REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

DIAGNOSIS OF DISEASES THROUGH GENESBY USING MACHINE LEARNING

Israa Jihad Abed ABED

Master`s Thesis

Supervisor

Asst. Prof. Dr. Oguz KARAN

Istanbul, 2023

DIAGNOSIS OF DISEASES THROUGH GENES BY USING MACHINE LEARNING

Israa Jihad Abed ABED

Information Technologies

Master`s Thesis

ALTINBAŞ UNIVERSITY

2023

The thesis titled Development of DIAGNOSIS OF DISEASES THROUGH GENES BY USING MACHINE LEARNING prepared by ISRAA JİHAD ABED ABED and submitted on 11/04/2023 has been **accepted unanimously** for the degree of Master of Science in Information Technologies.

____

Asst. Prof. Dr. Oguz KARAN

Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. . Oguz KARAN	Department of Electrical and Electronics Engineering, Altınbaş University	__________________
Asst. Prof. Dr. Abdullahi abdu IBRAHIM	Department of Computer Engineering, Altınbaş University	__________________
Assoc. Prof. Dr. Izzet paruğ DURU	Department of Medical Imaging Techniques, Istanbul Gedik University	__________________

I hereby declare that this thesis meets all format and submission requirements of a Master's thesis.

Submission date of the thesis to the Institute of Graduate Studies: ___/___/___

I hereby declare that all information presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Israa Jihad Abed ABED

Signature

ABSTRACT

DIAGNOSIS OF DISEASES THROUGH GENES BY USING MACHINE LEARNING

ABED, Israa Jihad Abed

M.Sc., Information Technologies, Altınbaş University,

Supervisor: Asst. Prof. Dr. Oguz KARAN

Date: April / 2023

Pages: 72

The Internet is a vast and ever-expanding source of texts, from casual posts to in-depth talks, with a wide range of perspectives and ends in mind. Attempts at polarity recognition and target extraction are time-consuming because it's important to figure out how the object is oriented and where it's going. The implications for national defence and the study of politics are profound. Some research has been done, however, on a subject that is peripheral to the present discussion. Therefore, the primary goal of this research is to identify the optimal solution, particularly with regards to Arabic political articles, for which there is currently no dataset. The need to reevaluate traditional approaches to extracting sentiment from texts has arisen in response to the growing complexity of both text sources and subjects. This dissertation's overarching goal is to help the government make better choices through the use of an opinion-mining-based support system. In particular, the research focuses on political Arabic articles and the classification method used to develop a new decision-making technique. This dissertation also suggested a new framework for detecting polarity and targets in published political articles using a combination of lexicon-based techniques, machine learning, and a hybrid approach. As part of the research, four distinct corpora were constructed (V1, V2, V3, and V4). The Mandalay archive houses these compiled corpora from various online sources. In order to determine the polarity of papers, this research utilised machine learning and a hybrid approach. Support Vector Machine (SVM), Naive

Bayes (NB), KNearest Neighbor (KNN), and Decision Tree (DT) are included in the first group, as are the two feature extractions, Bundle of Words (BOW), and n-grams. The latter strategy combined a lexicon-based vector with a classification based on Rough Set Theory (RST). In this research, the human-based seed word was used in conjunction with lower approximation, a sentence target extraction technique. The research improved target sentence extraction by combining the lower approximation method with sentence frequency. Guaranteeing the desired accuracy and verifying sentence extraction were both tested with human and non-human assessments. The KNN technique was found to be superior to the other machine learning techniques tested. Approximation accuracy with seed-vector and ensemble-vector hybrid methods was higher than with lower estimate methods. Hybrid approaches were found to be more effective than machine learning techniques. The performance of the lower approximation was worse than that of the lower estimate plus sentence frequency. According to the findings, the best approach to determining the polarity of articles is an accurate approximation technique based on an ensemble vector. Furthermore, it is suggested that a lower approximation of sentence frequency be used to extract the subjects of articles.

Keywords: KNN, Rough Set Theory, Bundle of Words, Naive Bayes, Support Vector Machine, SVM.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

DT	:	Decision Tree
kNN	:	k-Nearest-Neighbors
LR	:	Logistic Regression
ML	:	Machine Learning
NB	:	Naïve Bayes
ROC	:	Receiver Operating Characteristic
RT	:	Random Tree
SVM	:	Support Vector Machine
TF	:	Term Frequency

1. INTRODUCTION

1.1 RESEARCH BACKGROUND

There has been a dramatic increase in the amount of biomedical research that has been published and, hence, in the size of the underlying body of biomedical knowledge. Text mining and knowledge extraction are a couple of methods that can help academics handle this deluge of data. Applying text mining has yielded significant results in the areas of named entity recognition, text categorization, terminology extraction, relationship extraction, and hypothesis creation. Multiple laboratories are building versatile text-mining systems with a wide range of applications (Hahn, Cohen et al. 2012). Making these technologies helpful to biomedical researchers will be the primary issue of biomedical text mining during the next 5–10 years. To accomplish this, we will need to improve access to full text; our knowledge of the feature space of scientific literature; our ability to quantify the value of systems; and our willingness to work together with the biomedical research community to meet their demands (Rebholz-Schuhmann, Oellrich et al. 2012).

There are still many problems to be solved in biomedical text mining, such as using the retrieved information to answer specific research questions in translational research and identifying complex links between items of biomedical relevance. This is especially important for those studying human diseases who are having trouble keeping up with the plethora of publications in their field. The gap between the laboratory and the clinic can be narrowed with the aid of text mining techniques that assist in extracting useful information from massive amounts of data stored in document archives (Cases, Furlong et al. 2013).

Due to the availability of corpora and the pressure driven by specialised text mining issues, previous work in relationship text mining has primarily focused on the detection of interactions between proteins (Rebholz-Schuhmann, Oellrich et al. 2012). Whereas the identification of linkages between diseases, medications, genes, and their sequence variants has received considerable attention, the identification of relationships between things of biomedical significance has received less focus. The tide has turned, however, and there is now a great deal more enthusiasm for collecting this data than there was in the past.

Protected information might be tangible such as, (paperwork) or electronic information (like data Computer Storage) in any format (e.g., knowledge) Figure 1.2 shows the primary objective of information security is to ensure the three tenets of CIA triad confidentiality, integrity, and Availability of data.

Artificial Intelligence (AI) tools like machine learning, deep learning, and natural language processing (NLP) tools are very important to many studies because they can be used to make systems that can accurately extract information about human diseases and their genes. These systems can be used on large document repositories and can combine the extracted data with other data to help with further analysis and knowledge discovery (Arighi, Wu et al. 2014). But it is very hard to figure out all of the possible relationships using wet experimental methods, which are too expensive to be a good option right now. But relying on people to do the relation extraction (RE) task is still hard work and takes a lot of time.

1.2 PROBLEM STATEMENT

Diseases and genes are closely linked, which means that changes in genes may cause some diseases. Biology has been trying for a long time to find disease genes, which could help improve health care and help us understand how genes work and how they interact with each other. But there aren't many large-scale gene-gene association datasets, disease-disease association datasets, or gene-disease association datasets. The links between genes and diseases are very important for preventing, diagnosing, and treating diseases. Even though the connections between genes and diseases have been studied a lot, we still don't know a lot about how they work. So, finding diseases and genes by hand is expensive, hard, takes a long time, and requires skilled professionals. Also, finding links between genes and diseases by hand is a time-consuming job.

With the recent progress in machine learning and deep learning, different methods have been put forward to solve this problem, and the results have been outstanding. It is important to come up with ways to use computers that will make it easier to find and classify genes and diseases.

1.3 AIM OF THE THESIS

The main goal of this thesis is to design and implement a high-accuracy model for finding and classifying gene-disease links based on text mining, natural language processing, and deep learning. The main goals of this study are the following:

a. To get a large number of articles from different sources and use efficient pre-processing for the English language to get rid of unwanted words and normalize them.
b. To use SVM, NB, DT, and KNN methods to figure out how genes and diseases are related.
c. Compare the suggested model to the models used in other studies to see how well it fits in with the other work.
d. Evaluate the successful model with a high rate of performance accuracy, loss function, confusion matrix measurements, and F-score.

2. LITERATURE REVIEW

2.1 RELATED WORK

The amount of work published in the field of medicine is growing exponentially. Over 14 million articles have been added to PubMed's database. Curators have a tough time sifting through biomedical literature to find relevant material and organizing it for publication. Accordingly, the information covered by the curated databases is extremely limited. As a result, it has become increasingly difficult to glean useful information from the massive body of biomedical literature. Co-occurrence frequencies of genes and disorders have been the basis of the majority of prior text mining studies used to extract gene-disease relationships from the biomedical literature (Chun, Tsuruoka et al., 2006) illustrates a programme for mining MEDLINE for disease-gene associations. From six different public databases, they compiled a vocabulary of disease and gene names, which they then used to extract relation candidates via dictionary matching. Due to the high number of false positives generated by dictionary matching, we devised a method of machine learning-based named entity recognition (NER) to eliminate these erroneous diagnoses. The study concluded that NER filtering has a decisive effect on relation extraction performance and that filtering improves relation extraction precision by 26.7% at the cost of a slight drop in recall.

(Ravikumar, Wagholikar, et al., 2015) made and tested a text mining system called MutD that finds links between protein mutations and diseases in MEDLINE abstracts by using discourse level analysis and a benchmark dataset made up of carefully selected database records. MutD gets an F-measure of 64.3% when it tries to figure out how protein mutations and diseases are linked from records in a curated database. When the sentence-level association extraction was compared to the discourse-level analysis part of MutD, the F-measure went up by more than 10%. Mahmood, Wu, et al. (2016) developed a text-mining system called DiMeX to find links between mutations and diseases in publication abstracts. DiMeX is made up of a set of natural language processing modules. These modules take in text and use syntactic and semantic patterns to find connections between mutations and diseases. When tested on three different datasets to find links between mutations and diseases, DiMeX gets F-scores of 0.88, 0.91, and 0.89, which are all high. DiMeX has a separate part that pulls out mutations that are mentioned in the text and connects them to genes. DiMeX was used on a large set of abstracts from Medline to find links between mutations and diseases, as well as other useful information, such as the size of the patient/cohort and the size of the population.

(Bravo, Piñero et al. 2015) BeFree can find the best matches between genes and diseases, drugs and diseases, and drugs and their targets. It shows the value of the gene-disease associations that BeFree found by using a number of analyses and combining them with data from other sources. BeFree is a new text mining system that works well to find links between genes and diseases, drugs and diseases, and drugs and their targets. The analyses show that mining only a small part of MEDLINE produces a large dataset of gene-disease associations. But only 2% of this dataset is recorded in curated resources, which raises a lot of questions about how to priorities and organize data. (Xu, Zhang et al., 2016) introduced a reliable and efficient framework that takes large biomedical literature repositories as input, finds credible relationships between diseases and genes, and shows possible genes related to a given disease and possible diseases related to a given gene. The framework includes name entity recognition (NER), which looks for genes and diseases in text; association detection, in which we extract and evaluate features from pairs of genes and diseases; and ranking algorithms that estimate how closely the pairs are related. The NER phase has an F1-score of 0.87, which is higher than any other study. The association detection phase takes a lot less time than it did in the past, but the F1-score is still the same, at 0.86. For the top 50 genes linked to a disease, the end-to-end result gets a 0.259 F1-score, which is better than what was done before.

A novel network-based method, SLN-SRW, was proposed by Peng et al. (2017) to determine the weight of edges in an integrated network and then use this network for the prediction of gene-disease connections. SLN-strengths SRW's lie in its ability to estimate edge weight by classifying a variety of edge-types and in its use of a Laplacian normalization-based approach to evade the bias introduced by a few particularly influential nodes inside a network.

As a first step in developing a powerful search system geared at locating biomarker evidence in the literature, (Thompson and Ananiadou 2017) describes the TM method for the detection of sentence-level connections between genes and diseases. It uses either co-occurring mentions of genes and diseases or linguistic patterns obtained using data from roughly 1 million biomedical abstracts, and the level of intricacy of the detection process varies according to sentence complexity. The results show that the strategy outperforms using a single technique for association detection and compares very favorably to other relevant efforts in terms of accuracy. Moreover, it demonstrates that the observed relations can supplement those detected by other methods.

