of this study is to build an ML model that is able to help health care researchers and social scientists to detect irrelevant tweets without going through all the tweets manually.

1.1 Contributions of the thesis

In order to build an ML model that can help health care researchers to detect irrelevant tweets, we used three different datasets: (i) a dataset comprised of 5,058 tweets extracted from Twitter on COVID-19 and dementia from February 15 to September 7, 2020 to train, evaluate, and compare different models, (ii) a dataset comprised of 6,240 tweets from September 8, 2020 to December 8, 2021 to test the best model, and (iii) a dataset comprised of 1,289 tweets related to Canada's Alzheimer's Awareness Month from January 1 to January 31, 2022 to retrain and test the best model.

We used natural language processing in order to pre-process the tweets. To find the best ML model for irrelevant tweet detection, we trained eleven ML classifiers known as the Logistic Regression (LR) classifier, Naïve Bayes (NB) classifier, Multinomial Naïve Bayes (MNB) classifier, K-Nearest Neighbors (KNN) classifier, Support Vector Machine (SVM) classifier, Decision Tree (DT) classifier, Random Forest (RF) classifier, AdaBoost classifier, XGBoost classifier, BERT classifier, and ALBERT classifier using the first dataset and compared them in terms of their performance. Then, the best model from these classifiers is selected and applied to the second (completely unseen) dataset and reached the accuracy of 80% in classifying relevant and irrelevant tweets for people impacted by dementia and COVID-19.

Finally, to prove that the ALBERT model can be used for future studies in the context of people impacted by Alzheimer's disease/Dementia, we trained our model with 10% of the third dataset and tested it using 90% of the rest. Consequently, the model reached the accuracy of 88% in classifying relevant and irrelevant tweets, showing that the ALBERT model can be trained on a small sample of a labeled dataset and used to predict the rest of the unlabeled data efficiently. Thus, health care researchers can focus on the main goal of the study rather than manual labeling the data.

1.2 Outline

The thesis contains the research methods used and the results obtained when using the eleven ML models on three datasets to build an ML model that is able to detect irrelevant tweets efficiently. This thesis is organized as follows. Chapter 2 gives a summary of important aspects of this study such as thematic analysis, natural language processing, and ML techniques. Chapter 3 gives a description of the ML methods used. Chapter 4

gives a detailed description of the datasets used, the methodology, results, and discussion. Chapter 5 gives some conclusions and possible future research directions.

2 Literature Review

This section introduces concepts and background information associated with the main topics permeating this thesis. Section 2.1 provides information about thematic analysis. Section 2.2 describes the concept of natural language processing and its applications. Section 2.3 discusses machine learning as a whole and how it has been used for natural language processing. Section 2.4 provides general information about Ensemble Learning. Section 2.5 discusses transfer learning and some of the applications of BERT models.

2.1 Thematic Analysis

Thematic analysis is a method for identifying, analyzing, and reporting themes (patterns) within data. It is described as a descriptive method that organizes and describes datasets in detail. Thematic analysis is used commonly because of the wide variety of research questions and topics that can be addressed with this method of data analysis [12].

There are two primary ways to find the themes in thematic analysis: 1. *Inductive* 2. *Deductive* [10]. In the inductive or 'bottom up' approach, the process of coding the data is applied without trying to fit the data into a pre-existing coding frame. Therefore, the themes are linked to the data themselves. On the other hand, the deductive or 'theoretical' approach is usually driven by the researcher's theoretical or analytical interests. As a result, this form of thematic analysis tends to focus more on analyzing some aspect of the data rather than describing the entire data in detail.

2.2 Natural Language Processing

A language can be defined as a set of symbols controlled by a set of rules that are used for communication [9]. The term *natural* in natural language is used to distinguish human speech and writing from more formal languages like mathematical notations or programming languages that have a limited vocabularies and syntaxes. Natural Language Processing (NLP) can be defined as a set of computational techniques for analyzing and understanding human languages for a variety of tasks and applications [36]. Research in NLP can be classified into two broad categories: core areas and applications.

However, it is sometimes difficult to distinguish clearly to which areas a given issue belongs [54]. All these core areas solve the fundamental issues such as language modeling, which tries to find the relationships between naturally occurring words and helps to predict which word is more likely to appear next in the

sentence; morphological processing, which is the process of determining the morphemes used to construct a given word; syntactic processing or parsing, which checks the text for its logical meaning based on the rules of formal grammatical structure; and semantic processing, which attempts to understand meaning of signs, words, and phrases in the text. The application areas includes topics such as extraction of key information, translation of text between languages, summarization of written works, automatic questions-answering by inferring answers, and classification and clustering of documents [54].

2.3 Machine Learning

Learning as a generic process entails acquiring new behaviours, beliefs, information, abilities, or preferences, or adjusting current ones. The theories of personal learning, i.e., how humans learn, are defined by Behaviorism, Cognitivism, Constructivism, Experientialism, and Social Learning [2]. Machines rely on data for their learning. Machine learning (ML) is a type of artificial intelligence that enables computers to learn on their own and modify their actions to enhance their accuracy, with accuracy being defined as the number of times the chosen actions result in the right behaviors/decisions [2]. The term was coined by Arthur Samuel in 1959, who defined ML as a field of study that enables computers to learn without being explicitly programmed [63]. More recently, Tom Mitchell gave a definition that is more useful for scientific research: "A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E" [49].

ML is applied in a wide range of fields, e.g., automatic product recommendation [32], robotics, virtual personal assistants (like Google), computer games, pattern recognition, natural language processing, data mining, traffic prediction, market prediction, medical diagnosis, online fraud prediction, agriculture advisory, search engine result refining (e.g., Google search engine), bots (chatbots for online customer support), e-mail spam filtering, and crime prediction through video surveillance system [59].

ML contains multiple branches ranging from *supervised learning* and *unsupervised learning* to *ensemble learning* and *transfer learning*. Supervised learning consists of learning a function to map input data to known targets/labels based on a set of training data (often annotated by humans), whereas in unsupervised learning, algorithms try to find the structures in the input data without the help of any labels [2]. Supervised learning can be categorized into two types of problems: regression and classification.

In the regression problem, the models deal with continuous data, but in the classification problem, the data values are discrete or categorical [8]. Figure 2.1 shows an example of classification of objects with two features in supervised learning. Many algorithms attempt to find the best separating hyperplane by imposing different conditions with a same goal: reducing the number of misclassifications and increasing the noise-robustness. For example, consider the triangle point that is closest to the plane (its coordinates are about (5.1,3.0)). If the magnitude of the second feature were affected by noise and so the value was much smaller than 3.0, a slightly higher hyperplane could incorrectly classify it. Common classification algorithms

in supervised learning include: Decision Tree (DT), Support Vector Machine (SVM), K-nearest neighbors (KNN), Naive Bayes (NB). These algorithm have many application such as spam detection, cancer detection, sarcasm recognition in text data, etc.

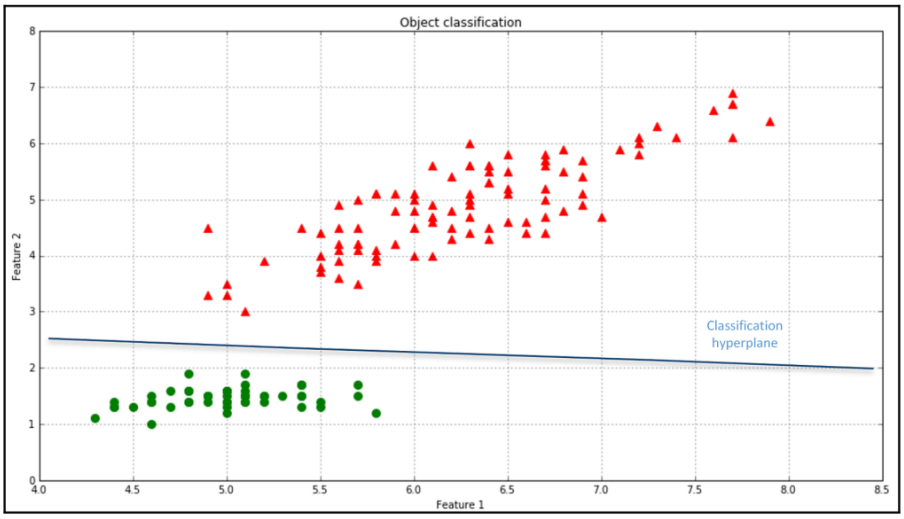


Figure 2.1: Example of linear classification in supervised learning [8]

One such example of using supervised algorithms is to detect depression in Twitter using content and activity features [1]. The activity features consist of the number of followers, number of following, total number of posts, time of posts, number of mentions, and number of retweets for each Twitter user that are categorized into four types (low, below average, average, and high), and they are defined using percentile values from the quartile distribution (Q1, Q2, Q3, and Q4). In [1], a dataset consisting of more than 1M tweets of 500 users is constructed. Tweets were sampled manually by two psychologists, and 334 of the Twitter users were labeled as depressed. For the pre-processing phase, tokenization and normalization (turning all the words into lowercase, removing punctuation, retweets, mentions, emojis, links, and stemming) are applied to the dataset. Then, the term frequency-inverse document frequency (TF-IDF) method is used to measure the words weights. The following three ML classifiers were evaluated using 10-fold cross-validation: NB, DT, and SVM using linear and radial kernel. The linear SVM algorithm proved to have the highest accuracy of 82.5% and recall of 0.85 for detecting depression in Twitter users.

In [19], five different ML classifiers are applied to categorize suicide-related tweets. The goal of this study was to compare the classification performance of four popular ML models with the Prism algorithm. The Prism algorithm is known to be simple and easy to understand, even though it is less popular than other machine learning algorithms. The Prism algorithm is based on the separate-and-conquer learning principle: a rule is learned that accurately predicts the value of the target attribute (the conquer stage) followed by removing the previously covered instances (the separate stage) and repeating the process until all instances are covered [19]. In this case, a dataset of 2,000 tweets was manually classified into seven

suicide-related categories by two human annotators. Two different approaches were carried out in this study: binary classification (Suicide and Flippant) and multi-classification (Suicide, Campaign, Flippant, Support, Memorial, Reports, and Other). For data pre-processing, the Bag of Words (BoW) technique was used to measure the words weights. The BoW is one of the popular techniques for feature extraction in data pre-processing that extracts the frequency of each word in the text documents known as the one-hot representation for classification [14]. Then, the following five ML classifiers were evaluated using 10-fold cross-validation: Prism, DT, NB, SVM, and Random Forest (RF). The results showed that the Prism algorithm had the highest F1-Score value of 0.85 for the binary classification and the RF algorithm reached the highest F1-Score of 0.69 for multi-classification.

A problem in Southeast Asia is methamphetamine addiction. In [14], two different techniques are used for data pre-processing to increase the accuracy in the classification of methamphetamine-related tweets. Although BoW is one of the popular techniques for feature extraction, it may result in a large feature dimension containing many null values resulting in sparse vectors in large documents; models thus face the challenge of extracting little information in such a large representational space [35]. The study aims to reduce the weakness of BoW by proposing a two-steps approach containing BoW and WordtoVec feature selection (BWF). The first step is to create the text representation, and the second step is to involve a domain-based feature selection. Data comprising 2,899 tweets are retrieved from Twitter using keywords related to methamphetamine consisting of the common name, slang name, and street name for Southeast Asia and are manually labeled by one expert into two classes: abuse or non-abuse. Then, stop words are removed, and stemming is used to reduce the number of features. Three different techniques, BoW, BWF, and TF-IDF, are used for feature extraction, and three classifiers including SVM, NB, and DT are evaluated on each approach using 10-fold cross-validation. The combination of BWF and DT classifier proved to have the highest accuracy and F1-Score of 81.5% and 0.818, respectively.

In [77], an automatic classifier is developed using SVM, NB, and RF algorithms to study public discourse and sentiment regarding older adults and COVID-19 on Twitter. The proposed system contains four stages: pre-processing, classification, sentiment analysis, and topic modeling. In the pre-processing stage, 8,453 random tweets (10.2%) from 82,893 scraped tweets are manually labeled by two researchers into four classes: informative, personal experiences, personal opinions, and jokes/ridicule. To identify ageist content, each tweet is rated by yes/no answers to three questions (referred to as attributes). Then, the text is converted to lowercase, and TF-IDF is used for feature extraction. Each of the ML algorithms is trained on each of the datasets, and the one that reached the highest accuracy of 79% is used to label the rest of the datasets. Latent Dirichlet Allocation (LDA) and the NRC Lexicon [77] are used to extract underlying themes and the dominant emotion for ease of interpretation of each tweet in different classes. The results show that 16.4% of the tweets in the dataset have ageist content, with most of them hinting at the element of senicide.

Sarcasm is the art of expressing an opposite suggestion from what is literally being suggested by the words in the context. It is commonly used on social media such as Twitter, Facebook, Instagram, and others.

Sarcasm is rich with features related to various expressional values that are widely used for the classification using different algorithms [66]. In [67], the detection of sarcasm on Twitter is examined using four different ML classifiers. In this case, a dataset containing 1,984 tweets that were manually labeled into two classes of 0 or 1 is pre-processed by removing stop words and symbols, turning to lowercase, lemmatization, tokenization, and vectorization using the TF-IDF method. Four ML classifiers including SVM, DT, Multinomial Naive Bayes (MNB), and Logistic Regression (LR) are trained using 70% of the data and are tested using the rest of the data (30%). The results present that the LR algorithm reached the highest accuracy of 97%, the precision of 0.97, recall of 0.97, and F1-Score of 0.97, and the SVM algorithm has the least accuracy of 42% in sarcasm detection of the tweets.

Bots, short for robots and also called internet bots, are one of the social media features that are mainly categorized into two groups: good and bad. Good bots are automatic programs that simulate human behavior and control the account without human intervention. For example, good bots act as players in multi-player online games to make them more entertaining. However, bad bots are malicious bots that assist hackers in breaching data. Bots appear to behave like humans; thus it is difficult to spot bad bots. In a study done by [57], four different ML classifiers are used to detect bots on Twitter. The behavior of non-bots (real social media users) and bots differs in that bots have far more followers on their profiles than friends, whereas non-bots have an equitable distribution of friends and followers [57]. In this case, users that have more than 10,000 followers or have more than 16,000 tweets are considered bots. Binary-class datasets (bots/non-bots) comprising 3,368 tweets containing several attributes such as URL, description, number of friends, number of followers, a screen name (used to communicate online), location, id, verified (if the user is authenticated), favorite (used for liked tweets), and the listed count are used. The closest values for the features nearing true positives are checked for all of the features using a Spearman correlation confusion matrix, and it was demonstrated that there is a strong correlation between verified, listed-count, friends-count, followers-count, and the target label. Then a new method named *bag of bots words* is proposed that adds a new column called *status-binary* consisting of words commonly used by bots to the data. Four ML classifiers including DT, KNN, LR, and NB are trained using all the features with and without using the *bag of bots words* method on training data containing 80% of the datasets and then tested on the rest of the data (20%). The proposed method showed that using *bag of bots words* and DT classifier has the highest test accuracy of 99.2% compared to other classifiers.

2.4 Ensemble Learning

Ensemble learning is usually referred to as the machine learning interpretation for the wisdom of the crowd, where multiple learning algorithms (individual models) are trained to solve a problem to improve classification and testing performance [52]. Ensemble methods can be categorized into two main frameworks: the independent framework and the dependent framework. In the independent framework, each model is built

independently from other models. There are multiple techniques based on an independent framework including Max Voting, Averaging, Weighted Averaging, Stacking, Bagging (e.g., RF algorithm), and Blending [62]. Figure 2.2 shows the basic idea of an ensemble classification model that generates classification results using multiple models and then, integrates multiple results into a consistency function to get the final result with voting schemes [23].

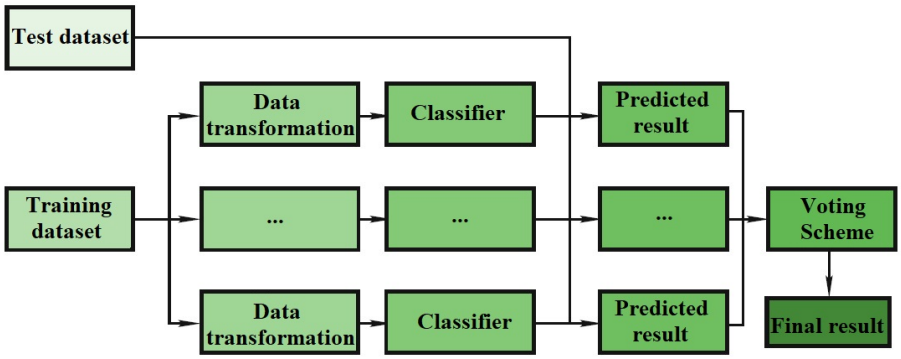


Figure 2.2: Example of independent framework in ensemble learning [23]

In the dependent framework, the output of each model affects the construction of the next model and the knowledge gained in previous iterations guides the learning in the next iteration. One such popular method of the dependent framework is Boosting comprises different algorithms such as Adaptive Boosting (AdaBoost), Gradient Boosting (GBM), and Extreme Gradient Boosting (XGBoost) [17]. Figure 2.3 shows the general framework of the Boosting method. Multiple sequential models are created, each correcting the errors from the previous model by assigning weights to the outputs/results that are incorrectly classified.

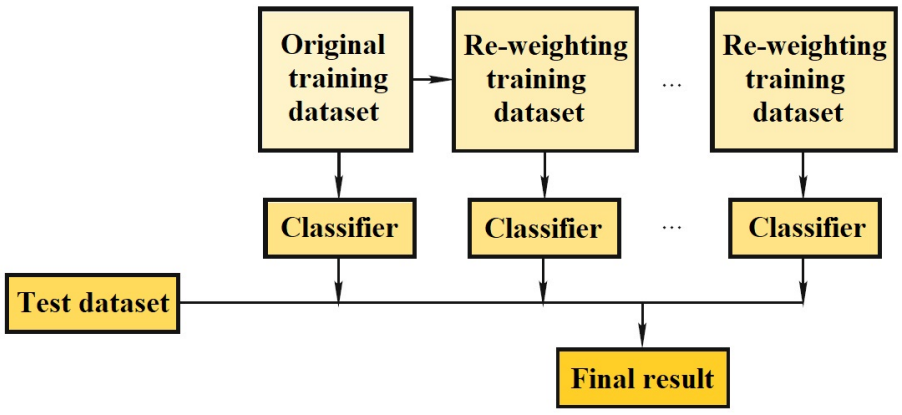


Figure 2.3: General framework of Boosting in ensemble learning [23]

Ensemble learning is extremely extensible to use for different types of tasks such as detecting fake news. In [52], the Synthetic Minority Over-Sampling Technique (SMOTE) and the classifier vote ensemble are proposed

to detect COVID-19 infodemic tweets on Twitter and protect the public from inaccurate and information overload. In this case, four different datasets are collected for Pre-lockdown, Post-lockdown, concatenation of pre/post-lockdown, and the subset of pre/post-lockdown with a total number of 176,877 tweets. Then a data pre-processing comprising stop word removing, stemming, and tokenization is applied to the data. These four datasets contain imbalanced classes. Therefore, in order to solve the problem of imbalanced data and bias factors, the SMOTE algorithm tries to resample datasets against challenges posed by imbalanced output class by oversampling instances with smaller class representation through the creation of fresh synthetic instances [52]. Then, the BoW and TF-IDF techniques are used for the feature extraction. Finally, five different base algorithms including NB, function-sequential minimal optimization, voted perceptron, KNN, and RF are used to form the ensemble classifier. The results show that both KNN and RF reached the highest accuracy of 99.66% compared to other algorithms.

In [53], the Bagging technique is used to detect Alzheimer's Disease (AD) stigma on Twitter. A total of 31,150 tweets are collected using nine AD-related keywords. Then, 311 random tweets are coded manually by two researchers according to 6 dimensions (metaphorical, personal experience, informative, joke, ridicule, and individual/organization) and used to train the Bagging algorithm and automatically code the remaining tweets. In order to examine the parameter space of the algorithm, the grid search method is applied using 3-fold cross-validation. The results are as follows: (1) the manual coding shows that 43.41% of all tweets are *informative*, 23.79% are *joke*, 21.22% are *metaphorical*, 19.29% are *organization*, 18.33% are *personal experience*, and 24.50% are *ridicule*, (2) the accuracy of the algorithm ranged from 95.15% (*informative*) to 86.38% (*organization*), and (3) the automated coding shows that 21.13% of all tweets used AD-related keywords in a stigmatizing fashion.

In [18], the classification of suicide-related tweets is examined using five different ML classifiers. The goal of this study was to compare the classification performance of four popular ML models with ensemble learning on binary-class, three-class, and seven-class datasets [18]. In this case, a dataset of 2,000 tweets is manually classified into seven suicide-related categories (Suicide, Campaign, Flippant, Support, Memorial, Reports, and Other) by four human annotators. For data pre-processing, stop words, punctuation, URLs, and non-ASCII characters are removed, and Part-of-speech tagging and stemming are applied. The BoW technique is used to extract features. Then, the following four ML classifiers are evaluated separately using 10-fold cross-validation: DT, NB, SVM, and RF. For ensemble learning, majority voting is applied using two different approaches: (1) a combination of DT, NB, and SVM algorithms, and (2) a combination of DT and SVM algorithms. The results show that for individual classifiers on the binary-class dataset, the achieved F1-Score is in the range of 0.411 to 0.776 using different ML classifiers, with the SVM having the highest F1-Score of 0.80 for the suicide class. Moreover, the SVM algorithm also has the best performance on the three-class and seven-class datasets. Finally, the results of the ensemble learning show that combining the DT, NB, and SVM algorithms achieved a higher performance on all three datasets than combining DT and SVM algorithms.

2.5 Transfer Learning: BERT Model

As part of natural language processing, text representation is a key element of converting natural language inputs into machine-readable data that can also be considered as a computer encoding of the text. The use of representations that reflect the internal content and conceptual structure of the text is suitable for machine learning problems [65]. Currently, the most advanced text models use transformers in Transfer Learning that deal with a text representation. Transfer Learning is defined as reuse of a pre-trained model as the starting point for a model on a new task to leverage the retrieved knowledge [83]. Transformers are a type of neural network in transfer learning that deal with sequence transduction problems, that is, such problems that both the input and output information is a sequence [22].

In late 2018, scientists from the Google AI Language laboratory proposed a new linguistic model called the Bidirectional Encoder Representations from Transformers (BERT). BERT is a deep bidirectional transformer architecture that supports multilingual universal language representation of 104 languages [22]. Two main sources that helped to pre-train the BERT model to obtain contextual embeddings are as follows: (1) unlabeled Wikipedia (2,500M words), and (2) Book corpus (800M words). BERT uses what is called a *WordPiece* tokenizer that splits words either into the full forms (i.e., one word becomes one token) or into word pieces (i.e., one word can be broken into multiple tokens). Figure 2.4 represents the architecture of the BERT model, comprising the BERT tokenizer, embedding layers, attention layers, fully connected, and softmax layer (all the layers are fully described in Chapter 3).

BERT consists of two steps: pre-training and fine-tuning. In pre-training, an unlabeled set of data is used to train the model across different pre-training tasks. Fine-tuning involves initializing the BERT model with pre-trained parameters and then fine-tuning them all using labeled data [79]. So far, based on different NLP tasks, multiple variants of the BERT model are proposed including ALBERT, DeBERTa, RoBERTa, ELECTRA, BERTSUM, and BART [65].

The BERT model has a lot of applications in text analysis including sentiment classification. The traditional sentiment algorithms proved to have multiple limitations such as complex feature engineering and the requirement of massive linguistic resources that are time-consuming and error-prone [27]. In [27], three variants of the target-dependent BERT model are proposed to overcome the issues with the traditional sentiment algorithms and examine whether the context-aware representation of BERT model can achieve similar performance improvement in aspect-based sentiment analysis. A target-dependent sentiment classification task predicts the sentiment polarity of a tuple consisting of a sentence and a target with the aim of determining the sentiment polarity of sentence towards the target [27]. For example, the sentence "great food but the service was dreadful" is positive for "food" and negative for "service". The three proposed target-dependent variants of BERT are as follows: TD-BERT, TD-BERT-QA-MUL, and TD-BERT-QA-CON. In TD-BERT, a max-pooling layer is used before data are fed to the fully connected layer to get output from the target items. The TD-BERT-QA-MUL and TD-BERT-QA-CON variants use element-wise multiplication

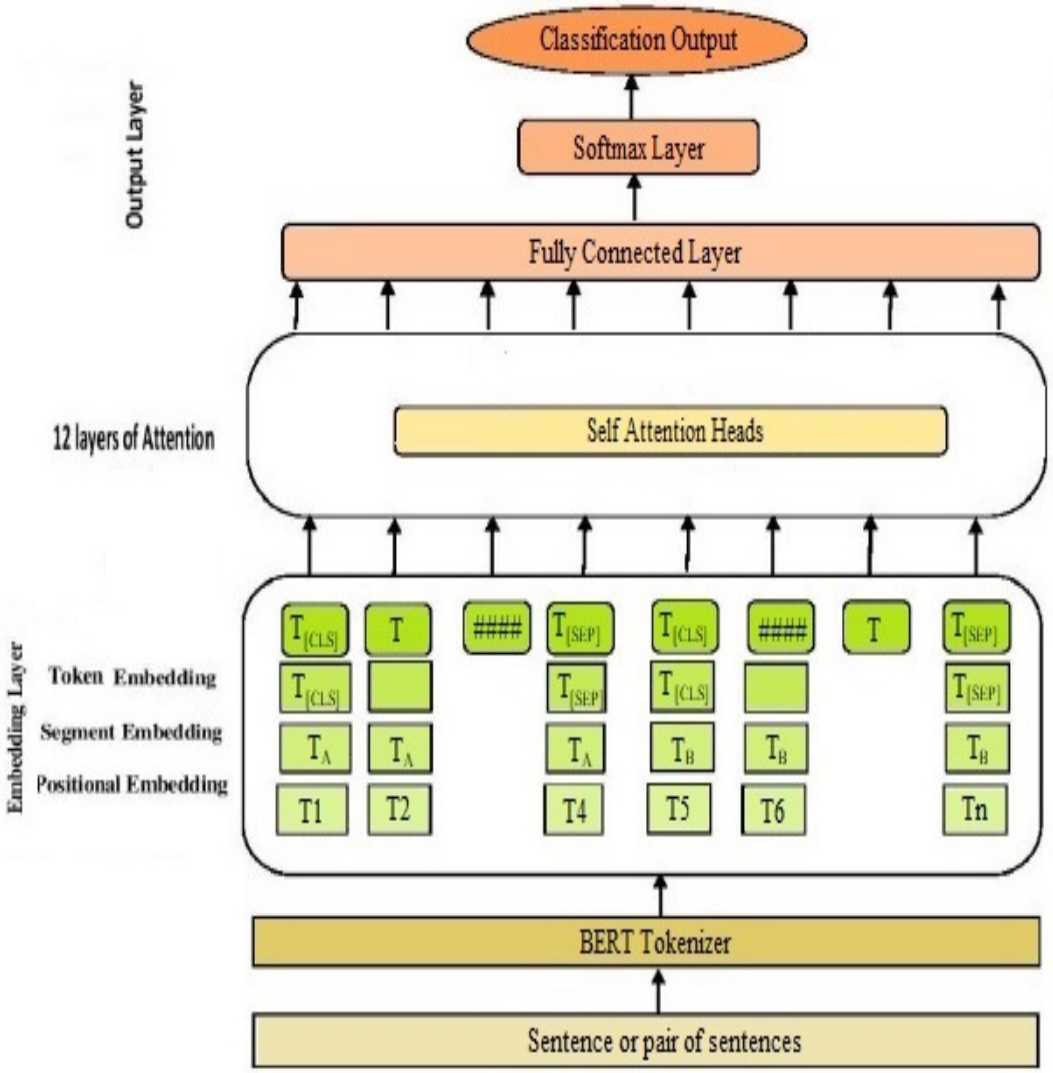


Figure 2.4: Architecture of the BERT model [22]

and concatenation for features extracted from the TD-BERT model. Three datasets of reviews in different domains such as laptops, restaurants, and Twitter (celebrities, products, and companies) having four labels of positive, negative, neutral, and conflict are collected and split into training and testing data. According to the experiment, the incorporation of target information proved to be a key factor in BERT's performance improvement. Moreover, the TD-BERT reached the highest accuracy of 78.87% on laptop data, TD-BERT-QA-MUL achieved the highest accuracy of 85.27% on restaurants data, and finally, TD-BERT-QA-CON had the highest accuracy of 77.31% on Twitter data.

In [69], the RoBERTa model is used to improve the detection of rumors on Twitter. In this study, a rumor on social media is defined as "information that is presented in a manner that attracts the reader's attention and incites them to share it, although its content is unverified and its true value can be questioned" [69]. Four different datasets of previous related works containing two labels of rumor/non-rumor are used. For data pre-processing, stop words, hashtags, links, and "four-letter words" are removed, and emojis are replaced with their corresponding words. Moreover, to remedy the issue of imbalanced classes in the datasets, the data augmentation method is used, that is, a technique to generate new samples based on the datasets that vary from the original ones. Both, the BERT and RoBERTa models are fine-tuned using all the datasets in five epochs to provide rumor-sensitive word embedding/representation for tweets. These representations are then fed to the three ML classifiers (RF, SVM, and DT) to evaluate the proposed approach using 5-fold cross-validation. The results show that RoBERTa outperformed the BERT model on unseen datasets with an average of a 3% increase in F1-score. Moreover, to evaluate the impact of data augmentation, the performance of RoBERTa is examined using augmented and non-augmented datasets, and the result shows that on average the performance of RoBERTa was increased by 3% in terms of F1-score when using the augmented dataset.

3 Methodology

This chapter discusses the data pre-processing and feature extraction techniques in NLP, ML, and transfer learning models that are applied to the data of this study to give a better understanding of how they are built and used. The metrics for evaluating the models are also described.

3.1 Data Pre-processing in NLP

An important part of NLP and its applications is data pre-processing, which is the first step in the text mining process. Data pre-processing is a method that converts the raw data containing noise into useful data for analysis. By data pre-processing, the quality of the data and the performance of the classifiers increase [47]. In general, some of the pre-processing steps for online datasets such as Twitter are removing emojis, punctuation, and possessive pronouns, filtering out non-English language tweets, @tweets (replies), duplicate tweets, and tweets with a permalink, and turning to lowercase [24][47]. In this chapter, we discuss the other key steps of data pre-processing in NLP.

3.1.1 Word Tokenization

Tokenization is the process of breaking a phrase, sentence, paragraph, or even an entire text document into individual words, characters, or subwords (parts of words) called tokens that help in understanding the context or developing the model for the NLP [75]. The most common way of forming tokens is based on white spaces [16]. Assuming space as a delimiter in the following sentence: “This is a cat.” The tokenization of the sentence results in 4 tokens: "This", "is", "a", and "cat". This is an example of word tokenization where each token is a word.

3.1.2 Stop Word Removal

Articles, prepositions, pronouns, etc., are the most commonly used words in text documents without any helpful information. Such words are called stop words [74]. Stop words are parts of natural language that should be eliminated from a text. They make the text look heavier, less important, and have more noise for analysts [16] [74]. Examples of stop words are: the, in, a, an, with, etc. Removing these commonly used words from the text can help the models focus on the important words instead [24]. Figure 3.1 shows an example of tokenization followed by stop word removal.

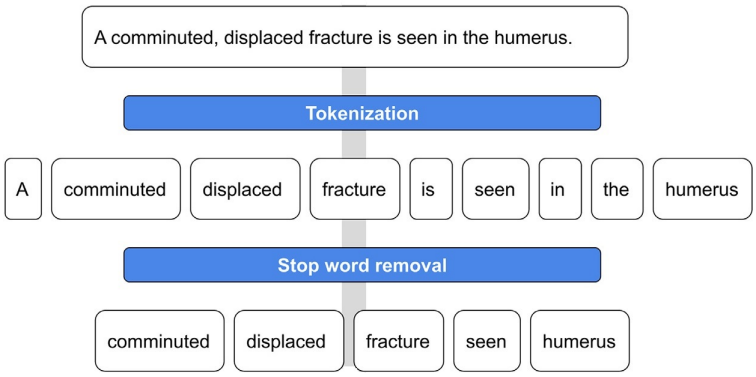


Figure 3.1: A sample process of tokenization and stop words removal [16]

3.1.3 Lemmatization

Lemmatization is a technique to remove inflectional parts in a word or convert the word into its base form [67]. The purpose of lemmatization is to reduce surface words to their canonical form; this latter is the base or familiar dictionary form of the word, which is known as the lemma [81]. The lemma relates different word forms that have the same meaning, for instance, the lemma of *best* and *better* is *good*. Lemmatization is found to be efficient, in particular, for feature extraction and information retrieval that depend on the frequency of the words to create better models [16].

3.2 NLP Feature Extraction: Vectorization

Vectorization in NLP is a technique to convert textual data into list of numbers/vectors to create machine-readable data for performing NLP tasks [68] [20]. To be more precise, vectorization applies a statistical measure to assign a weight to each token in textual data and produces feature representations. The features are ranked by weight. Features whose weights are greater than a predefined threshold are selected to be kept for classification, and the rest are removed from the feature space [2][13]. Examples of some feature extraction methods include the BoW, document frequency, *Term Frequency-Inverse Document Frequency*, known as TF-IDF, WordtoVec, GloVe, etc. [68].

3.2.1 TF-IDF Vectorization

TF-IDF is the most commonly used method for converting text documents into vectors, known as features for text classification or other NLP tasks [82][68]. The TF-IDF calculates the relative frequency of words in a specific document compared to the inverse proportion of that word over the entire text document. This calculation determines the importance of a given word in a document. Accordingly, words frequently used in a single document or a small set of documents have higher TF-IDF numbers/weights and are regarded as more

representative and kept. Moreover, words common to many documents, such as articles and prepositions have lower TF-IDF weights and are considered as less representative and disregarded [82] [58]. Given a document collection D, a word t, and an individual document $d \in D$, the TF-IDF can be defined as:

$$\begin{aligned} \omega_{t,d} &= TF \times IDF \\ &= tf_{t,d} \times \log\left(\frac{D}{df_t}\right), \end{aligned} \tag{3.1}$$

where $tf_{t,d}$ is the frequency of word t in the document d, D is the total number of documents in the collection, and df_t is the number of documents where word t occurs in [82]. For example, considering the below two text documents, the TF-IDF for the words "*it*" and "*lion*" are as follows:

Table 3.1: Document 1 (D1) [71]

Term	Count
lion	2
it	3
forest	1
man	1

Table 3.2: Document 2 (D2) [71]

Term	Count
cat	1
it	3
live	1
house	1

TF-IDF for the word "*lion*":

TF ("*lion*", D1) = 2/7 = 0.29

IDF("*lion*", D) = log(2/1) = 0.3

TF-IDF ("*lion*", D1) = TF("*lion*", D1) × IDF("*lion*", D) = 0.29 × 0.3 = 0.087

TF-IDF for the word "*it*":

TF ("*it*", D1) = 3/7 = 0.43

TF("*it*", D2) = 3/6 = 0.5

IDF("*it*", D) = log(2/2) = 0

TF-IDF ("*it*", D1) = TF("*it*", D1) × IDF("*it*", D) = 0.43 × 0 = 0

TF-IDF ("*it*", D2) = TF("*it*", D1) × IDF("*it*", D) = 0.5 × 0 = 0

The calculations show that the TF-IDF for the word "*lion*" is 0.087 and can be used as a feature. However, the TF-IDF for the word "*it*" is 0, which implies that the word "*it*" is not so influential in the whole document and can be ignored for further processes.