The proposed work (Bhasuran and Natarajan, 2018) uses a supervised machine learning approach to train an ensemble support vector machine to extract gene-disease relations from four gold standard corpora. The feature set covers conceptual, syntax, and semantic properties jointly learned with word embedding. The presented novel approach, which combines a rich syntax and semantic feature set with domain-specific word embedding via ensemble support vector machines and was evaluated on four gold standard corpora, is thought to be a new baseline for future work in gene-disease relation extraction from literature.

(Xing, Qi, et al., 2018) proposed a pipeline for retrieving phenotypes, genes, and their associations from medical literature. Improving the unsupervised word-embedding-to-sentence-embedding cascaded technique as representation learning, along with abbreviation revision and sentence template extraction, allows for better recognition of different types of broad phenotypic information in the literature. Additional work on gene recognition used a dictionary and a rule-based approach. Finally, it used one of the best-known information extraction systems, OLLIE, to discover the genetic basis for observable traits.

(Zhou and Fu, 2018) presented an MLM for anticipating possible gene-disease links by mining text documents on genes or diseases, and then compared the results of their prediction to those obtained using a different technique. Its prediction method yielded quantitative results, quantifying hypothesised relationships between genes and diseases. Cosine similarity between gene vectors and illness vectors is used in a unique approach that incorporates the MeSH database, word weight (TW), and co-occurrence algorithms to predict gene-disease relationships. The occurrence and position of genes or illness phrases in the text of publications in the PubMed database are used to generate vectors. During the vectorization procedure, the disease-related text data was improved. The method's originality originates from fusing text mining with a graphical model. That's why this approach has the potential to open up fresh paths toward learning more about the connections between genes and diseases, enhancing the accuracy of predictions and bringing together different popular graphical models.

(Wu, Luo, et al., 2019) came up with a new and effective way to find links between genes and diseases and put it into practise with a tool called RENET. RENET does a much better job than other tools because it uses deep learning to find important classification signals across the whole document. Other tools analyse sentences one by one using SVM or rule-based methods. It learns a representation of a document in two steps: (1) from a representation of a word to a representation of a sentence using CNN; and (2) from a representation of a sentence to a representation of a document using an RNN-based method. It has done a lot of tests on a big set of data that came from DisGeNet. Traditional sentence-level relation extraction methods are much worse than RENET (more than 20% worse in F-Score).

(Thillaisundaram and Togia, 2019) describes a system that can learn contextual language representations from a large unlabeled corpus and whose parameters can be fine-tuned to solve specific tasks with little extra architecture. It can store the two mentions and the text around them as two sequences that come one after the other and are separated by a special symbol. Then, it uses a single linear layer to divide their relationship into five classes: "no relation," "one relationship," "two relationships," and "three relationships." Even though there is a big difference between the classes, the system does a lot better than a random baseline, despite having a very simple setup and no special features.

To better extract gene-disease connections from full-text publications, Su, Wu et al. (2021) introduced RENET2, a deep learning-based relation extraction method that uses section filtering and ambiguous relations modelling. To address the dearth of labels on full-text articles, it developed a creative iterative training data expansion technique to construct an annotated full-text dataset. RENET2's F1-score of 72.13% is 27.22%, 30.30%, and 29.24% better than the state-of-the-art programmes BeFree, DTMiner, and BioBERT when it comes to extracting gene-disease connections from an annotated full-text dataset.

A framework was presented to investigate the link between bacteria and diseases (Park, Lee et al., 2021). This structure utilises machine learning and natural language processing techniques. For this purpose, we employed a hierarchical long-short-term memory network to identify sentences that adequately characterise the connection. Using a hybrid of two parse tree-based search strategies, we identified the relation-describing word in the determined sentences. In order to increase the accuracy of predictions, an ensemble model was developed using constituency parsing for structural pattern matching and dependency-based relation extraction. The extraction of the microbe-disease relationship was improved by our suggested framework, which combined deep learning and parse tree-based methods.

3. THEORETICAL BACKGROUND

3.1 INTRODUCTION

In Chapter 3, we lay out the theoretical framework and context for this thesis. Text mining (TM) is the primary focus of this study; subsequent sections will include the corpus and preprocessing phases, classification, and a description of the most popular classification techniques. All of these approaches will be talked about in detail, including lexicon-based, hybrid, and machine learning approaches.

3.2 BIOINFORMATICS

Essentially, bioinformatics is the study of how computers work, what they can do, and how they might be improved. In Chapter 3, we lay out the theoretical framework and context for this thesis. Text mining (TM) is the primary focus of this study; subsequent sections will include the corpus and preprocessing phases, classification, and a description of the most popular classification techniques. All of these approaches will be talked about in detail, including lexicon-based, hybrid, and machine learning approaches.

Tools for problem-solving and hypothesis-building are readily available across the biological sciences. It's a brand-new discipline that uses the tools of computing, mathematics, and IT to decipher and understand genetic information. [1]. By automating the study of data sets that are too huge or complicated for human review, bioinformatics has facilitated progress in the biological sciences. Gene finding algorithms, microarray expression analysis, molecular systems modeling, and the identification of biomarkers in mass spectra are all examples of such methods. At the molecular, cellular, and systems levels, computational tools are indispensable for organizing, analyzing, and extracting biological data. As a result, the demand for experts fluent in both biological and computational jargon continues to rise. Statistics and algorithms might not be familiar to a biologist who went to a more traditional school, and chemistry and biology might not be familiar to a computer scientist who went to a more traditional school.

Bioinformatics has helped advance biology by providing methods for evaluating data sets too large or complex to be handled by hand. Gene identification strategies, microarray expression analysis, molecular systems modeling, and the identification of biomarkers in mass spectra are all examples of such methods. Computational tools are indispensable for the organization, analysis, and extraction of biological data at the level of macromolecules, cells, and systems. Therefore, there is a growing need for capable people with backgrounds in both biology and computer science. Computer scientists and biologists may lack the requisite understanding of chemistry and biology to develop traditional programs [2].

3.3 TEXT MINING

Text data mining is also known as text mining [3] or text databases for kdd. Important text is extracted from unstructured material using data mining algorithms (natural language) [4, 5], [6, 7]. The main content is presented in an unstructured fashion, making text manipulation difficult. Therefore, it is essential to be able to extract useful data from texts written in natural languages [7]. While unstructured language may be simple for people to understand, it presents significant challenges for computers. So, it's clear that this collection of texts is a valuable source of information and knowledge [8].

Data mining (DM), also known as "knowledge discovery," is a method of sifting through and analyzing massive amounts of data in order to find patterns or insights. It is possible to foresee future trends and behaviors using data mining technologies. Use cases include companies taking preventative action based on solid information. [9–11]. The system's intended function specifies the knowledge discovery objectives to be met. One can distinguish between verification and discovery aims. In the case of verification, the system is restricted to testing the user's hypothesis, a task most commonly encountered in statistical analysis [12, 13]. The system learns to recognize novel patterns on its own with the help of knowledge discovery. [14, 15]. Sub-goals of discovery include prediction, in which the system finds patterns to anticipate the future behavior of particular entities; and descriptions, in which the system finds patterns to communicate with a user in a human-understandable manner. [16].

With the help of text mining, a chaotic document can be reorganized into a more manageable form. Processes like preprocessing, feature extraction, and text mining are commonly used to structure the input text, derive patterns from the resulting structured data, evaluate the outcome, and draw conclusions. Relevance, originality, and interest in the mined content are all indicators of high quality in text mining. [17]. Here are the several steps involved in text mining.

3.3.1 Text Mining Phases

It's challenging work to mine text from a large corpus of documents. To avoid this problem, the dataset must be organized in a way that makes further document analysis possible. [8]. There are several steps involved in text mining. [18, 19] The several stages of text mining are depicted in Figure 3.1, and explanations of each stage may be found below.

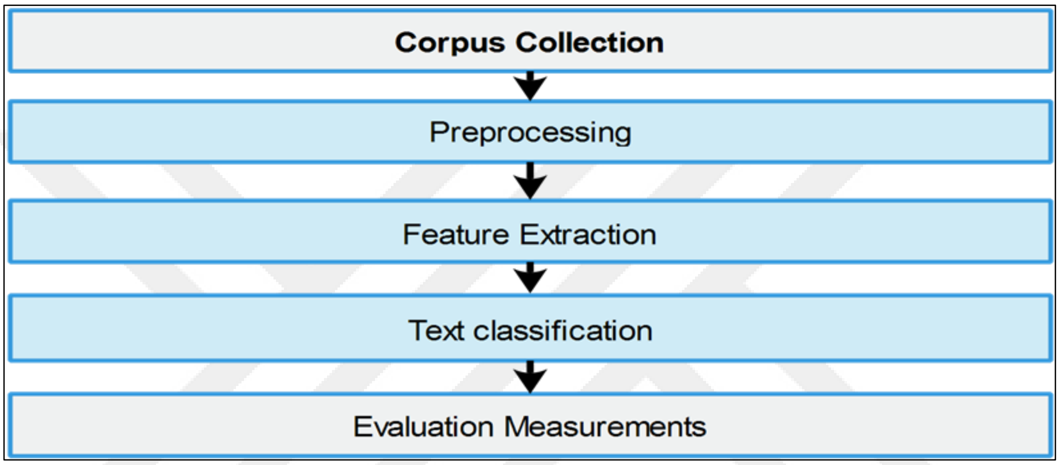

Figure 3.1: Text Mining Phases.

a. Corpus collection: During this stage, information is gathered in order to construct a corpus of the target language, which may be English, Arabic, or some other language.
b. Preprocessing: In this stage, you will learn about the many methods for working with the Arabic language. These methods include things like stemming, normalizing text, and removing stop words.
c. Feature extraction: At this stage, all the hallmarks of a pedagogically effective work are there.
d. Text classification: The majority of text classification methods are introduced at this stage.
e. Evaluation measurements: Accuracy, precision, recall, and F-score are just few of the many evaluation factors that can be used to assess a text classifier's efficacy. There will be an emphasis on visual analytics, including precision-recall, ROC, principal component analysis, and t-TNS.

3.4 PREPROCESSING

The success or failure of a word embedding model relies heavily on the quality of the text preparation phase. [20, 21]. For the purposes of textual targeting and polarity, it is used exclusively. It tries to turn a group of articles into a good text by using many methods, such as stemming, removing stop words, normalization, and tokenization.

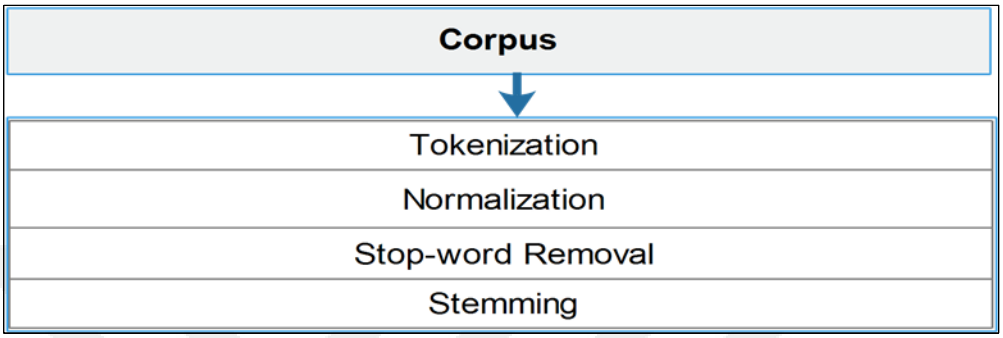

Figure 3.2: Pre-Processing Steps.

Text preprocessing makes use of many different methods to break down unstructured documents, such as legal documents, into standardized sequences of linguistic parts. The benefits of preprocessing procedures include [24]:

a. Reducing the number of frequently used terms improves the accuracy of the search results.

b. Condensing lengthy explanations into a few important terms allows for more rapid decision-making.

3.4.1 Tokenization

Through a process called tokenization, a piece of writing is reduced to its component parts, typically words or sentences. [25]. Common methods of separating ideas in writing include commas, spaces, quotation marks, semicolons, and periods. [26]. One-word tokens (verbs, nouns, conjunctions, pronouns, prepositions, numbers, articles, punctuation, and alphabetics) are collected and organised into a list without regard to their contextual relationships or implied meanings. Paragraphs, sentences, phrases, and individual words are the standard units of textual organization. Tokenization, often known as text segmentation, is the primary function of natural language processors. [27]. The tokens that are made can then be used for more work, like stemming, normalization, or getting rid of stop words.