3.3 Traditional Machine Learning Classifier

Machine learning involves assigning specific tasks to a computer program. Generally, a machine is considered to learn from its experience if its measurable performance on these tasks improves as it gains more and more experience. Thus, based on data, the machine makes predictions/forecasts. The three main types of machine learning problems are classification, regression, and clustering [59]. To apply the appropriate machine learning algorithm, one can choose from the multiple branches of supervised learning, unsupervised learning, ensemble learning, and transfer learning based on the availability of types and labels of training data. Because this study aims at predicting discrete values with labeled datasets, it only focuses on the classification in the three main branches of supervised learning, ensemble learning, and transfer learning. In the next few sections, all of the algorithms used are discussed in detail.

3.3.1 Logistic Regression Classifier

Logistic Regression (LR) is a machine learning algorithm within the supervised-learning approach for classification. In LR, probabilistic concepts are used to analyze and classify data. The hypothesis for logistic regression is to limit the output of the algorithm between 0 and 1 [67]. The equation for the logistic regression can be defined as:

$$P(Z) = \frac{\exp(Z)}{1 + \exp(Z)}, \tag{3.2}$$

where P is the probability of a binary outcome, and $Z = \alpha + \beta X$, with X the weight of a text document; α is a constant value that determines the intersection of the line on the X-axis with β being its slope. Figure 3.2 shows the sigmoid curve traced by the logistic function. P behaves like the distribution function of a symmetrical density, with midpoint zero and rises monotonically between 0 and 1 as Z moves on the real number axis [21] [7].

3.3.2 Naïve Bayes Classifier

Naïve Bayes (NB) is a probabilistic classifier that is based on the Bayes rule. It uses the assumptions of strong independence between variables to construct a simple and fast algorithm [30]. The Bayes rule for two

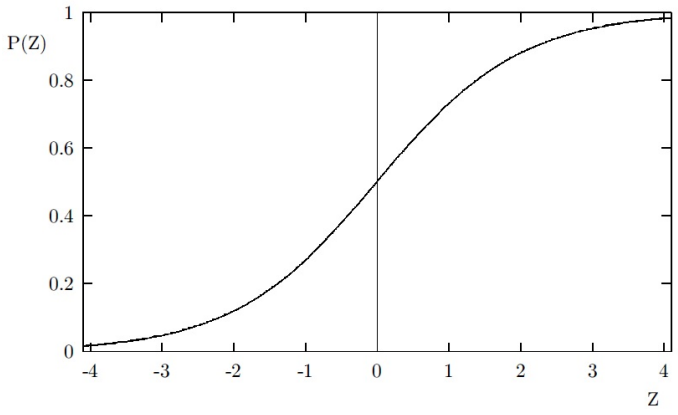


Figure 3.2: The logistic curve $P(Z)$ [21]

events of A and B is defined as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{3.3}$$

where $P(A|B)$ is the posterior probability, $P(B|A)$ is the likelihood, $P(A)$ is the prior, and $P(B)$ is the evidence. A major advantage of NB is its ability to efficiently combine evidence from various features [1]. Consider a set of training instances where each instance in the set is specified by a set of attribute values/weights $[x_1, x_2, ..., x_n]$ and a target/label. Let Y be a set of categories $[y_1, y_2, ..., y_j]$ defining the target function. Based on the attribute values for a text instances X, NB assigns the text instances to the category that scores the highest probability [13]. The probability that the text instances X belongs to a specific category y_j can be estimated as follows:

$$P(y_j|X) = \frac{P(X|y_j)P(y_j)}{P(X)}, \tag{3.4}$$

where $P(y_j|X)$ is the posterior probability of class y_j given a set of text instances X. On the basis of the Bayesian hypothesis that features are conditionally independent, the probability of category y_j can be reformed as

$$P(y_j|X) = P(y_j)\prod_{i=1}^{n} P(x_i|y_j), \tag{3.5}$$

where n indicates the number of features x_i that construct the training instances. The category of the given test instance X is found by NB classifier as

$$V_{NB} = \underset{y_j \in Y}{\operatorname{argmax}} P(y_j)\prod_{i=1}^{n} P(x_i|y_j), \tag{3.6}$$

where V_{NB} is the output of NB model and gives the category of the given test instance.

3.3.3 Multinomial Naïve Bayes Classifier

In order to solve text classification problems, a large number of naïve Bayes text classifiers are proposed, of which multinomial naïve Bayes (MNB) is widely used due to its simplicity, efficiency, and effectiveness

[33][38]. Given a test document X, represented by a word vector $x_1, x_2, ..., x_n$, MNB uses the equation 3.7 to classify the document X.

$$y(X) = \underset{y \in Y}{\text{argmax}} P(y) \prod_{i=1}^{n} P(x_i|y)^{f_i}, \tag{3.7}$$

where $y(X)$ is the class label predicted by MNB, Y is the set of all possible class labels y, n is the vocabulary size in the text collection (the number of different words in all of the documents), $x_i (i = 1, 2, ..., n)$ is word i that occurs in the document X, f_i is the frequency count of the word x_i in the document X, $P(y)$ is the probability that the document X occurs in the class y, and $P(x_i|y)$ is the conditional probability that the word x_i occurs given the class y, which can be calculated by the following equations:

$$P(y) = \frac{\sum_{j=1}^{n} \delta(y_j, y) + 1}{n + s}, \tag{3.8}$$

$$P(x_i|y) = \frac{\sum_{j=1}^{n} f_{ji} \delta(y_j, y) + 1}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ji} \delta(y_j, y) + m}, \tag{3.9}$$

where n is the number of training documents, s is the number of classes, y_j is the class label of the training document j, f_{ji} is the frequency count of the word x_i in the training document j, and $\delta(y_j, y)$, known as *Kronecker delta*, is a binary function,

$$\delta(y_j, y) = \begin{cases} 1 & \text{if } y_j = y, \\ 0 & \text{if } y_j \neq y, \end{cases} \tag{3.10}$$

which is one if its two classes are identical and zero otherwise.

3.3.4 K-Nearest Neighbors Classifier

The K-Nearest Neighbors (KNN) is a non-parametric classification algorithm, known for its simplicity and effectiveness. KNN classifies data based on closest or neighboring training examples in a given region [72]. Figure 3.3 shows a simple KNN structure. For a new input (blue point), KNN performs two operations. First, it analyzes the K points (nearest neighbors) closest to the new data input. Second, using the neighbor classes, KNN determines to which class the new data belongs.

Hence, the distances need to be calculated between the test sample and the specified training samples. KNN has three different approaches to calculate the distances: Euclidean, Manhattan, and Hamming. Euclidean distance and Manhattan distance are more common when the data are numerical with Euclidean distance being more popular. Hamming distance is used to measure the distance between categorical variables. The KNN algorithm assumes that it is possible to describe documents in a Euclidean space as points [76]. The distance between two points in the plane with coordinates $\mathbf{p}_1 = (x_1, y_1)$ and $\mathbf{p}_2 = (x_1, y_1)$ can be calculated by the following equation:

$$d(\mathbf{p}_1, \mathbf{p}_2) = d(\mathbf{p}_2, \mathbf{p}_1) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3.11}$$

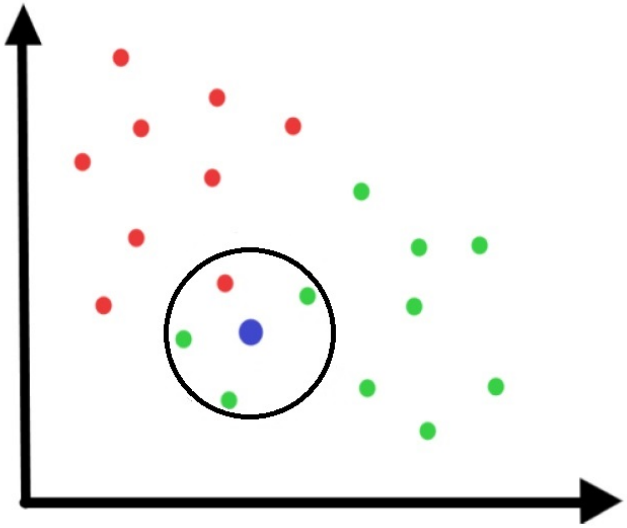


Figure 3.3: A simple KNN structure [72]

Figure 3.4 shows documents in Euclidean space for $K = 1$ (i.e., considering one nearest neighbor) and $K = 3$ (i.e., considering three nearest neighbors). The Euclidean space for $K = 1$ shows that the blue point can be similar to red square. However, the Euclidean space for $K = 3$ represents the blue point as being a green triangle.

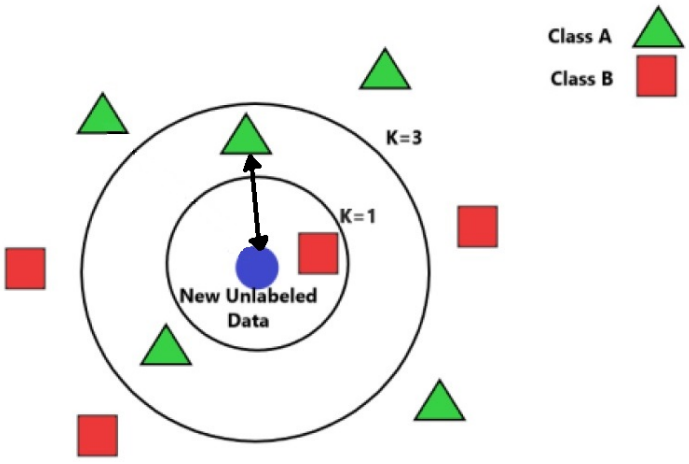


Figure 3.4: Euclidean space for $K = 1$ and $K = 3$ [72]

3.3.5 Support Vector Machine Classifier

Another ML classifier is the Support Vector Machine (SVM) with the idea of finding a hyperplane that best divides the data for a binary class problem. A hyperplane is a linear polynomial with n variables that separates and classifies data sets from the two classes, where $n > 1$ [1]. James et al. (2013) describe the SVM as a natural approach for classifying linearly separable datasets in finite-dimensional spaces [31]. However, some datasets are not linearly separable. In this case, another approach, known as *kernel trick*, is applied to the SVM [29]. Using the kernel trick approach, the SVM maps the vectors (data points) into a higher-dimensional space, then uses a linear classifier within the new space.

The best hyperplane is the one that creates the maximum margin from both classes. The margin is the distance between the hyperplane and the nearest vectors from either class. The nearest vectors are known as *support vectors*. The support vectors are the only vectors such that their movement directly affects the maximal margin hyperplane. The movement of other vectors has no effect on the maximal margin hyperplane.

Figure 3.5 illustrates an SVM applied to the dataset with two classes. The optimal hyperplane and the margins are visually described in the figure with solid and dashed lines, respectively. Vectors on the margins are the support vectors.

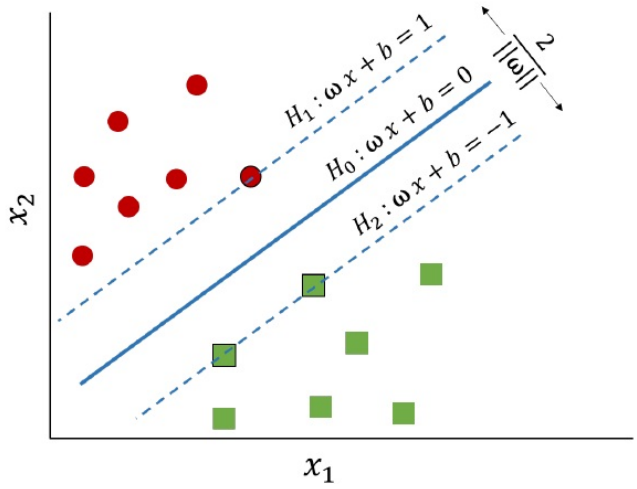


Figure 3.5: Optimal hyperplane with maximum margin for an SVM trained with a two-class dataset [1]

Here are the steps to find the optimal hyperplane for a linearly separable dataset as shown in Figure 3.5:

1. Define the hyperplane H_0 such that

$$H_0 : \boldsymbol{\omega}^T \mathbf{x} + b = 0, \tag{3.12}$$

where $\boldsymbol{\omega}$ is the vector of weights for each feature and $\mathbf{x}$ is the input vector.

2. By considering one class labeled as positive and the other class labeled as negative, two parallel

hyperplanes H_1 and H_2 that define the margins are described as

$$\begin{aligned} H_1 &: \boldsymbol{\omega}^T \mathbf{x} + b = 1, \\ H_2 &: \boldsymbol{\omega}^T \mathbf{x} + b = -1, \end{aligned} \tag{3.13}$$

where any vector on or above H_1 is related to the positive class and any vector on or below H_2 is related to the negative class. The vectors can be represented as

$$\begin{aligned} \boldsymbol{\omega}^T \mathbf{x}_i + b \geq +1 \quad \text{when} \quad y_i = +1, \\ \boldsymbol{\omega}^T \mathbf{x}_i + b \leq -1 \quad \text{when} \quad y_i = -1. \end{aligned} \tag{3.14}$$

where $\mathbf{x}_i$ is the input vector i in the document and y_i is the corresponding class label. Equations (3.14) can be combined into one equation as:

$$y_i(\boldsymbol{\omega}^T \mathbf{x}_i + b) \geq 1 \tag{3.15}$$

3. By recalling the distance between a point(x_0, y_0) and a line $Ax + By + C = 0$,

$$\frac{|Ax_0 + By_0 + C|}{\sqrt{(A^2 + B^2)}}, \tag{3.16}$$

the distance between H_1 and H_0 leads to

$$\frac{|\boldsymbol{\omega}^T \mathbf{x} + b|}{||\boldsymbol{\omega}||} = \frac{1}{||\boldsymbol{\omega}||}. \tag{3.17}$$

As a result, the margin, which is the total distance between H_1 and H_2, can be represented as

$$\frac{2}{||\boldsymbol{\omega}||} \tag{3.18}$$

In order to maximize the margin, $||\boldsymbol{\omega}||$ should be minimized.

3.3.6 Decision Tree Classifier

The Decision Tree (DT) Classifier is designed to solve classification problems by learning a hierarchy of if/else questions and answers and creating a tree representation that results in a decision [50]. The goal of the DT classifier is to get the right classification result by asking the least number of if/else questions. These series of questions can be illustrated as a decision tree.

In the first step, the algorithm starts with the whole training set and chooses the most informative feature about the target as the *root* node based on using different criteria. The root node is located on the top of the DT [15]. Each internal node contains either a question, which is called *test*, or a classification result (decision taken after computing all features), which is called *leaf* [59]. Moreover, the answer to a test is connected to the next test through *edges*. In the second step, the algorithm splits the set into internal nodes (sub-nodes) with the same feature value based on the test in the root. For each node, the two steps are repeated until all the edges lead to a leaf [56]. The length of the longest path from the root to a leaf is called the *depth*.

Table 3.3 shows a small data set, known as *Saturday morning*, for the weather condition and the possibility to do some activities that are not planned in advance. Each object's value of each attribute is shown together

with the class of the object (here, class P means the weather condition is suitable for activities that are not planned in advance and class N means the weather condition is not suitable.)[55].

Table 3.3: A small training set [55]

No.	Attributes				Class
	Outlook	Temperature	Humidity	Windy	
1	sunny	hot	high	false	N
2	sunny	hot	high	true	N
3	overcast	hot	high	false	P
4	rain	mild	high	false	P
5	rain	cool	normal	false	P
6	rain	cool	normal	true	N
7	overcast	cool	normal	true	P
8	sunny	mild	high	false	N
9	sunny	cool	normal	false	P
10	rain	mild	normal	false	P
11	sunny	mild	normal	true	P
12	overcast	mild	high	true	P
13	overcast	hot	normal	false	P
14	rain	mild	high	true	N

A simple DT that classifies each object in the training set is given in Figure 3.6. Also, a complex DT for the data in Table 3.3 is shown in Figure 3.7.

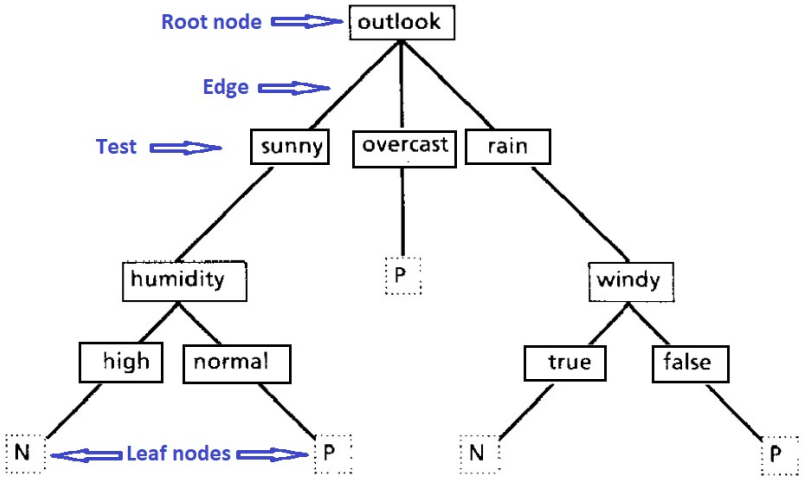


Figure 3.6: A simple DT[55]

Essentially, a suitable DT is constructed in a way that can be able to correctly classify not only objects from the training set but also unseen objects. Therefore, a DT must be able to capture some meaningful relationship between an object's class and the value of each of its attributes. Considering two decision trees, each of which is correct over the training set, it seems wise to select the simpler one. This is because the

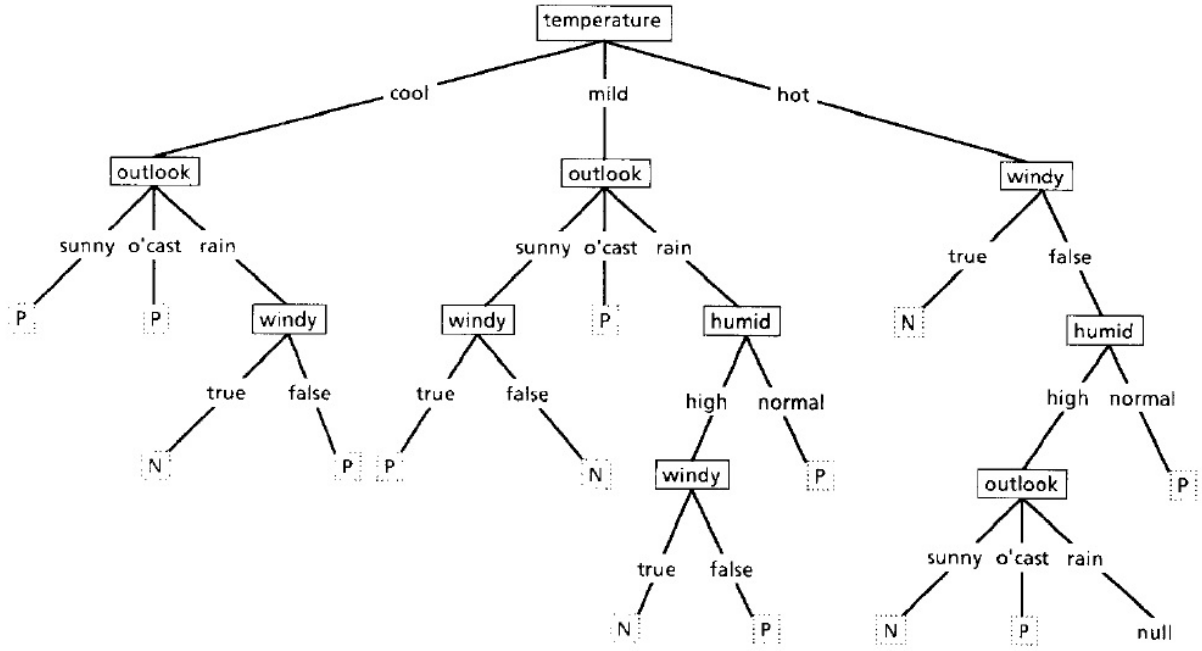


Figure 3.7: A complex DT[55]

simpler DT is more likely to capture the structure of the problem and would therefore be expected to correctly classify more unseen objects with less overfitting (overfitting will be discussed in the following sections) [55].

Depending on which feature is the most informative, the DT can use various criteria to decide how to split the data and order features. The two most popular criteria of DT are Entropy and Gini index. These two criteria measure how much information a feature provides about a class [56].

3.3.6.1 Splitting Criteria in DT: Entropy

Entropy is a common way of measuring the degree of impurity in a set of features [40]. For a set of features, X, the entropy calculation for a given feature, x_i, is

$$S(x_i) = -\sum_{j=1}^{N} p_{ij} \log p_{ij}, \tag{3.19}$$

where N is the number of classes in feature x_i, and p_{ij} is the proportion of instances belonging to class j considering feature x_i.

For a binary classification problem, if all examples from the data are from only one class, the entropy yields 0. If half of the records are of one class and half are of the other class, then entropy yields 1.

3.3.6.2 Splitting Criteria in DT: Gini Index

The Gini index is a metric to measure how often a randomly chosen element would be incorrectly identified [40] [15]. This means a feature with lower Gini index should be preferred. For a set of features, X, the Gini index for a given feature, x_i, is

$$\begin{aligned} G(x_i) &= \sum_{j=1}^{N} p_{ij}(1 - p_{ij}) \\ &= 1 - \sum_{j=1}^{N} {p_{ij}}^2, \end{aligned} \tag{3.20}$$

where N is the number of classes in feature x_i and p_{ij} is the proportion of instances of feature x_i that belong to class j.

3.4 Ensemble Learning Classifier

Ensemble classifiers combine multiple base classifiers to increase the accuracy of the final classification. Each base classifier can be any kind of supervised classification such as DT, neural networks, or SVMs. Using DT as a base classifier is more common than other ML classifiers [15]. Although a single DT can be an excellent classifier, increased accuracy often can be achieved by combining the results of a collection of DTs. Ensembles of DTs are sometimes among the best-performing types of classifiers [40]. The two main categories of ensemble learning are *Bagging* and *Boosting*. The next three sections discuss three different variants of the Bagging and Boosting methods in detail.

3.4.1 Bagging: Random Forests Classifier

One of the earliest ensemble algorithms is *Bootstrap aggregation*, also known as *Bagging*. Bootstrapping is a sampling technique in which multiple subsets of training sets (*bags*) are created from the initial training set with replacement [11]. As Figure 3.8 shows, each base classifier (a DT or any other classifier) is trained on a subset of the training set. A new instance is classified using a simple *max* (*majority*) *voting* scheme, whereby each classifier assigns a classification to the instance, and the final classification is the class with the highest number of votes [11] [15].

Random Forests (RF) is a variant of bagging algorithms that uses DTs as the base classifiers. As Figure 3.9 shows, RF randomly selects a set of features which are used to decide the best split at each node of the DT. Finally, to use the constructed RF for prediction on new data, the RF first predicts the target using each DT in the forest. Then, it uses the majority vote of all the DTs prediction and assigns the target with the highest probability to the new data [15] [40].

In the RF, there are different hyperparameters that affect the accuracy of the model, such as the number of DTs, depth, splitting criteria, and how random the data are chosen for the DTs [50]. Using a larger number

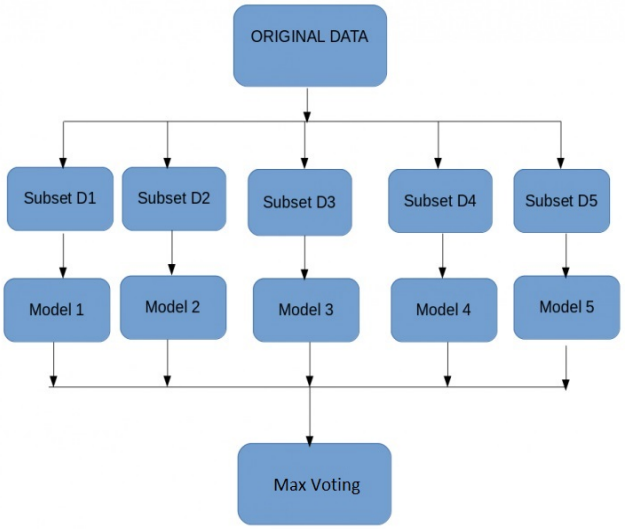


Figure 3.8: General framework of Bagging in ensemble learning

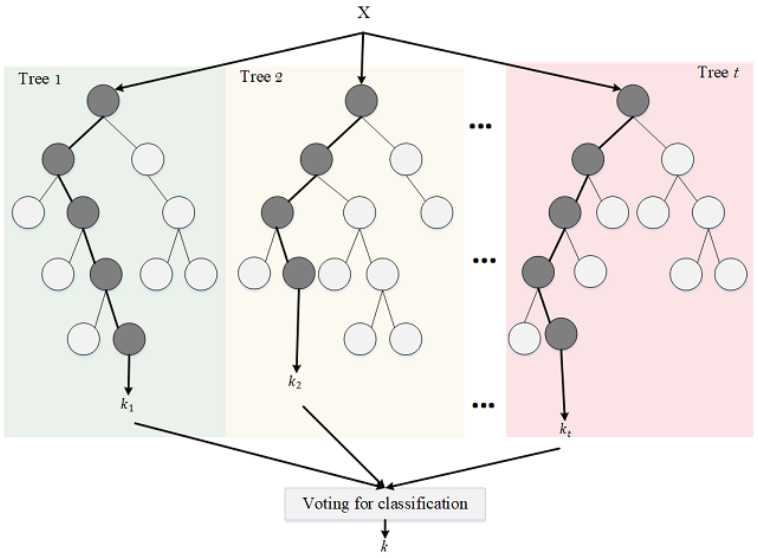


Figure 3.9: General framework in RF [43]. The dark gray circles are the features that RF considered to train each tree. The outputs for each tree are $(k_1, k_2, ..., k_t)$, and k is the final output based on max voting scheme.

of DTs creates a more robust model by reducing overfitting problem. However, having more DTs in the forest requires more time and memory to train the model.

3.4.2 Boosting: AdaBoost Classifier

Unlike the bagging technique, where classifiers are independent and work in parallel, boosting is a sequential process that adaptively changes the distribution of the training set based on the performance of previous classifiers [43]. In practice, boosting is often applied to combine decision trees [40]. Boosting uses multiple steps to do the classification [15]:

1. A subset is created from the original dataset in which all data points are given equal weights.

2. A base classifier is created on this subset and used to make predictions on the whole dataset.

3. Errors are calculated using the actual values and predicted values. The datapoints that are incorrectly predicted are given higher weights, whereas the weights of the datapoints that are correctly classified remain the same. (Based on Figure 3.10, the three misclassified blue-plus points will be given higher weights.)

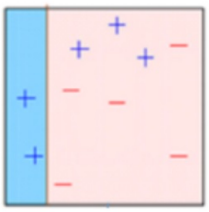


Figure 3.10: The binary classification result from the first base classifier. Three blue-plus points have been classified incorrectly. Therefore, they will be given a higher weight for the subsequent classifier.

4. Another classifier is created, and predictions are made on the dataset. This classifier tries to correct the errors from the previous classifier.

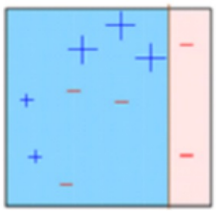


Figure 3.11: The binary classification result from the second base classifier using the weighted dataset. Three blue-plus points that were misclassified in the previous model have been given higher weight.

5. Similarly, multiple classifiers are created, each correcting the errors of the previous classifier. The final classifier (strong learner) is the weighted mean of all the classifiers.

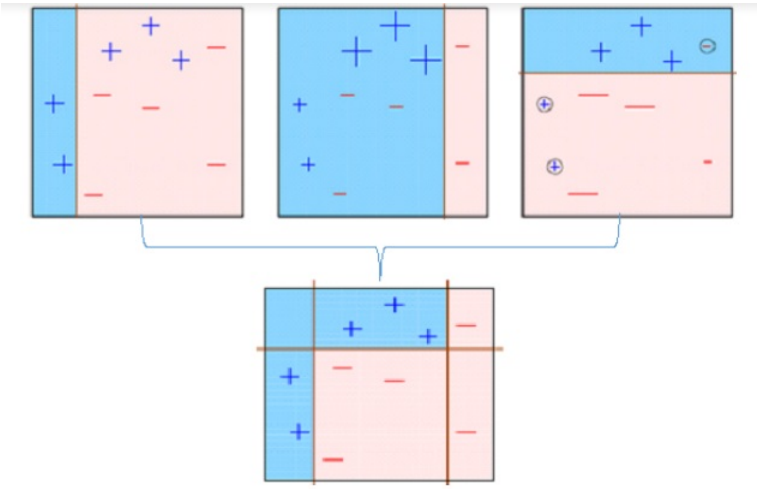


Figure 3.12: The final classifier

Thus, the boosting algorithm combines a number of weak classifiers to form a strong classifier. The individual classifiers would not perform well on the entire dataset, but they work well for some part of the dataset. Thus, each classifier is said to boost the performance of the ensemble.

Adaptive boosting, known as *AdaBoost*, is one of the simplest boosting algorithms, in which DTs are mostly used as the base classifier. [15] [43]. The Adaboost algorithm comprises of different hyperparameters that affect the performance of the model, such as the type of the base classifier, the number of base classifiers, depth, and how random the data are chosen for the base classifiers.

In the AdaBoost algorithm, a training set $(x_1, y_1), ..., (x_m, y_m)$ is taken as input where each x_i belongs to some instance space/vector X, and each label y_i is in some label set Y, assuming $Y = \{-1, +1\}$ for a binary classification [64]. AdaBoost calls a given base classifier repeatedly in a series of rounds $t = 1, 2, ..., T$. The main idea of the algorithm is to maintain a weight or set of weights over the training set. The weight of the training example x_i on round t is denoted $w_t(x_i)$. Initially, all weights are set equally, but on each round, the weights of incorrectly classified examples are increased so that the base classifier is forced to focus on the misclassified examples in the training set. The base classifier's job is to be trained appropriately for the weight w_t and then minimizing the error:

$$\epsilon_t = P\left[\hat{y}_i(x_i) \neq y_i\right], \tag{3.21}$$

where $\hat{y}_i$ is the output of the base classifier for input x_i. AdaBoost chooses a parameter α that intuitively measures the importance/weight that it assigns to $\hat{y}_i(x_i)$. For the binary classification

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right). \tag{3.22}$$

After base classifier is trained, the weight w_t is updated using

$$w_{t+1}(x_i) = \frac{w_t(x_i) \exp(-\alpha_t y_i \hat{y}_i(x_i))}{Z_t}, \tag{3.23}$$

where Z_t is a normalization factor. Finally, The combined classifier H is a weighted majority vote of the T base classifiers where α_t is the weight assigned to $\hat{y}_i$.

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t \hat{y}(x)\right). \tag{3.24}$$

3.4.3 Boosting: XGBoost Classifier

The gradient boosting classifier is another boosting algorithm in which the target outcomes are set based on the gradient of the error to the prediction. A new model is developed in each training set that minimizes prediction error. Extreme Gradient Boosting, known as XGBoost, is a distributed boosting algorithm that uses regression trees as a base classifier. XGBoost has high predictive power and is almost 10 times faster than the other boosting techniques as a result of parallel and distributed computing [17]. Moreover, XGBoost is designed to perform well on sparse features [80] and it uses more accurate approximations to find the best tree model. It also includes an automatic feature selection and a variety of regularizations (built-in L1 and L2 regularization), which reduces overfitting and improves overall performance. Hence it is also known as *regularized boosting* technique.

Like other algorithms, the XGBoost method has some hyperparameters that should be tuned to increase the performance and at the same time decrease the overfitting problem. These hyperparameters are the number of base classifiers, depth, the number of cores used for parallel processing, the number of leaves in a tree, and gamma, which is the minimum loss reduction required to make a split.

3.5 Transfer Learning Classifier

Transfer learning refers to transferring knowledge from different but related source domains to the target model in target domains in order to improve the performance of the target model. As a result, a target model can be constructed without having to rely on a large number of domain data [83]. The wide application prospects of transfer learning have made it one of the most popular and promising areas of machine learning and deep learning. Most recent deep learning approaches rely on transfer learning and pretrained models.

3.5.1 BERT Classifier

In Bidirectional Encoder Representations from Transformers (BERT), transfer learning and bidirectional transformers are combined in order to produce state-of-the-art models for a wide range of NLP tasks [39]. Bidirectional means the BERT model learns information from both the left and the right side of a token's context (the whole text passage) during the training phase to understand the meaning of each token. Moreover, *Encoding* means extracting features by reading and converting the input into numerical representations [65].

The BERT model comprises two stages: pre-training and fine-tuning. Two large corpora of unlabeled text including the entire Wikipedia and Book Corpus are used to train the model during pre-training. For fine-tuning, the model is initialized with the pre-trained parameters, and all the parameters are fine-tuned using labeled data for specific tasks [28]. Fine-tuning needs fewer data and results in quicker development and better results compared to traditional ML classifiers or implementing custom architectures from scratch [22].

The architecture of BERT models is based on the *encoder-decoder* framework, where the encoder is responsible for reading text input, processing, and extracting features. The decoder is responsible for producing a final output/prediction and solving the task [65]. In BERT models, the input representations are computed as follows: each word in the input is first tokenized into word-pieces, and then three *embedding* layers (token, position, and segment) are combined to transform words into vector representations of fixed dimension. In studies of BERT, the term *embedding* refers to the output of a transformer layer.