3.4.2 Normalization

Normalizing texts is a necessary process that should never be skipped. Normalization is the process of converting an unidentified letter form used in literature to a standard form. All non-English letters, punctuation marks, and numbers must be removed before normalisation may begin. [28]. In order to transform conversational text from an input into a standard output form, a technique is used to normalise texts by removing noises like dates, abbreviations, diacritics, and acronyms. [29, 30].

3.4.3 Stop Word Removal

Stop word removal, a common preprocessing technique, cleans data of irrelevant words and phrases by reformatting them into more usable formats. The text's meaning would not be altered by omitting this irrelevant detail. [31]. There are several different categories that stop-words can be placed into. Included in [32] are adverbs, metric system units, names of coins, interrogative and conditional pronouns, prepositions, pronouns, reference names/determinants, relative and verbal pronouns, and transformers (verbs, letters). Stop words can account for as much as 30–50% of all tokens in a large text collection; removing them from an article makes it longer and enhances its indexing performance. [33]. There is no exhaustive list of stop words in the English language. [32]. The NLTK library [34] is just one example of a tool that can be used to declare a list of stop words.

3.4.4 Stemming

The primary goal of stemming is to reduce complex derived and inflected words to their shortest possible forms by removing affixes and suffixes. The stemming procedure is essential to many tasks that might be categorised as text preparation. There are several uses for computers in the field of text analysis, including classification [35], summarization [36], retrieval [38], etc. Many languages, including English, have had various stemming methods proposed for them. French (40), Spanish (41), Hindi (42), and Malay (43). Every natural language has extra derivational morphemes, like affixes and suffixes, that can change the meaning of words.

3.5 FEATURE EXTRACTION

The BOW model is a popular method of representation for classifying texts. [44] and make vectors out of word documents [45]. Traditionally, BOW is used in the Vector Space Model (VSM) to provide a suitable weight for each token. [46, 47]. BOW is a method for extracting characteristics from written material. These features can be used for machine learning systems' training. It creates a dictionary of all the singular words that exist in the training set documents. It's just a bunch of words thrown together to make a point, and the sequence doesn't really matter. Words with more importance get higher weights, while keywords with less importance get lower weights. The Basic Outline of Words (BOW) method is a tool for gathering information from written materials. These features can be used for the training of machine learning algorithms. It creates a dictionary consisting of all the rare words that show up in every document in the training set. Simply said, it's a string of words that can be read as a whole, even if the order in which they appear doesn't matter much. It gives less value to less important words and more to more important ones. [48, 49]. In the final and fourth stage, the labelled vector is obtained. Word vectors will then be used to determine each word's TF and TF-IDF. n is the total number of samples in the corpus, count(w) and count (w n) are the counts of occurrences of w in the corpus, and TF(w) is the document frequency of w.

$$TF(w) = \frac{count(w)}{\sum count(w_n)} \tag{3.1}$$

In the equation (3-2), IDF(w) stands for the inverse file frequency of the word w, df stands for the number of examples in the corpus that contain the word w, and N stands for the total number of texts contained in the corpus. [1].

$$IDF(w) = log\left(\frac{N}{df}\right) \tag{3.2}$$

Estimating the products of the TF-IDF interaction is the goal of this method. Calculating the TF-IDE using equation (3- 3) in accordance with equation (3- 1) and equation (3- 2).

$$TF - IDF(w) = \mathrm{TF} * \mathrm{IDF} \tag{3.3}$$

The BOW model is advantageous since it is straightforward to learn and put into practice, can be tailored to a wide variety of different texts, and can be used to quantify the significance of individual words. Classification and language modelling are just two examples of the types of predictive challenges where BOW has shown useful [27]. There are, however, some disadvantages to employing BOW. One of these is the loss of semantic meaning. [51]). The basic BOW technique completely disregards each word's intended meaning (disregarding word order: same words can be used multiple times in context or nearby words). As a second thought, consider the length of the vector. When dealing with [52], which could be very big, there may be a huge amount of temporal complexity.

3.6 TEXT MINING LEVELS

As a rule, there are two tiers of operation at the mining level (also called the feature level). Sentence level [58, 59] and document level [53–55] analysis. At the document level, activities focus on labelling the entire document as positive, negative, or neutral based on the polarity information provided. [60]. Among the most common tasks at the document level is labelling the entire content as either positive, negative, or neutral based on the polarity information provided. [61]. On the other hand, A scale can also be used to express the categorization. (ex. 1 to 5).

Documents might be either subjective or objective. A textual analysis is used to establish whether or not a particular document is a text. [62]. An impartial piece of writing, like a review of a product, has one central focus. Since this text mining level does not go into the intricacies and instead conducts the review from a general and abstract standpoint, the mining process can be accomplished considerably more rapidly. The sentence level and the aspect level are prioritised more in the other two classes. This is not the best way to evaluate a summary because it assumes that there is only one thing.

The purpose of a document's level analysis is to guess its fundamental mood. Foretelling a document's polarity can be the target of a text classification issue. However, there are issues at the document level because any technique used will function well for the domain to which it is applied, but poorly in another domain. It's possible for a word's definition in one field to be identical to that in another. Given the importance of domain-specific factors, Yang et al. [63] mined data from movie reviews to create a dataset for product evaluations.

3.7 TEXT MINING METHODS

Reviews of movies or products can be sorted into positive or negative categories using the textual analysis technique. Below are described the three most common approaches to text mining: Three different approaches, including a hybrid one, are compared and contrasted [64]. Supervised learning, unsupervised learning, and semi-supervised learning are the three subfields of machine learning. The focus of this thesis is mostly on supervised learning.
Automatic text classification forms the basis of the supervised learning technique, which employs labelled training materials. An application of the training set that has been tagged with a certain category. Predicting the kind of document from a set of known categories requires a classification model. In the first step, a training set is used to teach the classifier. The model is then used on testing data to determine how to categorise the texts' emotional tone. Many sources agree that SVM, NB, KNN, and DT are the top classification methods for text mining [67].

3.7.1 Naive Bayes

The NB approach is both easy to use and widely used, making it a powerful tool for text mining. Here is a mathematical equation that describes the workings of NB. This strategy relies on the conditional probability formula (24 = 8).

$$P(l|d) = \frac{P(d|l)P(l)}{P(d)} \quad ,P(d) > 0 \qquad 3.4)$$

d is the document that needs categorising and l is the document class [68]. In practice, it was observed that P(d) did not have a noticeable impact, therefore it was removed. The Bayes rule was developed by researchers based on the equation (25).

$$P(l|d) = p(d|l)P(l) \qquad 3.5)$$

For all the document, they calculated the features or words in the documents [2] using the equation (3.6) described below:

$$P(l|d) = P(x_1, x_2, x_3, \dots., x_n|l)P(l)$$
$$P(x_1, x_2, x_3, \dots, x_n|l) = P(x_1|l).p(x_2|l).p(x_3|l) \dots P(x_n|l).p(l) \quad 3.6)$$

Features from document d are denoted by x1 through xn. The researchers have utilised the formula (26) to determine the value of each word in the document as a vector. An example of a matching Bayes classifier is the function y = class(c) that assigns a label class to the given c in the following equation (3, 7).

$$y = P(l_j)\prod_{i=1}^{n} P(x_i|l_j) \quad 3.7)$$

In this case, y corresponds to the last category given in the test and is compared to the test data label. Based on the NB, the maximum and minimum values of y were calculated using the formula (3–7). As P(l j) was constant across all NB types, they utilised the following two equations to determine P(l j) and P(x i | l j), respectively; however, P(x il j) values varied between techniques depending on the classifier employed.

$$P(l_j) = \frac{documentP(l = l_j)}{N_{doc}} \quad 3.8)$$

$$P(x_i|l_j) = \frac{count\ (x_i|l_j)}{\sum_{x\in \text{v}} count(x|l_j)} \quad 3.9)$$

The main drawback of the NB was that the chance of a given attribute value being zero based on frequency if the class label and the value do not occur together. By adding 1 to the problem-solving equation (3- 10) below, the researchers employed Laplace smoothing techniques.

$$P(x_i|l_j) = \frac{count\big(x_i\big|l_j\big) + 1}{\sum_{x\in \text{v}} count\big(x\big|l_j\big) + 1} \quad 3.10)$$

There are four types of NB have been used in the present study, as described in the next sections [3].

3.7.2 Support Vector Machine

One of the most popular classifiers is the support vector machine [75]. SVM's core idea is to employ hyperplanes to partition classes into distinct groups [61]. SVM has the advantage of high precision levels when working with linearly separable data but does not perform as well when working with non-linearly separable data. Using kernel functions to transform the information into a higher dimensional space is the key to fixing this problem. This allows for linear separation of the data. The primary objective of supporting vector machines is to pick the best kernel function and adjust the parameters of the kernel. Calculating the optimal course of action is a type of optimization issue. An appropriate decision plan would allow the kernel function to create linear decisions via a nonlinear transformation, as seen in the equation. (2.14).

$$f(x) = \mathcal{W}^T x_i + b$$
$$f(x) = \sum_{i=0}^{N} \lambda_i l_i (\mathcal{W}_i^T x + b) \quad 3.11)$$

The symbol WTindicates the weight of the vector, f(x) represents the feature sets for both classes, _i stands for the dual function that is returned after training, x stands for the dataset used in training, l stands for the classes that are produced, and b bias denotes an omega θ.

For SVM to function optimally, linearly separable criteria are required. But there are a few situations where a straight line can't be drawn between two points. When applied to data with non-linear features, SVM's performance suffers significantly. The kernel method, shown by equation (2.15), was created to address this problem. This method allows a linear classifier to be used in situations where the data is not linear. In addition to helping with non-linear problems, kernels also have the advantage of being very customizable. Using a radial basis function (RBF) kernel and optimising its sigma and c parameters is a popular kernel approach. Because it is flexible, the RBF kernel is often used to fit data. The polynomial kernel and the sigmoid kernel are two others widely used kernels that have been optimized.

In addition to the kernel and SVM parameters, the data itself is a crucial factor in classification accuracy. The SVM method is data-driven and requires high-dimensional data to plot training examples. While the kernel is responsible for the translation into high-dimensional space, the accessibility of the feature set has a significant impact on how straightforward the separation is. It contains a list of kernels of various types. (2 16) [76, 77].

$$f(x) = \sum_{i=0}^{N} \lambda_i l_i (\mathcal{W}_i^T x + b) K(x_i, x_j) \qquad 3.12)$$

The kernel become as explained in the following equation.

$$K(x_i, x_j) = \begin{cases} linear & (x_i . x_j) \\ rbf & e^{(-\gamma || x_i - x_j||^2)} \end{cases} \qquad 3.13)$$

3.7.3 K-Nearest Neighbour

Data mining, statistical pattern identification, and machine learning are just a few of the many domains where K-Nearest Neighbor (KNN) has proven useful. [78]. KNN organizes features into classes according to their proximity in the area of interest, as determined by the training samples. Figure 2-8 is a visual representation of KNN analysis in motion, where the query point (represented by the black circle) shows how to label an unexplored value. Instances are represented by the green squares and the yellow triangles. [79].

In Figure 2-8, the green line denotes the recommended path. After samples have been classified using the most similar sample from training, we can then establish whether or not these samples are actually green circles. The primary benefit of this approach is the ability to sort query results into categories according to the frequency with which they were used during training. To classify new objects from known examples, the KNN technique is used in this thesis. Another benefit of KNN is that it can be used without making any prior conclusions about the data distribution. compiled information [80].

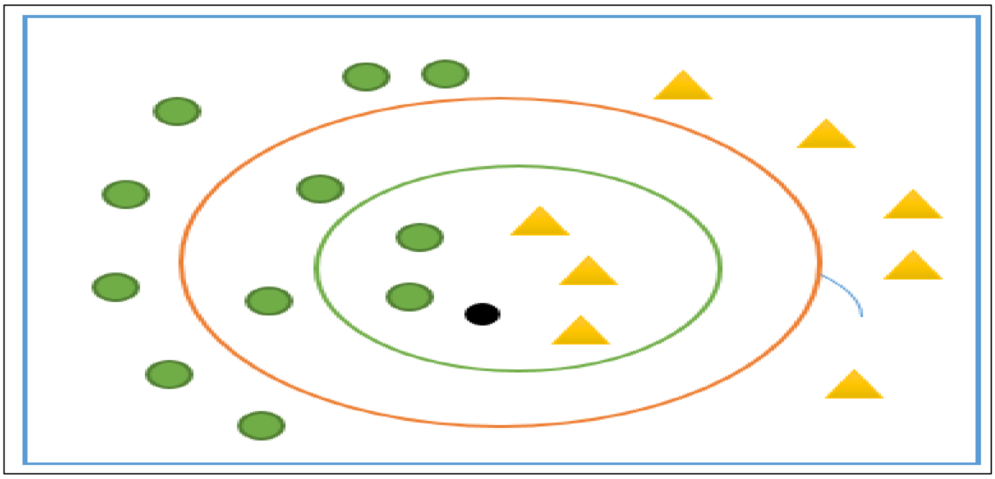

Figure 3.3: KNN Example.