Figure 3.13 shows how the embeddings are brought together to make the final input token. This process is called *Embedding/Encoding*, where after transforming all the tokens into vector representations, in token embedding, a special token [CLS] is used at the beginning of the first sentence for classification predictions, and [SEP] is added to the end of every sentence that separates input segments from each other [61][37]. Then the model creates segment embeddings by adding a segment 'A' or 'B' to distinguish between the sentences. It also adds the position of each token in the sequence to get position embeddings. The sum of the three embeddings is the final input to the BERT encoder.

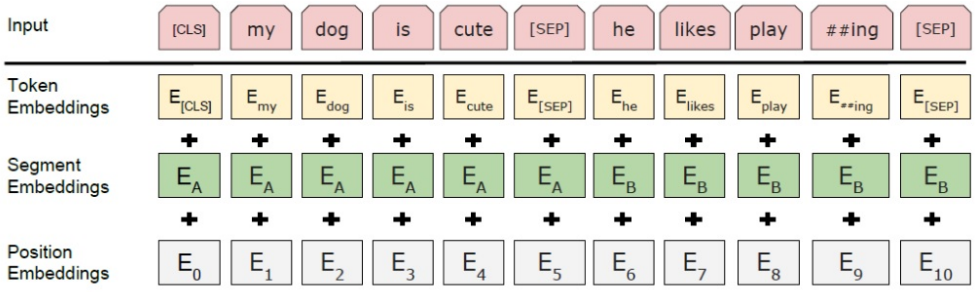


Figure 3.13: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings, and the position embeddings [37].

Then, each sentence will be padded or truncated to have a fixed length. Padding is done with a special [PAD] token, which is at index 0 in the BERT vocabulary [22]. The main reason for converting the words to embeddings is to make them easy for the model to work with. When the words are converted into embeddings, the model can understand the semantic importance of a word in numeric form, and thus it can perform mathematical operations on it.

The encoder-decoder framework performs efficiently on a variety of tasks, e.g., machine translation. However, because all information is stored in a list of feature vectors, the framework may not be able to understand

long and complex inputs. One approach to overcome this limitation is the *attention mechanism* [65]. The main idea is similar to the way humans perceive and understand their environment. Not all information has the same influence on the output. For example, in padding, the attention mechanism explicitly differentiates real tokens from padding tokens. The core idea of attention is calculating and assigning weights to annotations that further determines their amount of influence on the output.

The two pre-trained models of BERT are BERT_{base} (Layers=12, Attention Heads=12, Total Parameters=110M) and BERT_{large} (Layers=24, Attention Heads=16, Total Parameters=340M). Finally, in fine-tuning, one or more fully connected layers are typically added on top of the final encoder layer (Figure 3.14)[61].

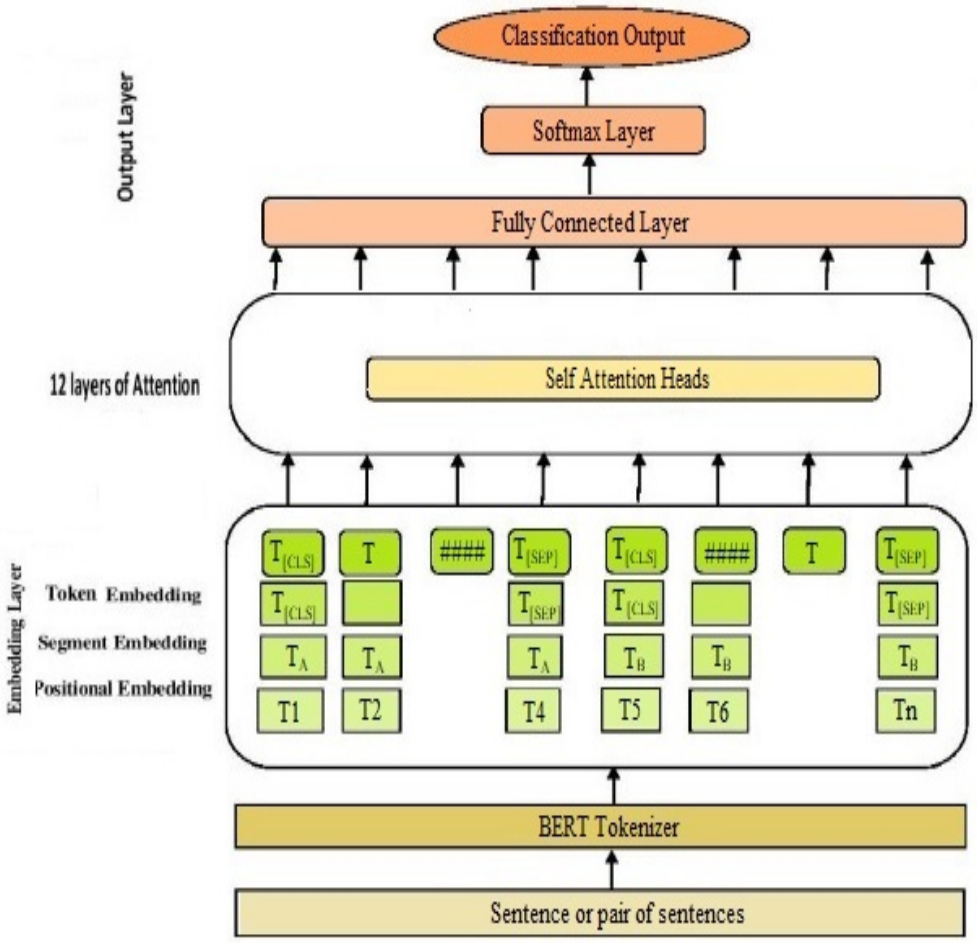


Figure 3.14: Architecture of the BERT model [22]

3.5.2 ALBERT Classifier

"A lite version of BERT", known as ALBERT, was proposed recently to enhance the training and results of the BERT architecture by using parameter sharing and factorizing techniques [44]. There are millions of parameters in the BERT model, which makes it difficult to train. Furthermore, having too many parameters affects the computation speed. To overcome such challenges ALBERT was introduced for its fewer parameters compared to BERT.

ALBERT uses two parameter reduction techniques [65][44]: The first technique is *cross-layer parameter sharing*, which is used to reduce the number of parameters in BERT. For example, BERT_{base} has 12 encoder layers, and during training, the parameters are learned across all encoder layers. However, when it comes to cross-layer, instead of learning parameters across all encoder layers, the parameter of the first encoder layer is shared with all the other encoder layers. This technique prevents the parameter from growing with the depth of the network.

The second technique is *factorized embedding parameterization*. In this case, the model decomposes the large vocabulary embedding matrix into two smaller ones, separating the hidden layers from the vocabulary embedding. This separation makes it easier to grow the hidden size without significantly increasing the parameter size of the vocabulary embeddings. Both techniques have the benefit of reducing the number of parameters for BERT and also improving parameter efficiency and performance.

ALBERT has four different variants [44]. In our study, the ALBERT_{base} is used that comprises 12 encoding layers and 12M parameters.

3.6 Optimization: Grid Search

Hyperparameters are characteristics of a model that are external to the model and whose value are not determined directly from the data. The value of a hyperparameter has to be set before the learning process begins. For example, c in SVM, k in KNN, and the number of hidden layers in Neural Networks.

In order to increase the performance of the model, all the hyperparameters should be optimized. Grid search is the most popular hyperparameter optimization method, in which a finite set of values for each hyperparameter is specified, and grid search evaluates the model on the Cartesian product of these sets [26]. Grid search can often easily be parallelized because the hyperparameter values that the algorithm works with are usually independent of each other [45].

3.7 Model Validation: Simple Split

Different methods can be used to validate a predictive model. This study uses a method called simple split. In the simple split method, the original data are randomized and split into two sets called training and testing sets. The samples are selected with a uniform distribution, e.g., each sample has the same probability of

being selected [60] [25].

The prediction error of model is calculated by comparing the predicted value $\hat{y}$ and the actual value y. Hence, the prediction error becomes

$$\begin{aligned}\epsilon &= \frac{\text{the number of misclassifications}}{\text{the total number of test cases}} \\ &= \frac{\sum_{i=1}^{N} \delta(\hat{y}, y)}{N},\end{aligned}$$

where ϵ is the prediction error and N is the total number of test cases.

3.8 Model Performance Evaluation

3.8.1 Confusion Matrix

Intuitively, after training a model, it is important to know how strongly we should trust that the model's results are correct. In other words, we should determine how effective the model is in terms of its performance on a test set [34]. Different metrics are used to evaluate the performance of a model. Every sample in a testing process has a *True* label and a *Predicted* label. The true label indicates the class to which the sample belongs. The predicted label is the output of the predictor. Let x_i $(i = 1, 2, ..., n)$ be one of n samples, y_i the true label of x_i and $\hat{y_i}$ the prediction result of x_i. Usually, in a binary predictor, $+1$ is used as the label of a positive sample and -1 as the label of a negative sample. The number of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) can be defined as the follows:

$$\begin{aligned}\text{TP} &= |\{x_i | y_i = +1, \hat{y_i} = +1\}|, \\ \text{TN} &= |\{x_i | y_i = -1, \hat{y_i} = -1\}|, \\ \text{FP} &= |\{x_i | y_i = -1, \hat{y_i} = +1\}|, \\ \text{FN} &= |\{x_i | y_i = +1, \hat{y_i} = -1\}|,\end{aligned} \tag{3.25}$$

A confusion matrix, also known as an error matrix, is a 2-by-2 contingency table used to summarize the performance of a classification model on the test data [34]. Confusion matrices are useful because they give direct comparisons of True Positives, False Positives, True Negatives, and False Negatives [50]. Confusion matrices are used to visualize important predictive analytics like specificity, accuracy, and precision [42].

A confusion matrix has two dimensions, labeled as actual and predicted, with a set of classes on both dimensions. Figure 3.15 presents an example of a confusion matrix for a binary classification model with the classes of positive and negative. In the table, values on the diagonal of the matrix show the number of correct predictions for classes positive and negative, whereas off-diagonal values show the number of misclassifications for those classes.

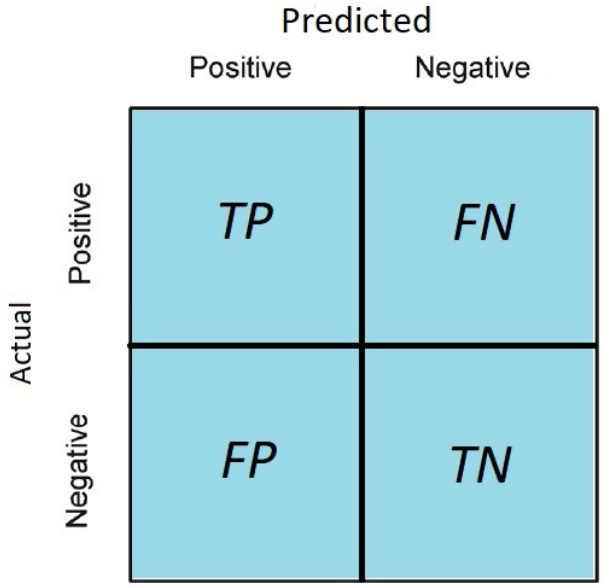


Figure 3.15: Confusion matrix for a binary classification model [34]

3.8.2 Accuracy

Accuracy is the number of correct predictions made by the model over the total number of predictions [14]. Using the four counts described in the previous section, the accuracy can be calculated by,

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{3.26}$$

3.8.3 Precision

Precision or positive predictive value shows the number of correct predictions among the samples that are predicted to be positive and is calculated by [50],

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{3.27}$$

3.8.4 Sensitivity

Sensitivity or true positive rate (TPR) is the frequency of correctly predicted positive samples among all real positive samples. It measures the ability of a model in identifying positive samples and is defined by [34],

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{3.28}$$

3.8.5 Specificity

Specificity or true negative rate (TNR) measures the ability of a predictor in identifying negative samples and is calculated by [34],

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}}. \tag{3.29}$$

Having sensitivity and specificity, false negative rate (FNR) and false positive rate (FPR) become

$$\begin{aligned} \text{FNR} &= 1 - \text{Sensitivity}, \\ \text{FPR} &= 1 - \text{Specificity}. \end{aligned} \tag{3.30}$$

3.8.6 F1-Score

The F1-Score is a middle ground between precision and sensitivity [50]. It combines precision and sensitivity by taking the harmonic mean of precision and sensitivity. The F1-Score is calculated by

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}}. \tag{3.31}$$

3.8.7 Receiver Operating Characteristic Curve

The receiver operating characteristic (ROC) curve is commonly used to illustrate the performance of a binary classifier [34]. The ROC curve is a two-dimensional curve representing the TPR and FPR on its vertical and horizontal axes, respectively. The performance of a binary classifier based on its ROC curve is evaluated by a single number that defines the area under the curve (AUC) or the area between the curve and the FPR axis. The AUC can also be used to compare the performance of multiple classifiers. Figure 3.16 depicts the ROC curve for a classifier. The diagonal in Figure 3.16 separates the square between (0,0) and (1,1) into two parts. This diagonal is called the line of no-discrimination.

The AUC can be calculated by [14],

$$\text{AUC} = \frac{1 + \text{TPR} - \text{FPR}}{2}. \tag{3.32}$$

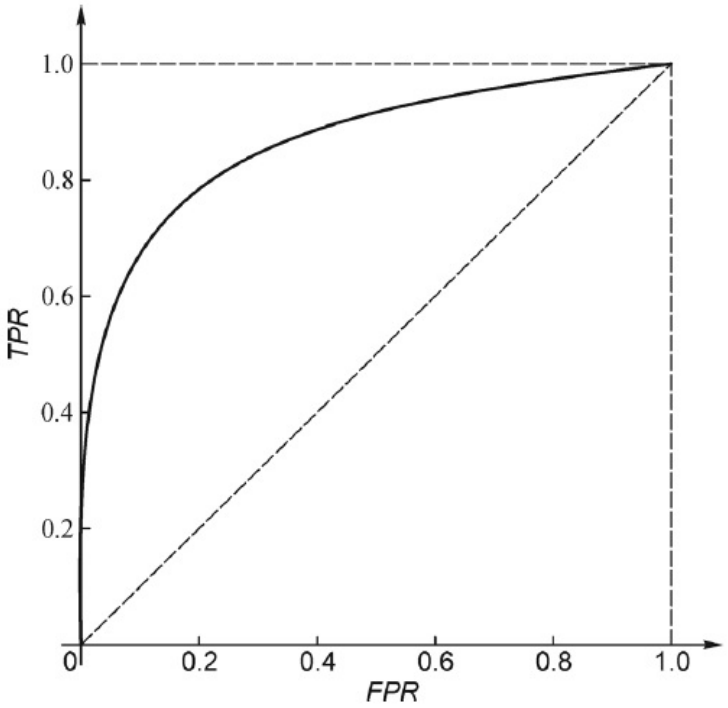


Figure 3.16: An ROC curve for a classifier [34]. The horizontal axis is the FPR and the vertical axis is TPR. The solid curve is the ROC curve. The dashed diagonal is called the line of no-discrimination. The closer the curve to the top left corner, the better performance the classifier has.

3.9 Overfitting and Underfitting

Overfitting and *underfitting* are two important issues in training the ML classifiers. Overfitting occurs when a model performs extremely well on the training set while fitting poorly on the testing set. In other words, the model does not generalize well from observed data to unseen data [78] [51]. In overfitting, models tend to reflect all the data, including unavoidable noise on the training set, instead of learning the relationships between inputs and outputs from training data [41].

The converse problem to overfitting is underfitting, where an algorithm lacks sufficient training data to fully learn the true relationship [6]. An underfit model will have low training and testing accuracy while an overfit model will have high training accuracy and a relatively low testing accuracy.

4 Methodology and Results

The purpose of this chapter is to describe the structure of the provided data files and the methodology used to clean the data and train eleven ML algorithms. Furthermore, the performance of the six ML algorithms LR, NB, MNB, KNN, SVM, and DTC and three ensemble learning algorithms RF, AdaBoost, and XGBoost, and two transfer learning methods BERT and ALBERT classifiers are compared to see whether any can be used to satisfactorily the predictive performance in the relevant tweet detection within the context of people impacted by dementia and COVID-19.

4.1 Tweet Extraction and Twitter Data Structure

The term *social media* refers to a type of computer-based technology that enables the sharing of ideas, thoughts, and information through the creation of virtual networks and communities. The largest social media networks include Twitter, Facebook, Instagram, YouTube, and TikTok. With over 330 million monthly users and an average of 500 million tweets daily, the microblogging and social networking website Twitter presents an innovative opportunity for people to share their COVID-19 experiences [4].

For the first dataset, known as *First wave* dataset, 5,063 dementia-related tweets from [4] were used, which were collected from Twitter during the period from February 15 to September 7, 2020. They used the following search terms: *Dementia* OR *Alzheimer* used in combination with *COVID-19*, OR *COVID*, OR *Corona*. Consequently, the dataset consists of experiences of people impacted by Alzheimer/dementia and COVID-19. The dataset was labeled manually into twelve categories by seven authors using thematic analysis(see Table 4.1). The *First wave* dataset was used to train different ML, ensemble learning, and transfer learning models. In [4], all the labels from 1 to 9 were relevant and used for the study, and the rest were neglected. Thus, to have a binary classification in this study, all the labels from 1 to 9 and 98 to 100 are merged and labeled as 1 (relevant) and 0 (irrelevant), respectively (see Table 4.2).

Because the existing Twitter study has already explored the early stages of the pandemic on people with dementia [4], for the second dataset, our study focused on the later stages of the pandemic (i.e., September 8, 2020 to December 8, 2021) [5] with the same search terms. As a result, a dataset, known as *Longitudinal* dataset, comprising 110,528 tweets was collected in CSV format using Twint, an advanced scraping tool that enables users to scrape tweets without the use of Twitter's application programming interface. Twint enables scraping tweets without certain restrictions such as the number of tweets scraped, the frequency and time period of scrapes, and the requirement of a Twitter account [73].

Table 4.1: The *First wave* data structure used in [4]

Code	Code Name	Instances
1	Death	391
2	Fear for Person with Dementia's Health/Wellbeing	637
3	Challenges & Unmet Needs	391
4	Separation/Restricted Visiting	856
5	Formal Care Provider/Workforce Challenges	157
6	Supports Described	206
7	Informal Caregiver's Health/Wellbeing	93
8	Stories of Survival	10
9	User Identifies as Person with Dementia	112
98	Questioning Cause of Death	461
99	No Intersection of COVID and Dementia	1,713
100	Dementia Stigma	36
Total:		5,063

Table 4.2: The *First wave* data structure used in this study

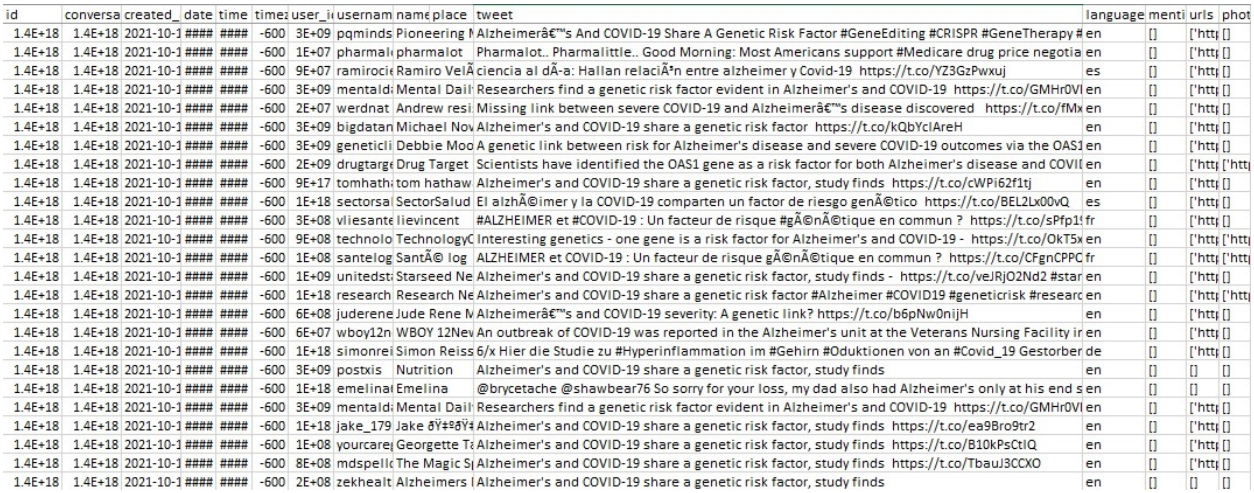
Label	Instances
1	2,853
0	2,210
Total:	5,063

Moreover, to examine the Twitter discourse on dementia during Alzheimer's Awareness Month in Canada, we collected a third dataset known as *Alzheimer's Awareness Month* dataset [3] comprising 1289 tweets during the period from January 1 to January 31, 2022. The search terms used in this study consisted of various phrases of Alzheimer's Awareness Month (i.e., #AlzheimersAwarenessMonth, #AlzAwareness, #dementia-awareness, dementia month, dementia awareness month, Alzheimer's awareness month, Alzheimer's month, January is Alzheimer's Awareness month) or tweets using a combination of either *Canada* and *dementia* or *Alzheimer's*. Other tweets scraped were from Canada's national and provincial Alzheimer's organizations (i.e., @alzCanada, @AlzheimerOnt, @AlzheimerSK, @DementiaAB_NT, @AlzheimerNS, @AlzheimerNB, @AlzheimerPEI, @alzheimerMB, @AlzheimerBC, @asnl2, and @FqsaAlzh).

4.2 Data Pre-processing

Tweets include many misspelled words, irrelevant characters, emoticons, and unconventional syntax, all of which are considered noise. Moreover, not all the columns in data files are relevant to our study. Figure 4.1 shows a sample of what information a raw CSV file of tweet collections include. Accordingly, to have consistent

data pre-processing with the research analyzing Twitter data in [4], all the pre-processing steps mentioned in the study are applied to the *Longitudinal* dataset.



id	conversa	created_	date	time	timez	user_i	usernam	name	place	tweet	language	menti	urls	phot
1.4E+18	1.4E+18	2021-10-1	####	####	-600	3E+09	pqminds	Pioneering M		Alzheimerâ€™s And COVID-19 Share A Genetic Risk Factor #GeneEditing #CRISPR #GeneTherapy #	en	[]	['http	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	1E+07	pharmal	pharmalot		Pharmalot.. Pharmalittle.. Good Morning: Most Americans support #Medicare drug price negotia	en	[]	['http	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	9E+07	ramiroci	Ramiro Vel	Ã	ciencia al dÃ-a: Hallan relaciÃ³n entre alzheimer y Covid-19 https://t.co/YZ3GzPwxuj	es	[]	['http	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	3E+09	mentald	Mental Dail		Researchers find a genetic risk factor evident in Alzheimer's and COVID-19 https://t.co/GMHr0Vl	en	[]	['http	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	2E+07	werdnat	Andrew resi		Missing link between severe COVID-19 and Alzheimerâ€™s disease discovered https://t.co/fMx	en	[]	['http	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	3E+09	bigdatan	Michael Nov		Alzheimer's and COVID-19 share a genetic risk factor https://t.co/kQbYclAreH	en	[]	['http	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	3E+09	geneticli	Debbie Moo		A genetic link between risk for Alzheimer's disease and severe COVID-19 outcomes via the OAS1	en	[]	['http	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	2E+09	drugtarg	Drug Target		Scientists have identified the OAS1 gene as a risk factor for both Alzheimer's disease and COVII	en	[]	['http	['htt
1.4E+18	1.4E+18	2021-10-1	####	####	-600	9E+17	tomhathi	tom hathaw		Alzheimer's and COVID-19 share a genetic risk factor, study finds https://t.co/cWPi62f1tj	en	[]	['http	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	1E+18	sectorsa	SectorSalud		El alzhÃ©imer y la COVID-19 comparten un factor de riesgo genÃ©tico https://t.co/BEL2Lx00vQ	es	[]	['http	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	3E+08	vliesante	lievincent		#ALZHEIMER et #COVID-19 : Un facteur de risque #gÃ©nÃ©tique en commun ? https://t.co/sPfp1	fr	[]	['http	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	9E+08	technolo	TechnologyC		Interesting genetics - one gene is a risk factor for Alzheimer's and COVID-19 - https://t.co/OkT5x	en	[]	['http	['htt
1.4E+18	1.4E+18	2021-10-1	####	####	-600	1E+08	santelog	SantÃ© log		ALZHEIMER et COVID-19 : Un facteur de risque gÃ©nÃ©tique en commun ? https://t.co/CFgnCPPC	fr	[]	['http	['htt
1.4E+18	1.4E+18	2021-10-1	####	####	-600	1E+09	unitedsti	Starseed Ne		Alzheimer's and COVID-19 share a genetic risk factor, study finds - https://t.co/veJRjO2Nd2 #star	en	[]	['http	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	1E+18	research	Research Ne		Alzheimer's and COVID-19 share a genetic risk factor #Alzheimer #COVID19 #geneticrisk #researc	en	[]	['http	['htt
1.4E+18	1.4E+18	2021-10-1	####	####	-600	6E+08	juderene	Jude Rene M		Alzheimerâ€™s and COVID-19 severity: A genetic link? https://t.co/b6pNw0nijH	en	[]	['http	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	6E+07	wboy12n	WBOY 12Nev		An outbreak of COVID-19 was reported in the Alzheimer's unit at the Veterans Nursing Facility in	en	[]	['http	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	1E+18	simonrei	Simon Reiss		6/x Hier die Studie zu #Hyperinflammation im #Gehirn #Oduktionen von an #Covid_19 Gestorber	de	[]	['http	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	3E+09	postxis	Nutrition		Alzheimer's and COVID-19 share a genetic risk factor, study finds	en	[]	[]	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	1E+18	emelina	Emelina		@brycetache @shawbear76 So sorry for your loss, my dad also had Alzheimer's only at his end s	en	[]	[]	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	3E+09	mentald	Mental Dail		Researchers find a genetic risk factor evident in Alzheimer's and COVID-19 https://t.co/GMHr0Vl	en	[]	['http	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	1E+18	jake_179	Jake ðŸ‡ªðŸ‡		Alzheimer's and COVID-19 share a genetic risk factor, study finds https://t.co/ea9Bro9tr2	en	[]	['http	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	1E+08	yourcare	Georgette Ta		Alzheimer's and COVID-19 share a genetic risk factor, study finds https://t.co/B10kPsCtlQ	en	[]	['http	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	8E+08	mdspell	The Magic S		Alzheimer's and COVID-19 share a genetic risk factor, study finds https://t.co/TbauJ3CCXO	en	[]	['http	[]
1.4E+18	1.4E+18	2021-10-1	####	####	-600	2E+08	zekhealt	Alzheimers I		Alzheimer's and COVID-19 share a genetic risk factor, study finds	en	[]	[]	[]

Figure 4.1: CSV file of raw tweet collections containing some of the columns

Here are all the steps to remove noise and obtain an appropriate dataset for analysis:

1. Remove either empty or irrelevant columns (e.g., id, conversation_id, time, timezone, user_id, username, name, etc.),

2. Filter out non-English language tweets,

3. Filter out duplicate tweets,

4. Filter out advertising tweets containing a permalink,

5. Filter out unrelated or political tweets about *Donald Trump* or *Joe Biden* or *Tom Seaver*, the major league baseball player who was reported to have died on August 31, 2020, due to COVID-19 and dementia,

6. Filter out reply tweets as they often were missing information and only contained half of the conversation,

7. Filter out tweets that did not contain synonyms for familial relationships or friends or acquaintances in order to improve the likelihood of scraping tweets that described personal experiences of dementia during the COVID-19 pandemic. Table 4.3 shows the familial/friend keywords used in this study. Figure 4.2 shows the number of tweets left after each filtering step. Figure 4.3 shows the filtered data after passing through all the aforementioned steps.

8. Remove emojis,

After the filtering steps, the remaining 6,243 tweets were divided among 11 coders in a research team for manual labeling using thematic analysis based on the codebook used in [4]. Inter-coder reliability was managed by the team leader, who reviewed 25% of all labeling on a random basis. The average inter-coder reliability was 83.4%. Again, the labels are merged and categorized into two classes of relevant (1) and

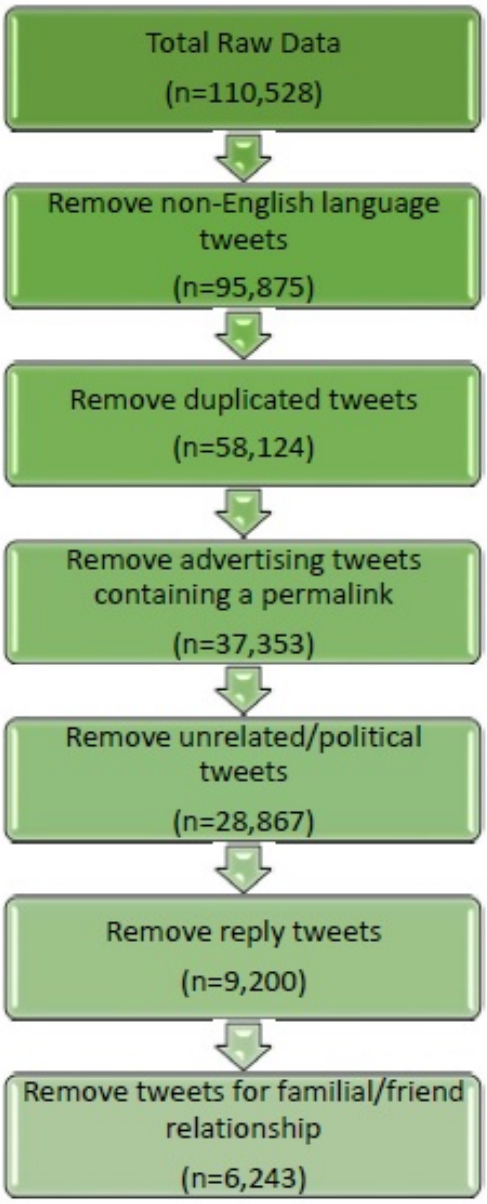


Figure 4.2: The flowchart of filtering steps containing the number of remaining tweets in each step.

Table 4.3: The keywords used to filter out tweets not containing synonyms for familial/friend relationships

Keywords
my, his, her, mine, family, sister, friend, dad, mom, mum, parents, husband, wife, uncle, aunt, grandma, grandpa, grandson, nana, grandparent, grandmother, grandfather, niece, nephew, grandchild, granddaughter, father, brother, mother, stepsister, stepbrother, mother-in-law, father-in-law, cousin, pal, neighbour

Text
I lost my mother to COVID-19 in April of this year. Alzheimer's robbed her of the ability to to talk, to walk, to think . . . to live meaningfully. Still, I treated her as if she was still there.
Guys, my cousin found the COVID vaccine, cancer cure and answer to Alzheimer's!
Seeing some peoples stories about them or their family members on the "The Caretaker" album are really getting to me. My Grandad had dementia before he passed due to covid, ar
So my 18 year old daughter has covid. Not from going out and seeing friends, from working as a carer for dementia patients without PPE. She deferred her paediatric nursing degree t
My resident (who has dementia so doesnt know what covid even is) asked me if theyre jealous of my pretty face and thats why they make me wear a mask
I know... My dear wife's Mom now has Covid on top of her dementia and nobody is allowed to go and see her... So sad
We were also a Jeopardy household. I lost my Mom to covid and dementia this year - the only thing sometimes to calm her was to watch Jeopardy together. Blessings to the Trebek f
Will we ever get our response to Covid right A pals elderly mum with dementia was taken into hospital. They diagnosed Covid and sent her home to isolate in an upstairs flat where r
Respected Modiji, I m facing serious issue with my mother to submit life certi for pension. She is 82 yrs with dementia and cannot go to bank for verification and also for COVID. Onl
we lost her earlier this summer. dementia. she spent the last six months of her life locked away from her family, not knowing why, in a nursing home under Covid quarantine. I think
rip both my grandmas one passed away from cancer, the other from type 2 diabetes. rip my grandpa who passed away not too long ago from COVID-19 and having alzheimers for way t
Ive listened to waaaay too many Matthew-McConaughey-as-a-guest podcasts lately due to his relentless promotion of the new memoir. Maybe Ive got the covid? Maybe its Stockholr
Im an only child living on the Left Coast. My parents and extended family back East....my Dad has dementia and Id do anything to spend holidays with them but I cant COVID-19 is re
Just up thinking about my uncle and how he's having to take care of an alzheimer's patient while battling covid alone. That's too much for anybody.
This has completely set me off. Good dementia and alzheimers care is a wonderful thing. Was just talking this weekend with a friend about financial impact of Covid-19. @jewishcare
She suffers from memory loss as part of her dementia, which in some ways has been a blessing and a curse. She can't remember if we've been or not, she can't remember being on de
Hahaha Well I used to listen to #JimCornette and his rants about wrestling. But since he thinks a corrupt pedophile w/dementia is gonna save America from COVID-19 I laughed at his
A friend of my sister has a mother who is dying in a nursing home. Her friend was only allowed to visit her one time because of Covid. Hes heartbroken even though her dementia has
Just carry on at normal. What Pandemic! Its a joke. In my area we have had 3 deaths from covid since March! I have no faith they were only covid related. As they tried to fake my M
This is beautiful. As a daughter of a 91 yr old mom with Alzheimers this makes me sob with joy. Ive always known my mom is still there. She may not remember me but she remembe

Figure 4.3: A sample of the filtered data after passing through multiple steps.

irrelevant (0). Table 4.4 shows the final structure of the labeled dataset. All the steps were repeated for the *Alzheimer's Awareness Month* dataset except the fourth, fifth, and the seventh steps. Table 4.5 shows the final structure of the *Alzheimer's Awareness Month* dataset. Figure 4.4 shows a sample of the filtered tweet from the previous steps.

In order to prepare the datasets to train and test the ML classifiers, a few more steps were applied to all datasets.