In Figure 3.3, the black circle depicts the sample being tested, which must be classified into one of two groups. (The yellow triangle or the green circle). Since three of the five numbers in the green line circle suggest K=5, that means the yellow triangles get it if K=5. If K is 7, then four of the seven numbers depicted by the orange line circles are green, indicating that circles.

Before the KNN method can be implemented, the parameter K and the total number of nearest neighbors must be established (NN). After the parameters are verified, the NN method of KNN minimal distance is used to determine how far apart the query feature and the training instances are. It is best to use a large K value to increase the probability of precise results and decrease the overall noise in the dataset. The example shows that if K is between 3 and 10, the results are more exact than if K were set to 1. Typically, the KNN employs either the Euclidean or Makowski distance. These evaluations boost the KNN's precision by employing specific models like neighborhood segment analysis or large margin nearest neighbor. [81]. The KNN algorithm has disadvantages, such as the difficulty in locating closest neighbors for each sample, as shown in equations (2- 17).

$$d_{st}(x, y) = \sqrt{\sum_{i=1}^{n} |x_2 - y_2|^p} \qquad 3.14)$$

Therefore, d_{st} refers to the distance metrics, x_i *and* y_j represents the element of *x*, if *p=1* the Minkowski metric or *p=2* then Euclidean metric and *y* as shown in equation (3.14).

3.7.4 Decision Tree

The DT approach employs a tree-like structure to construct classifiers. The tree structure is a method of partitioning a large dataset into manageable pieces using branches and leaves as nodes. The final result is a tree structure made up of individual nodes. The ML method, which involves the integration of computer tools and mathematical equations, can be thought of as aiding in the generalization, representation, and classification of a specific dataset. The non-leaf nodes in an ordered tree are often connected to the leaf nodes through arcs. [82]. Each non-leaf node is represented as an input feature, and its arcs are organised as feature values. Leaf nodes all belong to the same class value or probability distribution. The entropy method gives a number between 0 and 1 as a result. This number shows how consistent a dataset is [83].

It is built from the top down, beginning with a root node, and contains subsets whose features all have the same target (goal) values. [84]. In other words, entropy is a preset component level to quantify deterioration. Probability calculations, like those in the equation, are often used to determine this. [85] (two 18) and (two 19):

$$E(S) = \sum_{i=1}^{c} -p_i \, log_2 \, p_i \qquad 3.15)$$

$$E(T,X) = \sum_{c \in X}^{c} p_{(l)} E\,(l) \qquad 3.16)$$

In this equation, S represents the current state, Pi the probability of an occurrence, I the fraction of the state S, and I the node of S. The entropy of a collection of characteristics is expressed by a mathematical expression. (316). In this case, X is the desired quality of the present state T. Assembling a DT requires locating a component willing to recover the maximum information gain. (IG). It's crucial to zero in on the selected qualities of the split, as they produce the least amount of pollution. Characterizing the IG at any of the nodes in equation (2.20) yields the following [4].

$$IG = entropy\ (T) - ([\ weight\ Avarage\ ] * entropy\ (feature))$$
$$Gain(T, X) = Entropy(T) - Entropy(T, X) \quad 3.17)$$

3.8 EVALUATION MEASUREMENTS

A confusion matrix, which is also sometimes called a "contingency table," was used as a way to measure how well something worked. In the contingency table, there are four donations: "True Negative," "True Positive," "False Negative," and "False Positive" (FP). Donating TP is both one of the most accurate ways to describe a negative case and one of the most accurate ways to describe a positive case. Also, FN shows the positive instance, which is mistakenly grouped with the negative symbols, and FP shows the negative symbols, which are mistakenly grouped with the positive symbols. Table 3.1 shows all of the equations that are used to measure how well the classifiers work.

Table 3.1: Performance Metric.

Metric Name	**Calculation**
Sensitivity or Recall	$TP/(TP + FN)$
Specificity	$TN/(TN + FP)$
Precision	$TP/(TP + FP)$
F1 Score	$2 * (Precision * Recall)/(Precision + Recall)$
Accuracy	$(TP + TN)/(TP + FN + TN + FP)$
Area Under ROC Curve (AUC)	$0 <= Area\ under\ the\ ROC\ Curve <= 1$
ROC	1 – specificity

4. RESULTS & DISCUSSION

4.1 GEN DISEASE CLASSIFICATION

This study is mostly about the method of machine learning. Machine learning is used to solve the polarity problem, which is based on two feature extractions: TF and TF-IDF with unigram. In addition, this study used the disease abstract corpus and four important algorithms (SVM, NB, KNN, and DT) with unigrams. This study tries to compare the different ways that machine learning can be used to choose the best feature extraction and selection methods.

4.2 POLARITY AND TARGET CASES

Polarity is one of the most important problems that researchers all over the world have to deal with, especially when it comes to schools and the government. To get the polarity out, people have to work hard. Finding good solutions for things like polarity is part of the work that goes into making medical articles. Depending on the problem, the constraints can be different for different goals. Figure 4-1 shows an example of a short abstract that has to do with this work:

Abstract	**Polarity**
The NAT2 slow acetylator genotype is associated with bladder cancer in Taiwanese, but not in the Black Foot Disease endemic area population.*	Cancer

Figure 4.1: Example of Short Abstract.

In the short, abstract example shown above, cancer is the polarity. The goal of this thesis is to find the opposites in each abstract. Given a set of abstracts A, the task is to find the text (for example, "equation 3") that goes with the abstract.

$$Disease = f(\ P\ ) \qquad 4.1)$$

The polarity, P, is calculated using the formula, f. Assume that, for some j, when referred to by Each abstract is broken down into sentences as described in the preceding chapter. The quantity of words in sentence k of article i is represented by the variable r. Each of the k sentences in the piece is presented as a word cloud using the following equation.

$$W_i = \{ w^i_{k,r} | w_{k,r} \in s_k \ and \ s_k \in A_i \} \tag{4.2}$$

From the list of words $w^i_{k,r}$, this study create feature for building the vector as in the equation (4.3).

$$F_j = \bigcup_j w^i_{k,r} \quad , where \ w^i_{k,r} \ \in A^j_i \qquad 4.3)$$

This study used four abstract labels—Alzheimer's, Cancer, Diabetes, and Syndrome—where Fj is the finite set of words for generating the vector that can be used in this study and j refers to label of abstract. The problems associated with polarity are discussed in the next two parts.

4.3 THE PROPOSED MAIN FRAMEWORK

Figure 3 shows that the proposed main framework has two parts: one for polarity and the other for target.

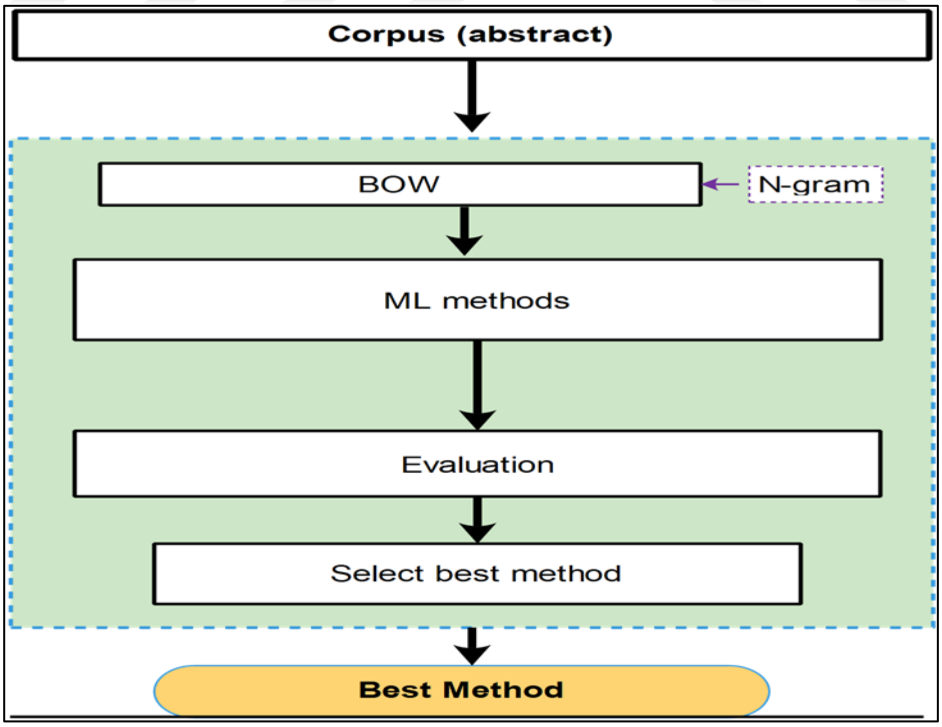

Figure 4.2: The Proposed Main Framework.

Two approaches to addressing the burden of illness are depicted in Figure 3.2. The polarity section uses two approaches: machine learning and a hybrid approach. SVM, NB, KNN, and DT were the four effective algorithms employed in the machine learning technique, along with two feature extractions (TF and TF-IDF). For these special effects, unigram was the method of choice. In Chapter 4, we learn all about the evaluation procedure.

The part about polarity talks about how to choose the best method for extracting features and the best algorithm for machine learning. After all the testing is done and the best thing for each method is chosen based on how the votes are counted, a comparison is made between the best machine learning method.

4.3.1 Corpus Collection

In an online search for disease abstracts, there is a very important problem to solve. This problem is made worse by the fact that there is no standard disease abstract dataset. So, in this experiment, a dataset made up of information from different sources was put together. Disease abstracts, on the other hand, were gathered based on certain criteria that fit with the proposed method. The chosen sets of data came from different places, such as (•). The disease abstract was put into six groups (• ——). Error! gives a summary of these groups. There was no source for the citation.

Table 4.1: Number of Selected Articles (Corpus).

Label of abstract	Abstract Number
Alzheimer	150
Asthma	150
Cancer	200
Diabetic	150
Syndrome	166
Total	816

4.3.2 Removal of Termination Words

It is important to find the words in a text that don't fit into any of the categories. Researchers who work on information retrieval have noticed that English words and phrases like "a," "the," "and" and "there" are not useful for indexing. When these words are used in English documents, they have a very low discrimination score. These words won't help you tell the difference between documents whose content is about different things. Removal of word termination is the process of taking out functional words that aren't based on content from a list of words made by word extraction. To get rid of the word "termination," a partially automatic process is used. First, a list of "stop" or "termination" words is made as part of the list of words to get rid of. This list is called a "negative dictionary list," and the words on it are the ones to get rid of. Once the words have been made by the word extraction process, each word is scanned, compared to a list of words that don't belong in the dictionary, and then taken out. The hardest part of getting rid of the termination word is deciding which words should go on the list. Sometimes, certain words are useful for text classification and are related to the subject of the documents that are available. Words that aren't on the list of words to get rid of will be put in a special set of documents. For example, words like "house," "book," "page," and "web" might not be considered ending words in an English document, but they might be in a document on the World Wide Web about sports, politics, news, etc.

4.3.3 Stemming of Words

The morphological variants of a term should be present in a collection of texts. Words like "observing" and "seeing" have morphological variations that can be used to express meaning more precisely. Variations in morphology result in words that look different from one another yet have the same meaning. Consolidating the various morphological forms of related words into a single canonical form is an important step in the process of text classification. The process of combining or merging various morphological variants into a common canonical form is known as stemming in the context of information retrieval. The term "stemmer" describes this element of stemming. It is preferable in the research to create stemmers in which the stems are crafted from a predefined word set using translation rules. When a suffix or prefix is added to a word, this technique is frequently used to remove it, creating a new term. To get rid of suffixes dependent on recognising plurals, this stemmer uses a procedure called affix removal. While this stemming technique recognises the words, it has difficulties because not all words have plurals or end in "s" (such as "success")).To get around these problems, the stemmers approach uses rules in the form of conditional statements that depend on whether or not certain conditions are met. Here is a possible set of guidelines to follow when eliminating plurals, If there is a word finishes with “ies” can replaced by “y” but not for the words with “aies” or “eies”If there is a word finishes with “es” can be replaced with suffix “ e” but not for words with “oes” or “ess”If there is word finished with “s” , it can remove the word with “s”, but not for word ends with “us” or “ss” Different stemmers will have different sets of conditional rules for translating various affixes (i.e., prefixes and suffixes), such as ed->VOID, ational->ATE, etc. These sets of conditional rules can be broken down into smaller, more manageable chunks, and then applied incrementally to various translations. In order to extract features, stemming can be used, which is a many-to-one operation. While the translations used in stemming can be helpful, human understanding is essential for solving the stemming puzzle.

4.3.4 N-Gram

In this study, n-grams are used to figure out what something means. As was said in Chapter 1, there are a lot of problems. Some of these problems can be solved by preprocessing or stemming. But there are still problems, like when the order of words in a sentence means the same thing and makes it hard to understand what is being said. As it says in Chapter 1, Section 1.4, this phrase needs more than one word to be clear and easy to understand. There are also problems with syntactic and semantic issues when it comes to text mining. Another problem is something like the negation problem. The negation problem can be seen as when we have a sentence that means something, but when we find negation in English, the sentence means something completely different.

This study suggests using n-grams, and more specifically, unigrams, because the English language is hard to understand. As was already said, it is very important to use a unigram to solve the problem and come up with the right sentence. Also, to compare them and decide which one to go with, machine learning gets hurt. For more details, consider chapter 4. It shows a single word for a unigram. as 4.1 of a simple n-gram algorithm shows.

Start Algorithm 3.1 with the abstracts A = A 1, A 2, A 3,... A n. Finally, it assigns a value between one and three grams, written as G = one, two, three. The partition is weight-based for each individual abstract, A i. The gramme-specific symbol is then appended to T. The process of selecting gram units continues until all of the unigrams have been created. In this case, we have T_1, T_2, and T_3, where 1, 2, and 3 refer to specific gram amounts.

4.3.5 Multiclass Classification Experiment

Machine learning methods are used in this experiment, two malicious attack types of DoS Attacks-Hulk and DoS Attacks-Slow HTTP Test are detected using machine learning methods. The performance results for the machine learning methods utilized are based on four Metrics: precision, accuracy, f1-score, and recall. in Figure 4.20 shows Total Number of Samples in Each Category used in the Second Experiment (Multiclass Classification Experiment) between (DoS Attacks-Hulk and DoS Attacks-Slow HTTP). By analyzing the Python code, the result of all algorithms performance was as follows.

4.4 MACHINE LEARNING ALGORITHMS FOR DISEASE

The article's orientation has been determined with the help of polarity detection machine learning. Machine learning is employed here because it is simple to tune the settings and get precise outcomes. In addition, it can be used as a benchmark for the proposed hybrid approach. There are four major categories of algorithms used in machine learning. As discussed in Chapter 2, these types are widely used in text mining. SVM, NB, KNN, and DT are the names of these algorithms. This research makes use of machine learning algorithms on the PAAD datasets to determine how each algorithmic variant affects the task of determining an article's orientation. The corpora are mined for their n-gram characteristics. After employing TF and TF-IDF to construct the vector's weights, the vector is partitioned into training and testing sets. Figure 3 shows the complete set of classifiers.

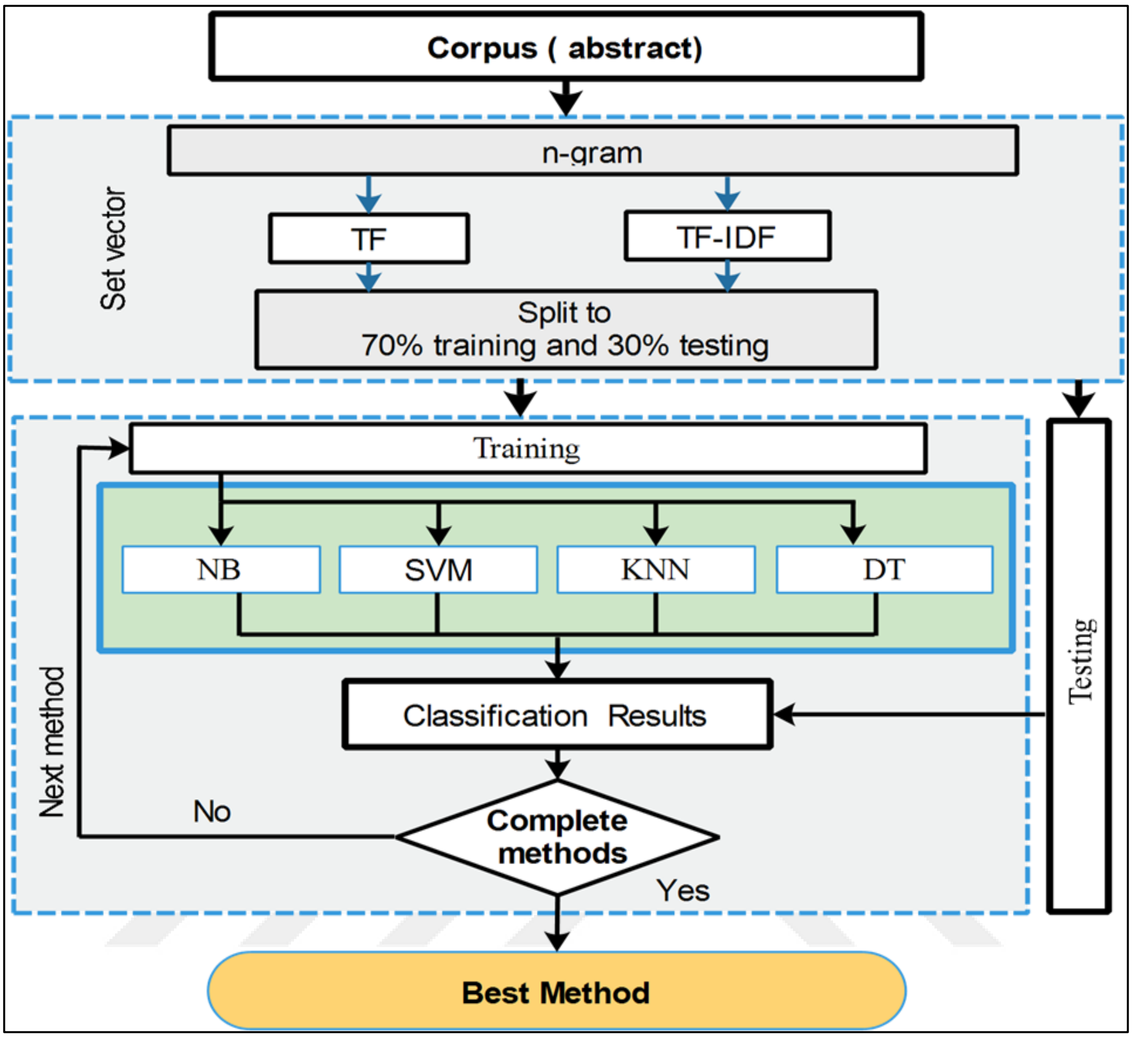

Figure 4.3: Machine Learning Method.

4.4.1 Feature Extraction (BOW)

Following the completion of the document-word extraction process. The text is parsed into a set of stems that can be used to find comparable words elsewhere in the document. The following step is to locate the union among all these groups; hence, the union set will include all the groups of stems associated with the words that appeared in the document group. Each set of stem collections in the union collection has had any redundant items deleted. The indexing terms for a given set of documents will be stored in a collection of stems during the information retrieval process. Indexing vocabulary is a compilation of terms used in the process of indexing.

Measurements are made for each term in the vocabulary based on how important it is in the document. This can produce a set of initial features from the indexing vocabulary. The process is done by giving each term a weight, so that each term in the documents has some kind of meaning. The process of giving each indexed term a weight is called "term weighting." When an indexing term is in the indexing vocabulary and has related term weights, the document is represented as a feature vector. Where is the total number of indexing terms in the vocabulary, or the size of the vocabulary, and what is the weight given to each indexed term in Document J? The inverse document frequency method is used to figure out how much each term is worth. It is a factor that is used to make an impact on each term's discrimination score. It is written as follows: where N is the total number of documents in the collection set, and n is the total number of documents where the indexing term appears. From what we know, inverse document frequency goes up when a term appears in a few documents. There are some assumptions that are based on what we know and are used in the proposed work. For example, documents with different sets of topics can be distinguished from limited documents with the same terms. Term weight is found by multiplying the number of times a term appears and the inverse of the number of times it appears in a document. So, a term that only appears in a small number of documents will have a higher term weight value. The five different "grammes" of Bag of Word (BOW) used here are the unigram, the bigram, the trigram, the four-gram, and the five-gram. In order to construct the machine learning approach, these grammes are employed to generate the vector. BOW employs two techniques: TF and TF-IDF. We adopted these techniques because TF just checks the article-word relation and the article-word effect, while TF-IDF checks the article-word and corpus-word effects. It determines whether or not the article's word has any connection to the corpus by looking up the word in the corpus. In addition, machine learning works with the numerical features offered by BOW; therefore, its use is preferred. Below, you'll get a breakdown of how often certain words appear in each article using BOW. Articles are presented in this research with phrase frequency weighting using the BOW, a machine learning technique. Equality (3, 4) can be used to determine the weight vector.

$$t_i^j = \begin{array}{c} \begin{matrix} & T_1 & T_2 & \cdots & T_m & \end{matrix} \\ \begin{pmatrix} A_1 & w_1^1 & w_2^1 & \cdots & w_k^1 & l_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_i & w_1^i & w_2^i & \cdots & w_k^i & l_j \end{pmatrix} \end{array} \qquad 4.4)$$

T is the features derived from the n-gram algorithm 4.1, where T_1, T_2, and T_3 are the weights of the term m in the article i, and w_ki is the weight of the term m. For the purposes of this paper, w_k_i is calculated using both TF and TF-IDF. The values of the weights, w_k_i, can either stand for individual words or for groups of words. For the purposes of TF and TF-IDF, this thesis employs a five-gram method for gathering term T and presenting it as a list of term words. As shown in a TF algorithm 4.2 and a TF-IDF algorithm 4.3, two different approaches are used to tally the weight w_ki as equation (4.4).

Algorithm 4.2: TF

Input: Set of articles

A=(A1, A2, A3,, An}, set of articles. G={1,2,3,4,5), the number of n-gram.

Procedure

T= set of unique term from n-gram (A, G). # Call algorithm 4.1

For each term tm ET # each TE (J, T2, T3, T4, T5}

For each article a, E A w= number of occurrences of term tm in article a,, #Eq (3.1)

End for.

End for

Output: Weight matrix

Weight matrix w such as equation (4.4)

Algorithm 4.3: TF-IDF

Input: Set of articles

A (A1, A2, A3, ..., An} list of articles. G-(1,2,3,4,5), the number of n-gram.

Procedure

T= unique term from n-gram (A, G). #Call algorithm 4.1

TFTF (A, G), this will calculate the TF weight matrix. # Call algorithm 4.2

Calculate IDF weight

For each term tm ET

For each article a, E A

If TFm.

End for

IDF = log(total articls / af_1)

0 then af ++ # Number of Articles in term m

Total articls A

#Eq (3-2)

Calculate TF-IDF weight

For each article a, E A

wk TFm. • IDF

this Eq for TF-IDF, Eq(3.3)

End for

End for

Output: Weight matrix

Weight matrix w such as equation (4.4)

Equation (4.4) is finished using TF and TF-IDF to account for the matrix weight. The percentages in Article A dictate how this matrix should be divided for use in both training and assessment. Based on Table 4.1, we recommend allocating 70% of A to instruction and 30% to testing.

Table 4.2: Number of Training and Testing.

Class	Training number	Testing number	Total
Alzheimer	150	150	
Asthma	150	150	
Cancer	200	200	
Total	816	816	

4.4.2 Machine Learning Construction

Discovering machine learning's inner workings is crucial. SVM, NB, KNN, and DT were the four methods utilized in this research. Testing these methods on the PAAD datasets helped determine the winners. This research set out to determine the polarity of political Arabic articles, provide a detailed account of the performance of each algorithm, and select the one that performed best overall and with BOW in specific. Using a corpus with a large number of examples, such as V1 through V4, and testing various techniques on each one is essential.