9. Tokenization (see Figure 4.5),

10. Turn to lowercase (see Figure 4.6),

11. Remove punctuation and possessive pronouns (see Figure 4.7),

12. Lemmatization (see Figure 4.8),

"I sympathise. My mother-in-law's care home has had to stop all visits because visitors disobeyed rules and now they have new cases of covid. She also suffers from dementia. The problem was that visitors cared only about their own family members and not about other residents/staff"

Figure 4.4: A sample of the filtered tweet from the previous steps

"'i', 'sympathise.', 'my', "mother-in-law's", 'care', 'home', 'has', 'had', 'to', 'stop', 'all', 'visits', 'because', 'visitors', 'disobeyed', 'rules', 'and', 'now', 'they', 'have', 'new', 'cases', 'of', 'covid.', 'she', 'also', 'suffers', 'from', 'dementia.', 'the', 'problem', 'was', 'that', 'visitors', 'cared', 'only', 'about', 'their', 'own', 'family', 'members', 'and', 'not', 'about', 'other', 'residents/staff'"

Figure 4.5: A sample of the filtered tweet after tokenization

"'i' 'sympathise.' 'my' 'mother-in-law's' 'care' 'home' 'ha's 'had' 'to' 'stop' 'all' 'visits' 'because' 'visitors' 'disobeyed' 'rules' 'and' 'now' 'they' 'have' 'new' 'cases' 'of' 'covid.' 'she' 'also' 'suffers' 'from' 'dementia.' 'the' 'problem' 'was' 'that' 'visitors' 'cared' 'only' 'about' 'their' 'own' 'family' 'members' 'and' 'not' 'about' 'other' 'residents/staff'"

Figure 4.6: A sample of the filtered tweet after turning to lowercase

"i' 'sympathise' 'my' 'motherinlaw' 'care' 'home' 'has' 'had' 'to' 'stop' 'all' 'visits' 'because' 'visitors' 'disobeyed' 'rules' 'and' 'now' 'they' 'have' 'new' 'cases' 'of' 'covid' 'she' 'also' 'suffers' 'from' 'dementia' 'the' 'problem' 'was' 'that' 'visitors' 'cared' 'only' 'about' 'their' 'own' 'family' 'members' 'and' 'not' 'about' 'other' 'residentsstaff''

Figure 4.7: A sample of the filtered tweet after removing punctuation and possessive pronouns

"i' 'sympathise' 'my' 'motherinlaw' 'care' 'home' 'have' 'have' 'to' 'stop' 'all' 'visit' 'because' 'visitors' 'disobey' 'rule' 'and' 'now' 'they' 'have' 'new' 'case' 'of' 'covid' 'she' 'also' 'suffer' 'from' 'dementia' 'the' 'problem' 'be' 'that' 'visitors' 'care' 'only' 'about' 'their' 'own' 'family' 'members' 'and' 'not' 'about' 'other' 'residentsstaff''

Figure 4.8: A sample of the filtered tweet after lemmatization

Table 4.4: The *Longitudinal* data structure used in this study

Label	Instances
1	3,014
0	3,229
Total:	6,243

Table 4.5: The *Alzheimer's Awareness Month* data structure used in this study

Label	Instances
1	1,048
0	241
Total:	1,289

13. Remove stop words (see Figure 4.9),

```
' 'sympathise' 'motherinlaw' 'care' 'home' 'stop' 'visit' 'visitors' 'disobey' 'rule' 'new' 'case'
'covid' 'also' 'suffer' 'dementia' 'problem' 'visitors' 'care' 'family' 'members'
'residentsstaff''
```

Figure 4.9: A sample of the filtered tweet after removing stop words

14. TF-IDF vectorization (see Figure 4.10). TD-IDF has an important hyperparameter, called *max-features*, which is a cutoff value or threshold that determines the number of the most representative words that should be considered for the rest of analysis. Setting this threshold is based on experience or looking at the scores of different models [82]. Here, a threshold of 1250 is used based on experiments and scores of different ML classifiers. In other words, considering 1250 most representative words leads to the highest performance.

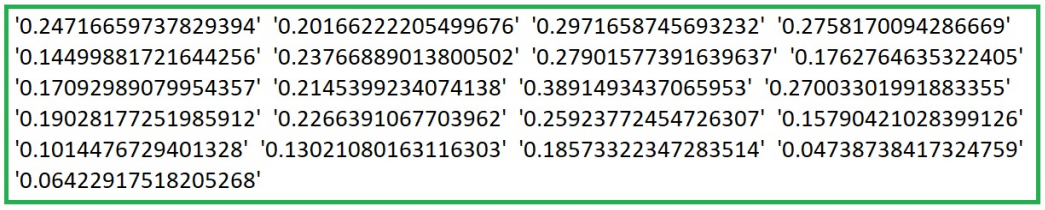


Figure 4.10: A sample of vectors/weights for the filtered tweet after the TF-IDF vectorization

4.3 Classifier Selection

In this study, the goal of building a model from the dataset is to provide health care researchers with a tool that helps them identify relevant tweets without having to go through all the tweets manually. In this case, eleven ML classifiers were tuned, trained using the *First wave* dataset, and compared in terms of their performances. In subsection 4.3.1, the result of hyperparameter tuning for each classifier is discussed. Moreover, in subsection 4.3.2, the results for all the ML classifiers have been measured according to accuracy, precision, sensitivity, specificity, AUC, FNR, FPR, and F1-score. All results reported were obtained using the Python programming language version 3.8.5 on an *AMD A10-5750M APU with Radeon(tm) HD Graphics* personal computer with 16GB RAM and *Nvidia K80 GPU* within Google Colab.

4.3.1 Tuning Classifiers

In this section, the result of tuning the hyperparameters of each classifier is discussed. Using grid search on the training set, the optimal hyperparameters for the studied classifiers are found. Table 4.6 shows all the values of the hyperparameters used in grid search for each classifier. The optimal values selected are shown in bold-face. It should be mentioned that some of the classifiers used in this study do not have any significant hyperparameter (e.g., NB and MNB classifiers).

4.3.2 Comparison of the Classifiers

In this section, the results of ML classifiers are discussed. The *First wave* dataset is split up into two groups, 90% training and 10% testing. This puts 5,554 data points in the training set and 509 in the testing set. The results are obtained by applying the LR, NB, MNB, KNN, SVM, DTC, RF, AdaBoost, XGBoost, BERT, and ALBERT classifiers on the split data set. Table 4.7 shows the validity scores for the eleven studied classifiers. The best values are shown in bold-face.

The results from Table 4.7 show that for the training set, the SVM classifier obtained the highest accuracy, precision, sensitivity, specificity, AUC, and F1-score compared to the other classifiers. The RF classifier followed the SVM classifier and scored second in accuracy, precision, specificity, AUC, and F1-score. For the

Table 4.6: All the values of the hyperparameters used in grid search for each classifier.

Method	Values
LR	C = [0.1, **0.5**, 1, 2], Multi-calss = ['**ovr**'], Solver = ['newton-cg', 'lbfgs', '**liblinear**', 'sag', 'saga']
KNN	N_neighbors = [3, **5**, 7,..., 51] , Metrics = ['**euclidean**', 'manhattan']
SVM	C = [0.1, 0.5, **1**, 1.5, 2], Kernel = ['linear', 'poly', '**rbf**', 'sigmoid']
DT	Criterion = ['**entropy**', 'gini'], Min-sample-leaf = 1, 2, **4**, 6, 8], Min-sample-split = [1, 2, 3,..., **10**,..., 20]
RF	N_estimators = [100, 200, **300**, 400, 500], Criterion = ['**gini**', 'entropy'], Max_depth = [2, 4, 6,..., **32**,..., 64]
AdaBoost	Base_estimators = [LR, **DT**, SVM], N-estimators = [10, 20, 30,..., **100**,..., 500], Max-depth = [1, 2, 3, 4, **5**,..., 20]
XGBoost	Max-depth = [1, 2, 3, 4, 5, **6**,..., 20], learning_rate = [0.01, **0.015**, 0.02, 0.025,..., 0.1]

testing set, the ALBERT classifier reached the highest accuracy, AUC, and F1-score. Figure 4.11 shows the ROC curves and AUC values for the worst, medium, and best classifiers.

According to the results and the fact that the ALBERT model had the least overfitting problem compared to all other classifiers, it is then applied to the *Longitudinal* dataset and reached the accuracy of 80% in classifying relevant and irrelevant tweets.

Finally, the ALBERT model is applied to the *Alzheimer's Awareness Month* dataset and reached an accuracy of 30% due to the differences existing in the *Alzheimer's Awareness Month* dataset compared to the first and the second datasets. Accordingly, this model is then retrained using 10% of the tweets in the *Alzheimer's Awareness Month* dataset and tested on 90% of the rest. The result was an accuracy of 88% for the classification of relevant and irrelevant tweets showing that this model is an agile and flexible model, which can be applied to the other related datasets.

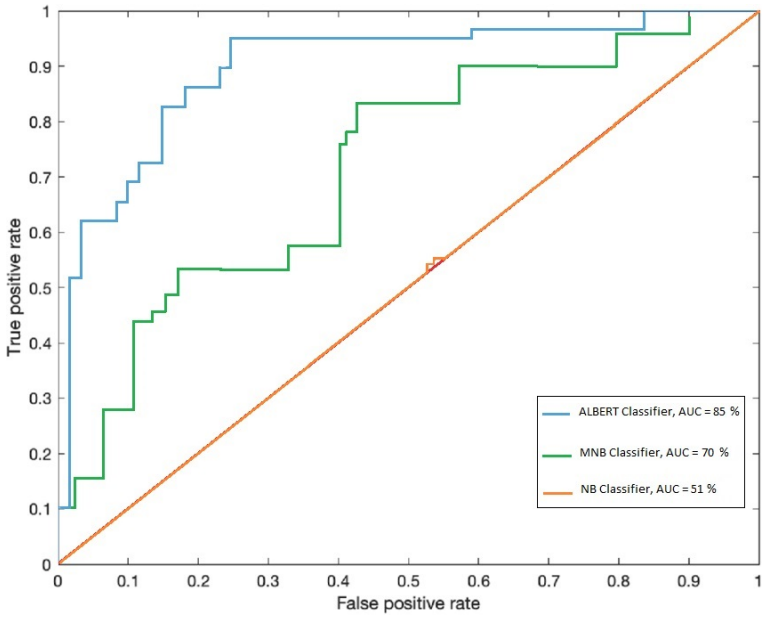


Figure 4.11: Comparison of the ROC curves and AUC values for the worst, medium, and best classifiers.

Table 4.7: Comparison of the eleven ML classifiers using various performance metrics

		Accuracy	Precision	Sensitivity	Specificity	AUC	FPR	FNR	F1-Score
LR	Train	86	84	82	88	85	12	18	83
	Test	70	67	65	74	70	26	35	66
NB	Train	79	76	76	81	79	19	24	76
	Test	66	45	**99**	3	51	97	**1**	62
MNB	Train	79	85	64	91	78	9	36	73
	Test	72	76	53	87	70	13	47	62
KNN	Train	78	79	67	87	77	13	33	72
	Test	63	60	53	71	62	29	47	56
SVM	Train	94	98	93	99	96	1	7	96
	Test	76	80	63	87	75	13	37	71
DT	Train	77	73	74	79	76	21	26	73
	Test	67	64	59	73	66	27	41	62
RF	Train	89	93	80	96	88	5	20	86
	Test	77	**83**	60	**90**	75	**10**	40	70
AdaBoost	Train	74	70	72	76	74	24	28	71
	Test	75	74	76	75	75	26	24	73
XGBoost	Train	73	78	54	88	71	12	46	64
	Test	74	80	56	88	72	12	44	66
BERT	Train	81	79	86	85	86	15	14	83
	Test	80	79	84	85	84	15	16	81
ALBERT	Train	83	80	87	86	87	14	13	84
	Test	**82**	78	86	84	**85**	16	14	**82**

4.4 Discussion

Looking at Table 4.7, in order to detect future tweets for people impacted by dementia during the COVID-19 pandemic, a model with high sensitivity should be considered, which refers to percentage of tweets which were correctly identified as being relevant. As far as the sensitivity is concerned, the NB classifier with 99% could be the clear winner followed by the ALBERT classifier with 86%. However, the NB classifier suffers from overfitting.

Similarly, in order to detect irrelevant tweets, a model with high specificity should be considered. Although the RF classifier had the highest specificity and correctly classified 90% of the actual irrelevant tweets, it suffers from overfitting. In order to have a balanced predictive power that is good for detecting relevant tweets but also careful in not incorrectly labeling tweets as being relevant, the F1-score can be used as a general metric of the predictive performances. Doing so, the ALBERT classifier provided the best performance (82%), with the BERT model following at 81%.

Moreover, according to the Figure 4.11 and Table 4.7, the ALBERT classifier had the highest AUC value compared to all other classifiers. A model with a higher AUC value has better performance in distinguishing between the positive and negative classes. In other words, the ALBERT model is able to separate the two relevant and irrelevant classes better than the rest of the classifiers.

Considering the overfitting problem, the results for the training set and testing set showed that the LR, NB, MNB, KNN, SVM, DT, and RF classifiers suffer from overfitting because they performed poorly on the testing set compared to the training set. For example the SVM classifier reached the accuracy of 94% on the training set. However, on the testing set, it obtained 76%. When the training accuracy is high, but the test accuracy is not as high, then the classifier likely has an overfitting problem, and instead of fitting the relationship between the input and output, it fits the details and noise in the training data such that it negatively affects the performance of the model on test data. Accordingly, the classifier does not obtain the proper output for each unseen input in the testing set.

Moreover, the AdaBoost and XGBoost classifiers suffer from underfitting due the fact that the values in the performance metrics for both training and testing sets were low. Also, some performance metrics such as accuracy, precision, sensitivity, and F1-score in the training set were less than the testing set. For instance, the AdaBoost classifier reached the accuracy of 74% and 75% on the training set and testing set suggesting that the classifier may lack sufficient training. In some classifiers, inadequate training data lead to poor training and underfitting.

BERT and ALBERT classifiers had the least overfitting problem compared to the other classifiers, and the ALBERT classifier beat the BERT classifier with better results in most of the performance metrics. Moreover, the ALBERT classifier, which had the least amount of overfitting compared to all other classifiers, is applied to the *Longitudinal* dataset (completely unseen dataset) and reached the accuracy of 80% in classifying relevant and irrelevant tweets for people impacted by dementia and COVID-19.

Finally, the ALBERT model was trained using 10% of the *Alzheimer's Awareness Month* dataset and reached an accuracy of 88% on the rest in classifying relevant and irrelevant tweets. It can be concluded that, in the context of people impacted by Alzheimer's/Dementia, the ALBERT model can be trained on a small sample of a labeled dataset and used to predict the rest of the unlabeled data efficiently in order to save time for health care researchers and help them focus on the main goal of the project rather than manually labeling the data.

5 Conclusion and Future Work

5.1 Conclusion

Research in health care has lagged behind other disciplines despite the availability of large datasets from social media networks such as Twitter. One of the problems for the health care researchers is the unstructured nature of the data scraped from Twitter, from which extracting insights can be a challenging task. Accordingly, pre-processing methods should be applied on the data in order to analyze, understand, organize, and sort useful data.

In addition, the problem is not only the large amount of data to be processed, but also the fact that some of the data contains a lot of irrelevant information, which restricts their use for further analysis. Generally, the data need to be labeled manually. Manual labeling, which is a part of thematic analysis, is a tedious, time-consuming, and daunting task. Thus, health care researchers and social scientists would be helped by an automated tool that is able to detect irrelevant tweets. This research performs a comparison of ML techniques within the specific context of people impacted by dementia and COVID-19 in order to construct a tool to be available for health care researchers to detect relevant and irrelevant tweets efficiently.

The *First wave* dataset was the one used in [4], and the *Longitudinal* and Alzheimer's Awareness Month datasets were scraped from Twitter and were cleaned and organized to be used for the purpose of this research. A number of ML classifiers known as the Logistic Regression (LR) classifier, Naïve Bayes (NB) classifier, Multinomial Naïve Bayes (MNB) classifier, K-Nearest Neighbors (KNN) classifier, Support Vector Machine (SVM) classifier, Decision Tree (DT) classifier, Random Forest (RF) classifier, AdaBoost classifier, XGBoost classifier, BERT classifier, and ALBERT classifier were trained and tested using the *First wave* dataset and compared in terms of their performance at classifying the relevant and irrelevant tweets. The ALBERT model is chosen as the best model because it provides the least amount of overfitting and the highest accuracy, AUC, and F1-score on the testing set.

In the second part of this thesis, the ALBERT model is applied to the *Longitudinal* and Alzheimer's Awareness Month datasets. The results show that the ALBERT model is successful in classifying the relevant and irrelevant tweets within the context of people impacted by dementia and COVID-19 with an accuracy of 80% for the *Longitudinal* dataset. Moreover, the ALBERT model trained on 10% of the Alzheimer's Awareness Month dataset reached an accuracy of 88% in classifying the relevant and irrelevant tweets related to dementia discourse during Canada's Alzheimer's Awareness Month.

This study showed that the ALBERT model that is tuned and trained on a dataset related to people

impacted by dementia and COVID-19 can be used for future studies with large volumes of data and slightly different features. Accordingly, this model can be trained on a small sample of the labeled dataset and used to predict the label for the rest of the data. This is an efficient way to save time for health care researchers and enable them to focus on the main goal of the project rather than manual labeling.

As a result, this study concludes that choosing the best algorithm for constructing a model depends on the properties of the available data, such as the number of features and the type of input values, whereas choosing the best model from the constructed ones depends on the desired goal. Generally, it is a good idea to start with a simple model, such as the LR classifier, NB classifier, or the DT classifier, and assess the results. After observing and comparing the results, one can use more complex models, such as the RF, the BERT, or the ALBERT classifier, and focus on improving the performance of the model and decreasing the overfitting and underfitting by tuning its parameters.