Equation (4.4) in the BOW portion supplies the features that are provided by the vector. If we assume that the article's fixed output is the label L and that X is a collection of features, we get Imagine we have an article, A, in the articles space, and a collection of articles, Tr, that have been labeled as part of our training set. Specifically, a learning classifier function that assigns names to articles is required for the learning approach to work. In our disease dataset, we have three categories in set L and four approaches to classifying them. The classifier cannot be built without first training the technique with the respective feature sets. Each method has its own output x, which is the class label while relates to the method, when given the input. The four procedures that were developed are as follows: Appendix stores the settings used for managing all algorithms.

4.4.2.1 Naïve bayes

The Bayes rule of probability theory is at the heart of Naïve Bayes classification, which is a way to learn by watching. The classification uses training data that has been labelled and is based on the strong assumption that all of the attributes in the training data are independent of each other. To figure out polarity from the PAAD datasets, a Naive Bayes classifier is built. Naive Bayes classifiers do a good job and can make decisions quickly. Their effectiveness has been shown by using training examples with multiple attributes. The main reason for this good performance is the assumption of independence.

Algorithm 4.4 returns training sets of articles as features (x_1, x_2,... x_n) and the class of the articles (l_(1,) l_(2,) l_(3,...)...,l_j), where x i is an input feature. The shape of the feature will be (x n,l j).In this case, all methods (Gaussian, Multinomial, Complementary, and Bernoulli) that use the P(l_j) for each class as in equation (2.8), where P(x i | l j) calculates the likelihood for each class depending on the method used, are applicable. Algorithm 3.4 looks at all of these ways and picks the best one.

Algorithm 4.4: Naïve Bayes
Input: Set of training and testing samples
Tr= ((x4)=(1,2,3,,n),j=
Error! Bookmark not defined.), set of training samples and class
Z= (z|i = (1,2,3,..., e), set of e test samples
Initialization:

Read the traning Tr

Y🡐 6;

Calculate the parameter for predict class Computation:

For ze Z do

a. P(1) calculate articles class by equation 3-8);

b. P(x) calculate the likelihood for each class depend on model (Gaussian, Multinomial);

c. y the label predicted by applying equation 3-7) on z, according to

d. and (b);

e. YY U{v};

Output: Set of predicted class

Y-ylie (1,2,3,.,e)}- the test samples in Z with the set of predicted class labels.

4.4.2.2 Support vector machine

A support vector machine is a binary classifier. It takes a number of variables as input and puts each feature (also called an input) into one of two groups. This study is trying to figure out how to recognise geometric shapes with three orientations and values for the amount of disease. It focused on the different numbers of kernels that SVM uses to figure out how well classification techniques work. In this case, this thesis used SVM with general kernel classification (linear, polynomial, sigmoid, and RBF) to sort PAAD into groups.

Given a training corpus like V1, V2, V3, and V4 with inputs and outputs, the inputs are sample features (x 1,x 2,x (3,,)x n), and the outputs are classes (l 1,l 2,l 3,...), (x n,l j), where x i is an input feature and l j is a reform, conservative, or revolutionary class. The optimization problem in equation (2.15) can be solved with this method (Chapter 2). There is a set of weights, W i or (W), that can be used to guess the right value of (l), which tells us how the article is oriented. Algorithm 4.5 below picks the best kernel for figuring out which way the article is facing.

a. Algorithm 4.5: SVM

b. Input: Set of training and testing samples

c. Tr={(x,4)|i = {1,2,3,...,n), j = (1, 2, 3}], set of training samples and class

d. Z (z, (1,2,3,, e), set of e test samples

e. Initialization:

f. Y 🡰 0;

g. Computation:

h. For ze Z do

i. Ke the kernel functions (Linear and RBF) according to Tr

j. y the class predicted by applying Ke on z

k. (Y, YUV)

l. Output: Set of predicted class

m. Y-(y,l. (1,2,3,.,e)}- the set of predicted class for the test samples in Z.

4.4.2.3 K-nearest neighbor's

K-Nearest Neighbor (KNN) is a type of machine learning method that uses a distance metric. This method uses the Euclidean distance metric and the Minkowski distance metric, which are described in Chapter 2 Equation (2.2.17). The benefit of using KNN is that you don't need to know anything about how the disease article dataset is split up in advance. This is because the KNN algorithm can use training samples to classify a new object. For the KNN algorithm to work, the parameter K and the total number of Nearest Neighbors must be known (NN). For testing the data, the algorithm used 3-K subsets, and it chose the 3-K with the least distance. If you choose small values for K, the result can be very accurate and perform well.

Given a training set for each corpus the sample features (x_1,x_2,x_(3,…,) x_n) and the output classes {(l_(1,) l_(2,) l_(3,)…,l_j), (x_n,l_j)} where x_i∈input features and 〖 l〗_j∈{classes}. Algorithm (3.6) below selects the best metric for detecting the orientation of the article.

a. Algorithm 4.6: K-Nearest neighbor's algorithm (learning method).

b. Input: Set of training and testing samples.

c. Tr={(x,,4)|i ={1,2,3,...,n), j = {1,2,3}}, set of training samples and class

d. Z = {z,i = {1,2,3,..., e), set of e test samples (e) K-the total number of nearest neighbors (3);

f. A-A distance measures model (Euclidean or Minkowski);

g. C-A classification method.

h. Initialization:

i. Y-8;

j. Computation:

k. For ze Z do

l. (a) Nthe nearest refers to K neighbors to z, from Tr according to A;

m. (b) f ← the discriminant procedure of C trained on element n;

n. (c) y -the class label predicted by employing fon z,;

o. (d) Y YU {y}3;

p. Output: Set of predicted class.

q. Y= {y_1|i = {1,2,3,..., e}} the test samples in Z with the set of predicted class labels.

4.4.2.4 Decision tree

Grouping strategies are developed using a tree layout by the DT algorithm. The tree structure is used to divide each corpus in a PAAD collection into smaller groups. Nodes are represented by the stems and leaves. In this investigation, the DT was created, the PAAD datasets' corpus was run using the selected number of trees, and then the proximity values were updated. The training procedure informed the development of the classification. Cases were ranked on a decision tree from its starting point at the root node to its exit at the last leaf node. The instance's categorization was thus determined. In the case of the new node, this procedure is repeated for the child subtree.

For the purposes of Algorithm 4.7, we'll assume that the number of classified instances L=l_(1,) l_(2,), and l_(3,) and that the characteristics of each instance l_(), which are p-dimensional, are x_(1i,) x_(2i,)..., x_pi. Features such as p_(i)'s category membership are represented by x_i. In order to determine the entropy of each class in each PAAD dataset, the method uses the formula (3.16).

a. Given a training set((x1,4),..., (xn, 4)], where x, E Ra and I, represents classes.

b. Compute entropy of corpus using Eq (3.16).

c. For every feature xn.

d. Calculate entropy of features for all classes # pick root.

e. Take average information entropy for the current feature.

f. Calculate the information gained for the current feature using Eq 3.17).

g. Pick the highest information gained feature.

h. Repeat until getting the tree desired # complete rules.

4.5 RESULTE

4.5.1 DT

Table 4.3: TF Feature Extraction with DT Algorithm.

Class	Precession	Recall	F-score	Accuracy
Alzheimer	0.97	0.87	0.92	
Asthma	100	100	100	
Cancer	0.98	0.88	0.92	
Diabetic	0.90	0.98	0.94	
Fabry	0.82	0.96	0.88	
Syndrome	0.92	0.90	0.91	
				92.683

In this table, class (Al-Azma) recorded the highest percentage in the precision and the Recall class, while class (Fabry) recorded the lowest value in the Precision and the lowest percentage in the Recall class. It was found in the Cancer category, with the highest thousands score in the Crisis category and the lowest in Class Fabry.

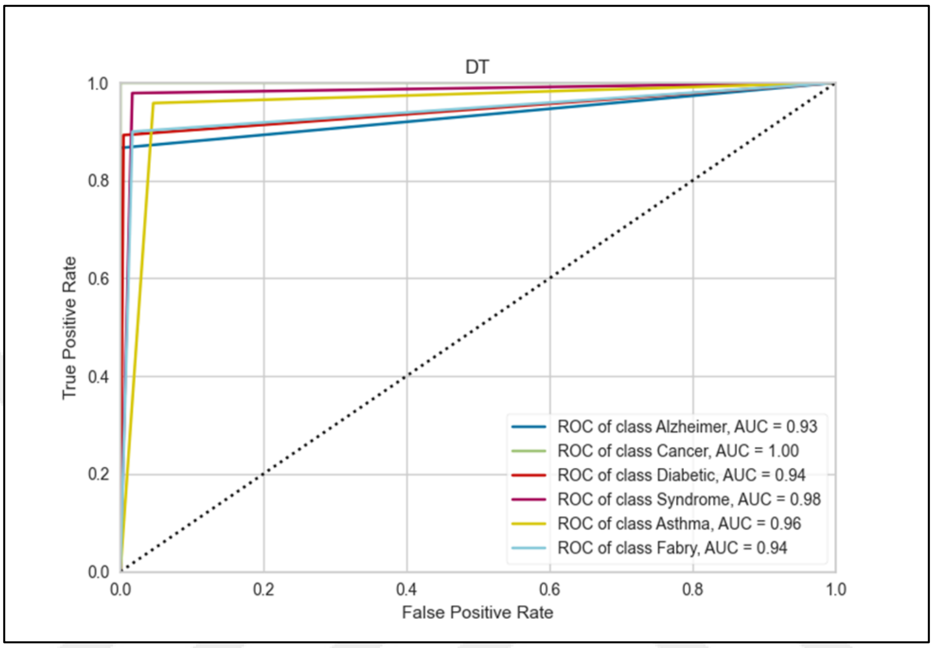

Figure 4.4: ROC with TF Algorithms.

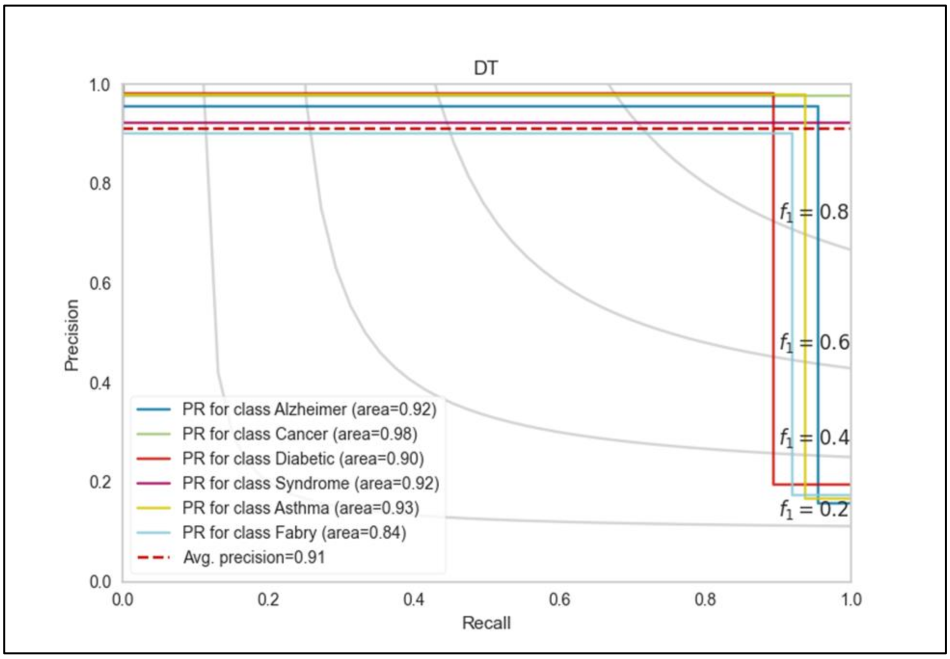

Figure 4.5: PR with TF Algorithms.

The highest PR for class Cancer was 0.98, and the lowest PR for class Fabry was 0.84, while the average (precision) was 0.

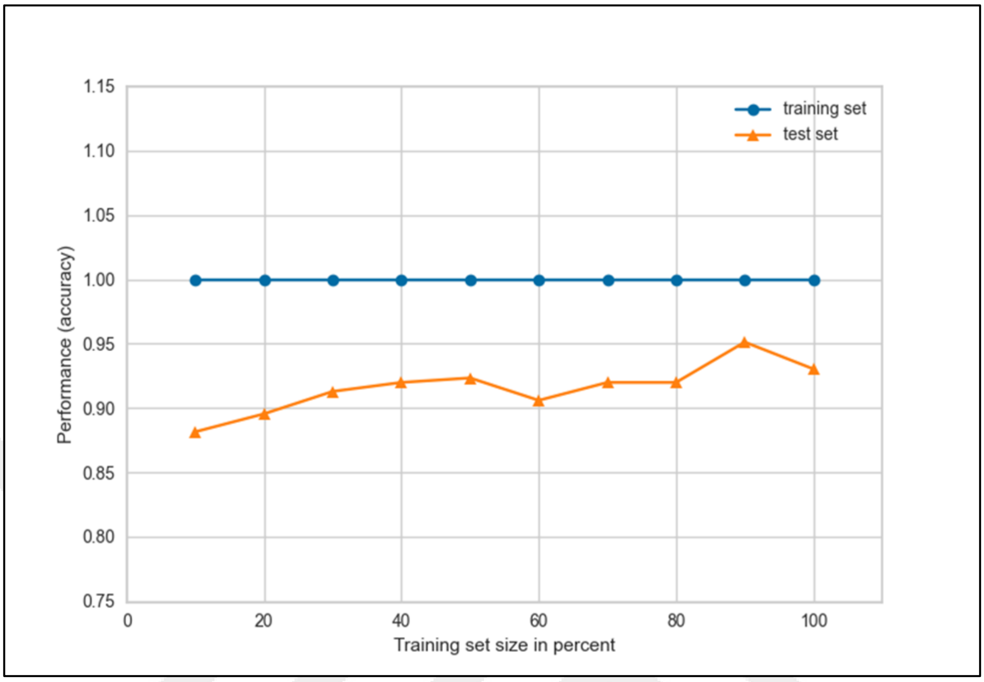

Figure 4.6: Training of TF Algorithm.

Table 4.4: Term Frequency – IDF - 1.

Class	Precession	Recall	F-score	Accuracy
Alzheimer	1.00	0.93	0.97	94.077
Asthma	1.00	1.00	1.00	
Cancer	0.95	0.93	0.94	
Diabetic	0.92	0.96	0.94	
Fabry	0.87	0.96	0.91	
Syndrome	0.94	0.88	0.91	

Here is the highest value (for Currency) recorded by Class Cancer, which is 1.00, and the lowest value recorded by Class Fabry, which was 0.93.

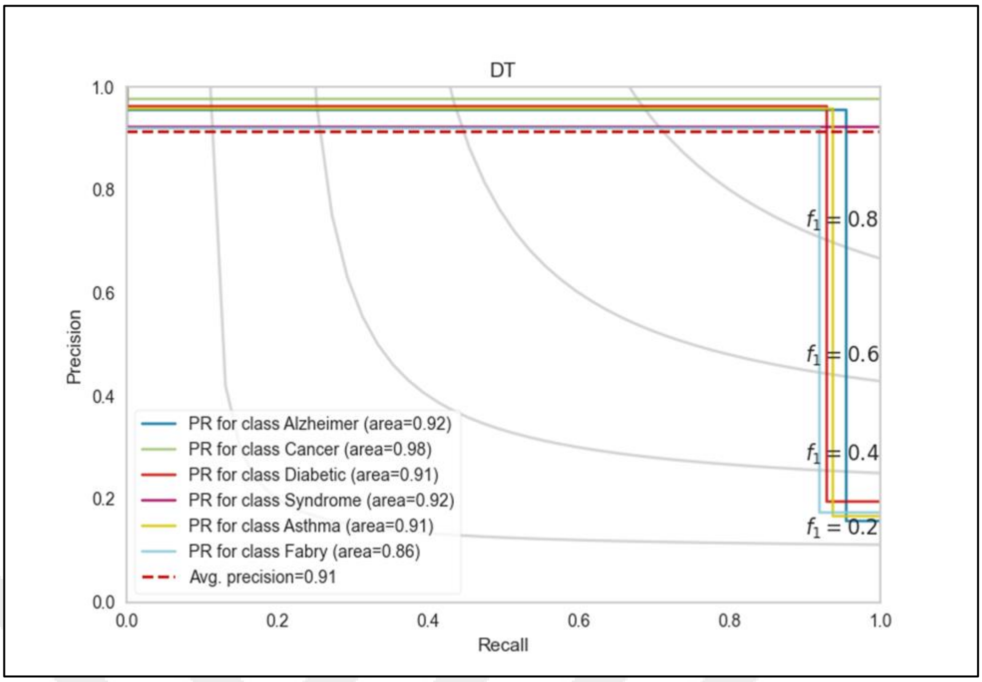

Figure 4.7: PR with TF Algorithms.

In this drawing, the PR of the class for cancer is the highest percentage, which is 0.98, while the PR of Class Fabry is the lowest value, which is 0.86, and the rate of exposure is 0.91.

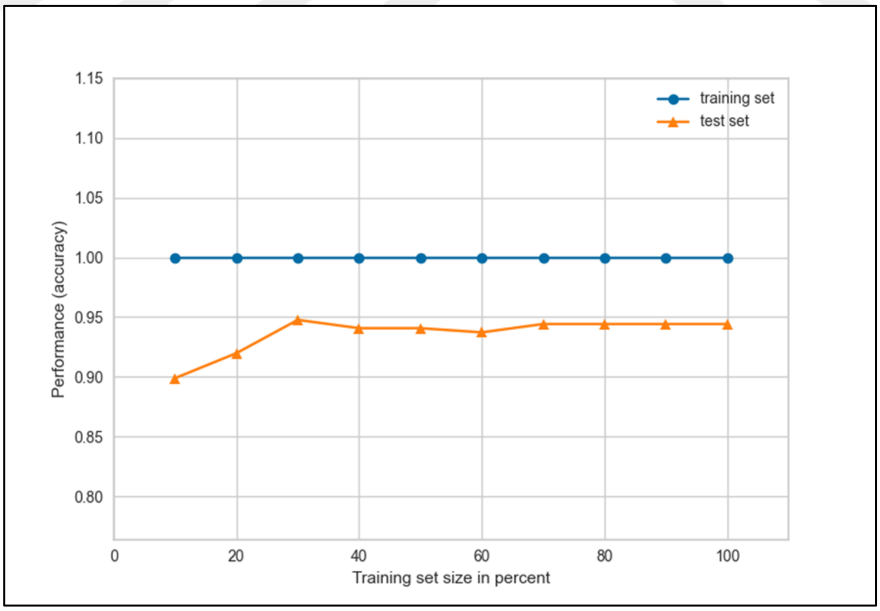

Figure 4.8: Training of TF Algorithms.

4.5.2 Support Vector Machine SVM

In the Algorithm ( ) Class Crisis, the highest percentage was recorded in Precision and Recall, which is 1.00. While the lowest percentage was in the Precision class, which was 0.75, it was in the Fabry class, while in the Recall, the lowest percentage was for Alzheimer's class, which was 0.76. Also, the highest percentage in thousands of scores was for the Crisis class, which was 1.00, and the lowest percentage was in the Alzheimer's and Cavalier classes, which was 0.85.

Table 4.5: Term Frequency - 1.

Class	Precession	Recall	F-score	Accuracy
Alzheimer	0.97	0.76	0.85	88.85
Asthma	1.00	1.00	1.00	
Cancer	0.88	0.95	0.91	
Diabetic	0.97	0.77	0.86	
Fabry	0.75	0.98	0.85	
Syndrome	0.86	0.88	0.87	

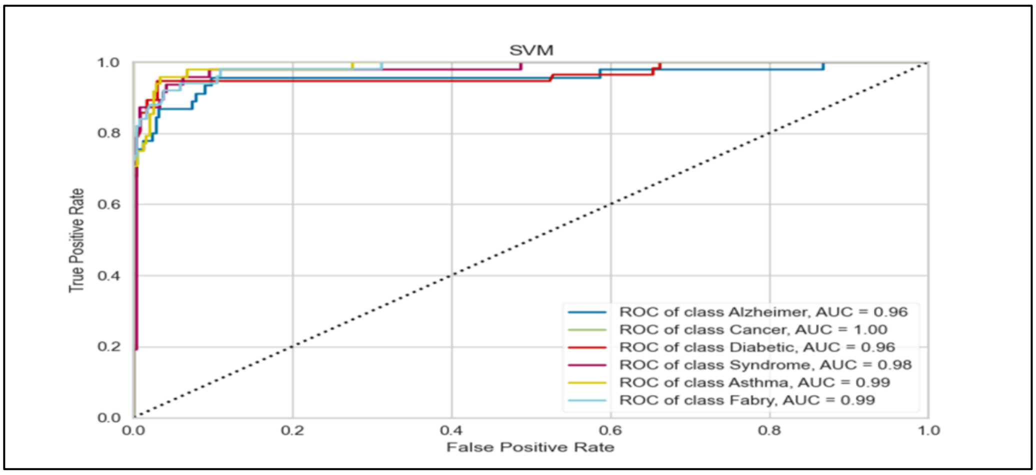

Figure 4.9: ROC of SVM Algorithms.

In this graph, the highest rate of Accuracy was recorded by the cancer, which is 1.00, and the lowest percentage was recorded by Alzheimer's and Diabetes classes, which is 0.96.

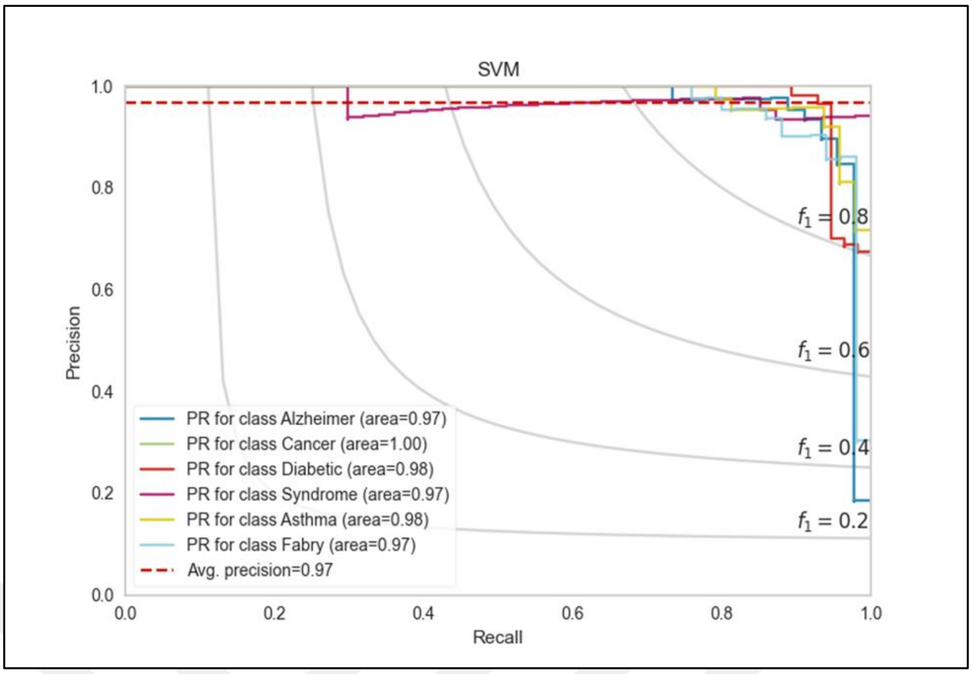

Figure 4.10: PR of SVM Algorithms.