5.2 Future Work

For future extension of this study, the following research directions are suggested:

- The ALBERT model used in this study was tested using two different datasets. However, we hope that health care researchers use this model in future studies to show the validity and reliability of the trained model.
- It is suggested to use a larger volume of data to reach better performance in classification. Using a greater amount of data can remedy the underfitting and overfitting problem for most of the ML classifiers.
- This study did not explore all possible techniques in terms of classifying relevant and irrelevant tweets for people impacted by dementia and COVID-19. Transfer learning techniques are mainly used for NLP tasks. Therefore, it is of interest to investigate the performance of the other transfer learning algorithms on the same datasets.
- Social media have become an important source of information; however, these online spaces come with types of downsides not present in face-to-face environments. One of the main downsides is the harms of bad bots/inauthentic responses. It is suggested to consider inauthentic responses and try remove them in future studies, which may have a positive impact on the model's performance.
- The purpose of this study was to help health care researchers and social scientists analyze the experiences of people impacted by dementia and COVID-19. The same procedure can be used to analyze social media discourse for future diseases.

References

[1] AlSagri, H. S., and Ykhlef, M. Machine learning-based approach for depression detection in Twitter using content and activity features. In *IEICE Transactions on Information and Systems* (2020), vol. 103, The Institute of Electronics, Information and Communication Engineers, pp. 1825–1832.

[2] Alzubi, J., Nayyar, A., and Kumar, A. Machine learning from theory to algorithms: An overview. In *Journal of physics: conference series* (2018), vol. 1142, IOP Publishing, p. 012012.

[3] Bacsu, J.-D., Cammer, A., Ahmadi, S., Azizi, M., Grewal, K., Green, S., Gowda-Sookochoff, R., Berger, C., Knight, S., Spiteri, R., and O'Connell, M. Examining Twitter discourse on dementia during Alzheimer's Awareness Month in Canada: Infodemiology study (Preprint).

[4] Bacsu, J.-D., O'Connell, M. E., Cammer, A., Azizi, M., Grewal, K., Poole, L., Green, S., Sivananthan, S., and Spiteri, R. J. Using Twitter to understand the COVID-19 experiences of people with dementia: Infodemiology study. In *Journal of Medical Internet Research* (2021), vol. 23, JMIR Publications Inc., Toronto, Canada, p. e26254.

[5] Bacsu, J.-D. R., O'Connell, M. E., Cammer, A., Ahmadi, S., Berger, C., Azizi, M., Gowda-Sookochoff, R., Grewal, K. S., Green, S., Knight, S., et al. Examining the impact of COVID-19 on people with dementia from the perspective of family and friends: Thematic analysis of tweets. In *JMIR aging* (2022), vol. 5, JMIR Publications Inc., Toronto, Canada, p. e38363.

[6] Bashir, D., Montañez, G. D., Sehra, S., Segura, P. S., and Lauw, J. An information-theoretic perspective on overfitting and underfitting. In *Australasian Joint Conference on Artificial Intelligence* (2020), Springer, pp. 347–358.

[7] Bittencourt, H. R., de Oliveira Moraes, D. A., and Haertel, V. A binary decision tree classifier implementing logistic regression as a feature selection and classification method and its comparison with maximum likelihood. In *2007 IEEE international geoscience and remote sensing symposium* (2007), IEEE, pp. 1755–1758.

[8] Bonaccorso, G. *Machine Learning Algorithms: Popular algorithms for data science and machine learning*. Packt Publishing Ltd, 2018.

[9] Bonvillain, N. *Language, culture, and communication: The meaning of messages*. Rowman & Littlefield, 2019.

[10] Braun, V., and Clarke, V. Using thematic analysis in psychology. *Qualitative research in psychology 3*, 2 (2006), 77–101.

[11] Breiman, L. Bagging predictors. In *Machine learning* (1996), vol. 24, Springer, pp. 123–140.

[12] Castleberry, A., and Nolen, A. Thematic analysis of qualitative research data: Is it as easy as it sounds? In *Currents in pharmacy teaching and learning* (2018), vol. 10, Elsevier, pp. 807–815.

[13] Chantar, H., Mafarja, M., Alsawalqah, H., Heidari, A. A., Aljarah, I., and Faris, H. Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification. In *Neural Computing and Applications* (2020), vol. 32, Springer, pp. 12201–12220.

[14] Chayangkoon, N., and Srivihok, A. Text classification model for methamphetamine-related tweets in Southeast Asia using dual data preprocessing techniques. *International Journal of Electrical & Computer Engineering (2088-8708) 11*, 4 (2021).

[15] Che, D., Liu, Q., Rasheed, K., and Tao, X. Decision tree and ensemble learning algorithms with their applications in bioinformatics. In *Software tools and algorithms for biological systems* (2011), Springer, pp. 191–199.

[16] Chen, P.-H. Essential elements of natural language processing: What the radiologist should know. *Academic Radiology 27*, 1 (2020), 6–12.

[17] Chen, T., and Guestrin, C. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (2016), pp. 785–794.

[18] Chiroma, F., Cocea, M., and Liu, H. Detection of suicidal Twitter posts. In *UK Workshop on Computational Intelligence* (2019), Springer, pp. 307–318.

[19] Chiroma, F., Liu, H., and Cocea, M. Suicide related text classification with prism algorithm. In *2018 International Conference on Machine Learning and Cybernetics (ICMLC)* (2018), vol. 2, IEEE, pp. 575–580.

[20] Chowdhury, K. Spam identification on Facebook, Twitter and Email using machine learning. *CERES* (2020), 19.

[21] Cramer, J. S. The origins of logistic regression. Tinbergen Institute Working Paper.

[22] Deepa, M. D., et al. Bidirectional Encoder Representations from Transformers (bert) language model for sentiment analysis task. *Turkish Journal of Computer and Mathematics Education (TURCOMAT) 12*, 7 (2021), 1708–1721.

[23] Dong, X., Yu, Z., Cao, W., Shi, Y., and Ma, Q. A survey on ensemble learning. In *Frontiers of Computer Science* (2020), vol. 14, Springer, pp. 241–258.

[24] Etaiwi, W., and Naymat, G. The impact of applying different preprocessing steps on review spam detection. In *Procedia computer science* (2017), vol. 113, Elsevier, pp. 273–279.

[25] Faraway, J. J. Does data splitting improve prediction? In *Statistics and computing* (2016), vol. 26, Springer, pp. 49–60.

[26] Feurer, M., and Hutter, F. Hyperparameter optimization. In *Automated machine learning*. 2019, pp. 3–33.

[27] Gao, Z., Feng, A., Song, X., and Wu, X. Target-dependent sentiment classification with BERT. In *Ieee Access* (2019), vol. 7, IEEE, pp. 154290–154299.

[28] González-Carvajal, S., and Garrido-Merchán, E. C. Comparing BERT against traditional machine learning text classification. *arXiv preprint arXiv:2005.13012* (2020).

[29] Hofmann, T., Schölkopf, B., and Smola, A. J. Kernel methods in machine learning. In *The annals of statistics* (2008), vol. 36, Institute of Mathematical Statistics, pp. 1171–1220.

[30] Huang, Y., and Li, L. Naive bayes classification algorithm based on small sample set. In *2011 IEEE International conference on cloud computing and intelligence systems* (2011), IEEE, pp. 34–39.

[31] James, G., Witten, D., Hastie, T., and Tibshirani, R. *An introduction to statistical learning*, vol. 112. Springer, 2013.

[32] Jayaraman, S., Choudhury, T., and Kumar, P. Analysis of classification models based on cuisine prediction using machine learning. In *2017 international conference on smart technologies for smart nation (SmartTechCon)* (2017), IEEE, pp. 1485–1490.

[33] Jiang, L., Wang, S., Li, C., and Zhang, L. Structure extended multinomial naive bayes. In *Information Sciences* (2016), vol. 329, Elsevier, pp. 346–356.

[34] Jiao, Y., and Du, P. Performance measures in evaluating machine learning based bioinformatics predictors for classifications. In *Quantitative Biology* (2016), vol. 4, Springer, pp. 320–330.

[35] Johns, B. T., and Jamieson, R. K. A large-scale analysis of variance in written language. In *Cognitive Science* (2018), vol. 42, Wiley Online Library, pp. 1360–1374.

[36] Joseph, S. R., Hlomani, H., Letsholo, K., Kaniwa, F., and Sedimo, K. Natural language processing: A review. *International Journal of Research in Engineering and Applied Sciences 6*, 3 (2016), 207–210.

[37] Kenton, J. D. M.-W. C., and Toutanova, L. K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT* (2019), pp. 4171–4186.

[38] Kibriya, A. M., Frank, E., Pfahringer, B., and Holmes, G. Multinomial naive bayes for text categorization revisited. In *Australasian Joint Conference on Artificial Intelligence* (2004), Springer, pp. 488–499.

[39] Kici, D., Malik, G., Cevik, M., Parikh, D., and Başar, A. A BERT-based transfer learning approach to text classification on software requirements specifications. *The 34th Canadian Conference on Artificial Intelligence* (2021), 1–13.

[40] Kingsford, C., and Salzberg, S. L. What are decision trees? In *Nature biotechnology* (2008), vol. 26, Nature Publishing Group, pp. 1011–1013.

[41] Koehrsen, W. Overfitting vs. underfitting: A complete example. *Towards Data Science* (2018).

[42] Kononenko, I., and Kukar, M. *Machine learning and data mining*. Horwood Publishing, 2007.

[43] Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., and Brown, D. Text classification algorithms: A survey. In *Information* (2019), vol. 10, Multidisciplinary Digital Publishing Institute, p. 150.

[44] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. Albert: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942* (2019).

[45] Liashchynskyi, P., and Liashchynskyi, P. Grid search, random search, genetic algorithm: A big comparison for NAS. *arXiv preprint arXiv:1912.06059* (2019).

[46] Lindeza, P., Rodrigues, M., Costa, J., Guerreiro, M., and Rosa, M. M. Impact of dementia on informal care: a systematic review of family caregivers' perceptions. In *BMJ Supportive & Palliative Care* (2020), British Medical Journal Publishing Group.

[47] Manisha, M., Kodali, A., and Srilakshmi, V. Machine classification for suicide ideation detection on Twitter. *International Journal of Innovative Technology and Exploring Engineering 8*, 12 (2019), 4154–60.

[48] Masterson-Algar, P., Allen, M. C., Hyde, M., Keating, N., and Windle, G. Exploring the impact of COVID-19 on the care and quality of life of people with dementia and their carers: a scoping review. In *Dementia* (2022), vol. 21, SAGE Publications Sage UK: London, England, pp. 648–676.

[49] Mitchell, T. M., et al. Machine learning. McGraw-hill New York.

[50] Müller, A. C., and Guido, S. *Introduction to machine learning with Python: a guide for data scientists*. O'Reilly Media, Inc., 2016.

[51] Nichols, J. A., Herbert Chan, H. W., and Baker, M. A. Machine learning: applications of artificial intelligence to imaging and diagnosis. In *Biophysical reviews* (2019), vol. 11, Springer, pp. 111–118.

[52] Olaleye, T., Abayomi-Alli, A., Adesemowo, K., Arogundade, O. T., Misra, S., and Kose, U. SCLAVOEM: hyper parameter optimization approach to predictive modelling of COVID-19 infodemic tweets using smote and classifier vote ensemble. In *Soft Computing* (2022), Springer, pp. 1–20.

[53] Oscar, N., Fox, P. A., Croucher, R., Wernick, R., Keune, J., and Hooker, K. Machine learning, sentiment analysis, and tweets: An examination of Alzheimer's disease stigma on Twitter. In *Journals of Gerontology Series B: Psychological Sciences and Social Sciences* (2017), vol. 72, Oxford University Press US, pp. 742–751.

[54] Otter, D. W., Medina, J. R., and Kalita, J. K. A survey of the usages of deep learning for natural language processing. In *IEEE transactions on neural networks and learning systems* (2020), vol. 32, IEEE, pp. 604–624.

[55] Quinlan, J. R. Induction of decision trees. In *Machine learning* (1986), vol. 1, Springer, pp. 81–106.

[56] Raileanu, L. E., and Stoffel, K. Theoretical comparison between the gini index and information gain criteria. In *Annals of Mathematics and Artificial Intelligence* (2004), vol. 41, Springer, pp. 77–93.

[57] Ramalingaiah, A., Hussaini, S., and Chaudhari, S. Twitter bot detection using supervised machine learning. In *Journal of Physics: Conference Series* (2021), vol. 1950, IOP Publishing, p. 012006.

[58] Ramos, J., et al. Using TF-IDF to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning* (2003), vol. 242, Citeseer, pp. 29–48.

[59] Ray, S. A quick review of machine learning algorithms. In *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)* (2019), IEEE, pp. 35–39.

[60] Reitermanova, Z., et al. Data splitting. *WDS 10* (2010), 31–36.

[61] Rogers, A., Kovaleva, O., and Rumshisky, A. A primer in BERTology: What we know about how BERT works. In *Transactions of the Association for Computational Linguistics* (2020), vol. 8, MIT Press, pp. 842–866.

[62] Sagi, O., and Rokach, L. Ensemble learning: A survey. In *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* (2018), vol. 8, Wiley Online Library, p. e1249.

[63] Samuel, A. L. Some studies in machine learning using the game of checkers. In *IBM Journal of research and development* (1967), vol. 11, IBM, pp. 601–617.

[64] Schapire, R. E. The boosting approach to machine learning: An overview. In *Nonlinear estimation and classification* (2003), Springer, pp. 149–171.

[65] Schomacker, T., and Tropmann-Frick, M. Language representation models: An overview. In *Entropy* (2021), vol. 23, Multidisciplinary Digital Publishing Institute, p. 1422.

[66] Siddiqui, T., and Alam, M. A. Discovery of fuzzy censored production rules from large set of discovered fuzzy if then rules. In *Proceedings of World Academy of Science, Engineering and Technology [PWASET]* (2009), vol. 37, Citeseer, pp. 2070–3740.

[67] Siddiqui, T., et al. Sarcasm detection from Twitter database using text mining algorithms. *Turkish Journal of Computer and Mathematics Education (TURCOMAT) 12*, 11 (2021), 1916–1924.

[68] Singh, A. K., and Shashi, M. Vectorization of text documents for identifying unifiable news articles. *Int. J. Adv. Comput. Sci. Appl 10*, 7 (2019).

[69] Slimi, H., Bounhas, I., and Slimani, Y. Adapting pre-trained language models to rumor detection on Twitter. In *Journal of Universal Computer Science* (2021), vol. 27, GRAZ UNIV TECHNOLGOY, INST INFORMATION SYSTEMS COMPUTER MEDIA-IICM, pp. 1128–1148.

[70] Suarez-Gonzalez, A., Livingston, G., Low, LFand Cahill, S., Hennelly, N., Dawson, W., Weidner, W., Bocchetta, M., Ferri, C., Matias-Guiu, J., Alladi, S., Musyimi, C., and Comas-Herrera, A. Impact and mortality of COVID-19 on people living with dementia: Cross-country report. In *International Long-Term Care Policy Network, CPEC-LSE* (2020), LTCcovid.org.

[71] Suleymanov, U., Kalejahi, B. K., Amrahov, E., and Badirkhanli, R. Text classification for Azerbaijani language using machine learning. *Comput. Syst. Sci. Eng. 35*, 6 (2020), 467–475.

[72] Taunk, K., De, S., Verma, S., and Swetapadma, A. A brief review of nearest neighbor algorithm for learning and classification. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)* (2019), IEEE, pp. 1255–1260.

[73] Twint. Twint project. https://github.com/twintproject/twint, 2021.

[74] Vijayarani, S., Ilamathi, M. J., Nithya, M., et al. Preprocessing techniques for text mining: An overview. *International Journal of Computer Science & Communication Networks 5*, 1 (2015), 7–16.

[75] Webster, J. J., and Kit, C. Tokenization as the initial phase in NLP. In *COLING 1992 Volume 4: The 14th International Conference on Computational Linguistics* (1992).

[76] Wu, H., Cao, Y., Wei, H., and Tian, Z. Face recognition based on haar like and euclidean distance. In *Journal of Physics: Conference Series* (2021), vol. 1813, IOP Publishing, p. 012036.

[77] Xiang, X., Lu, X., Halavanau, A., Xue, J., Sun, Y., Lai, P. H. L., and Wu, Z. Modern senicide in the face of a pandemic: An examination of public discourse and sentiment about older adults and COVID-19 using machine learning. In *The Journals of Gerontology: Series B* (2021), vol. 76, Oxford University Press US, pp. e190–e200.

[78] Ying, X. An overview of overfitting and its solutions. In *Journal of physics: Conference series* (2019), vol. 1168, IOP Publishing, p. 022022.

[79] Zahera, H. M., Elgendy, I. A., Jalota, R., and Sherif, M. A. Fine-tuned BERT model for multi-label tweets classification. In *TREC* (2019), pp. 1–7.

[80] Zeng, W., Gautam, A., and Huson, D. H. On the application of advanced machine learning methods to analyze enhanced, multimodal data from persons infected with COVID-19. In *Computation* (2021), vol. 9, Multidisciplinary Digital Publishing Institute, p. 4.

[81] Zeroual, I., and Lakhouaja, A. Data science in light of natural language processing: An overview. In *Procedia Computer Science* (2018), vol. 127, Elsevier, pp. 82–91.

[82] Zhang, Y., Zhou, Y., and Yao, J. Feature extraction with TF-IDF and game-theoretic shadowed sets. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems* (2020), Springer, pp. 722–733.

[83] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. A comprehensive survey on transfer learning. In *Proceedings of the IEEE* (2020), vol. 109, IEEE, pp. 43–76.