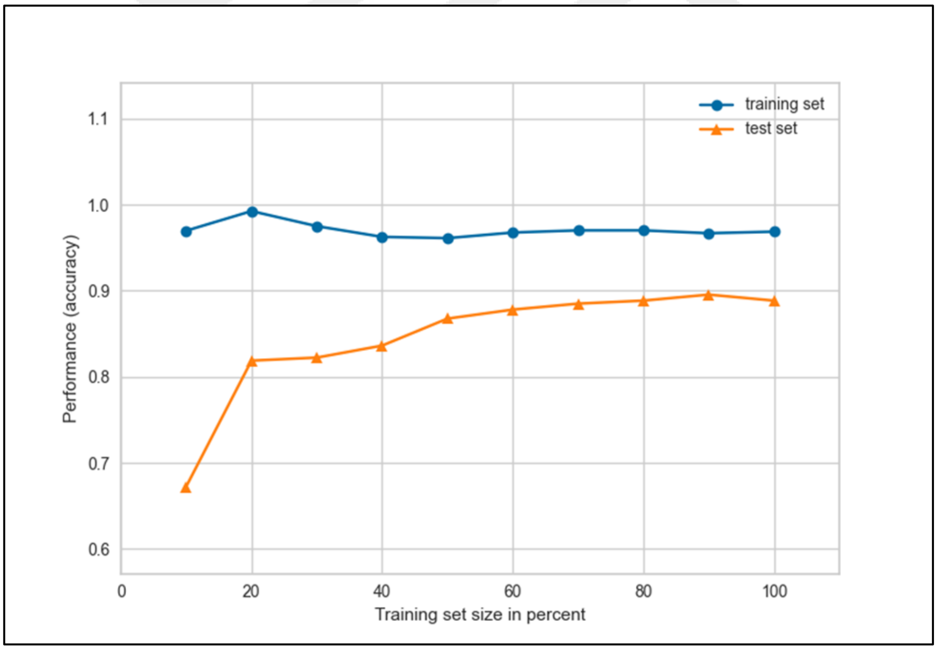

Figure 4.11: Training of SVM Algorithms.

Table 4.6: TF-IDF - 2.

Class	Precession	Recall	F-score	Accuracy
Alzheimer	0.98	0.93	0.95	95.819
Asthma	1.00	1.00	1.00	
Cancer	1.00	0.96	0.98	
Diabetic	0.98	0.94	0.96	
Fabry	1.00	0.96	0.98	
Syndrome	0.83	0.96	0.89	

Here in this log, notice that the highest value in the precision was for the Crisis, Cancer and Fabry classes, which is 1.00, while the lowest percentage was for the Sanrum class, which is 0.83.

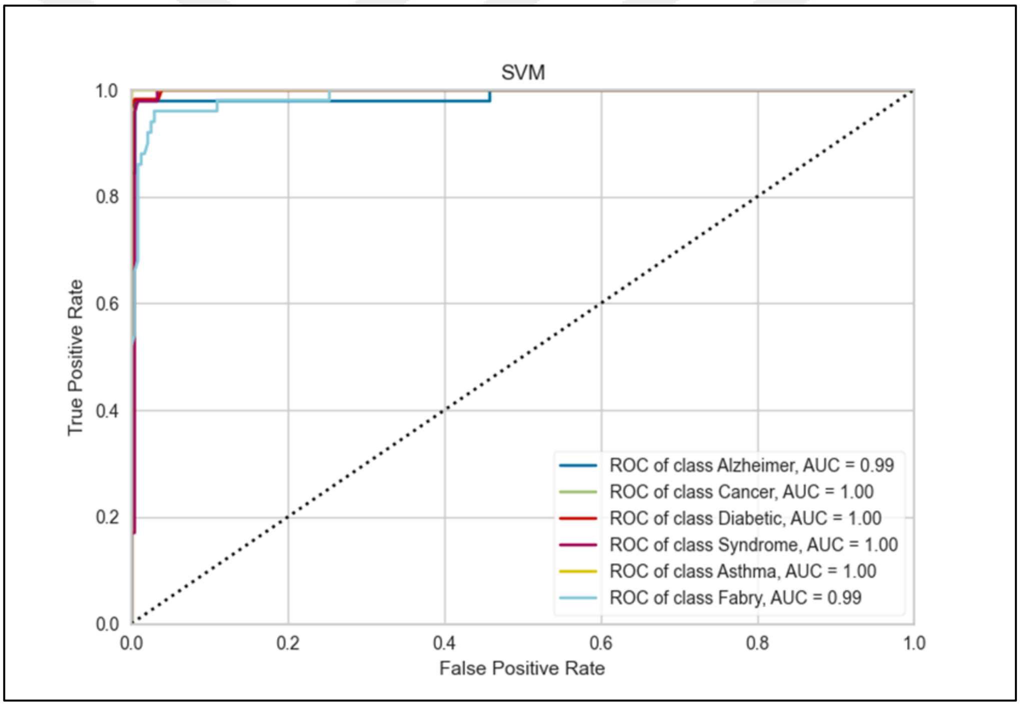

Figure 4.12: ROC of SVM Algorithms.

This table indicates that the acuity has the highest value in each of the classes of cancer, crisis, diabetic, and syndrome, which is 1.00, while the lowest value was recorded in the Alzheimer’s and Vibrio classes, which is 0.99, which is.

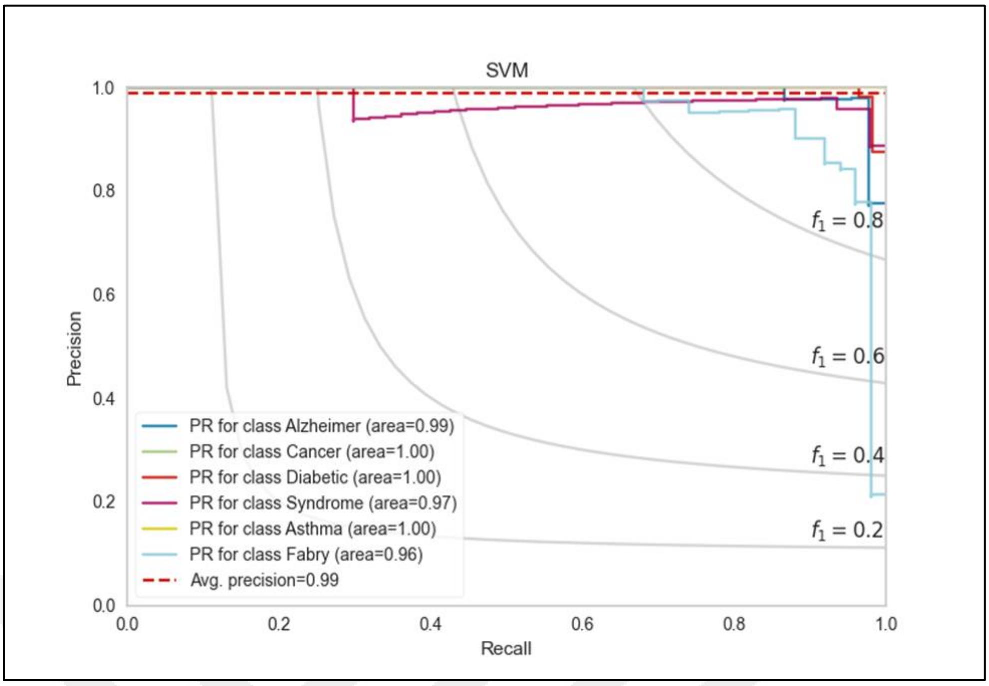

Figure 4.13: PR of SVM Algorithms.

In this figure note that the LPR in the class of Cancer and Dipti crisis recorded the highest value, which is 1.00, while the LPR in the class of Fabry recorded the lowest value, which is 0.96.

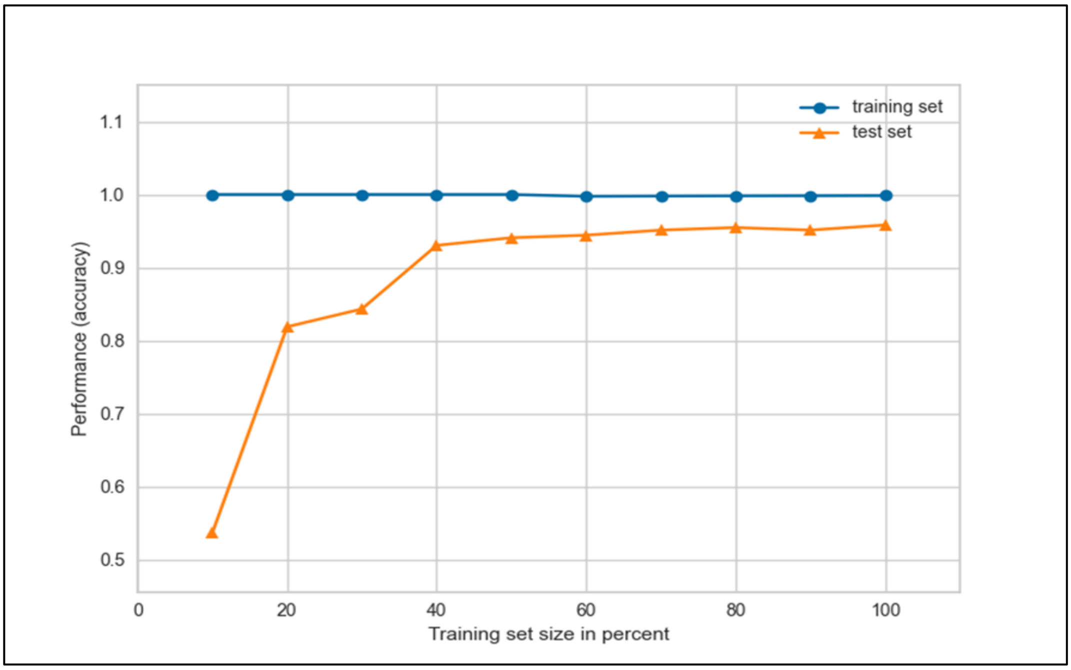

Figure 4.14: Training of SVM Algorithms.

4.5.3 KNN

Table 4.7: Term Frequency - 2.

Class	Precession	Recall	F-score	Accuracy
Alzheimer	0.71	0.33	0.45	52.613
Asthma	1.00	0.22	0.36	
Cancer	1.00	0.45	0.62	
Diabetic	0.94	0.32	0.48	
Fabry	0.35	0.92	0.51	
Syndrome	0.47	0.86	0.61	

In this table of the ironing algorithm, note that the highest value in the Precision was in the Crisis and Cancer classes, which is 1.00, while the lowest value was in the Fabri class, 0.35, while in the Recall, the Viberian class recorded the highest value, which is 0.92, while the lowest value was in the Diabatic class, hoping for thousands of scores The highest value was recorded in the cancer class and the lowest value in the crisis class.

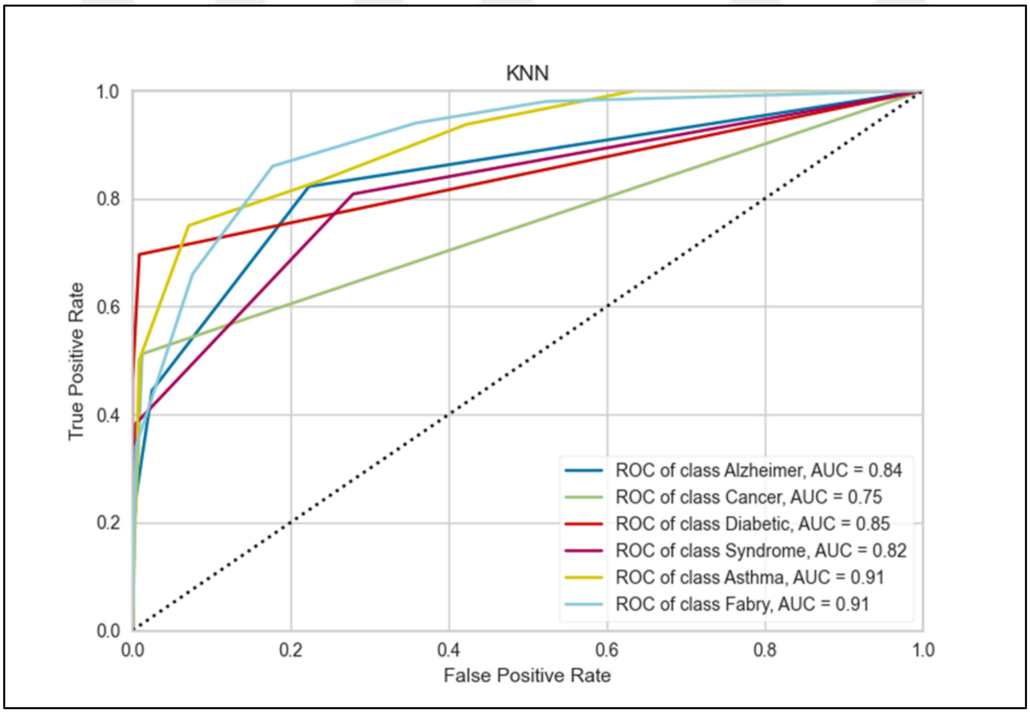

Figure 4.15: ROC of KNN Algorithms.

Here, the highest value of Currency was in the Crisis We class, which is 0.91, and the Fabry class and class, and the lowest value in the Cancer class, which is 0.57.

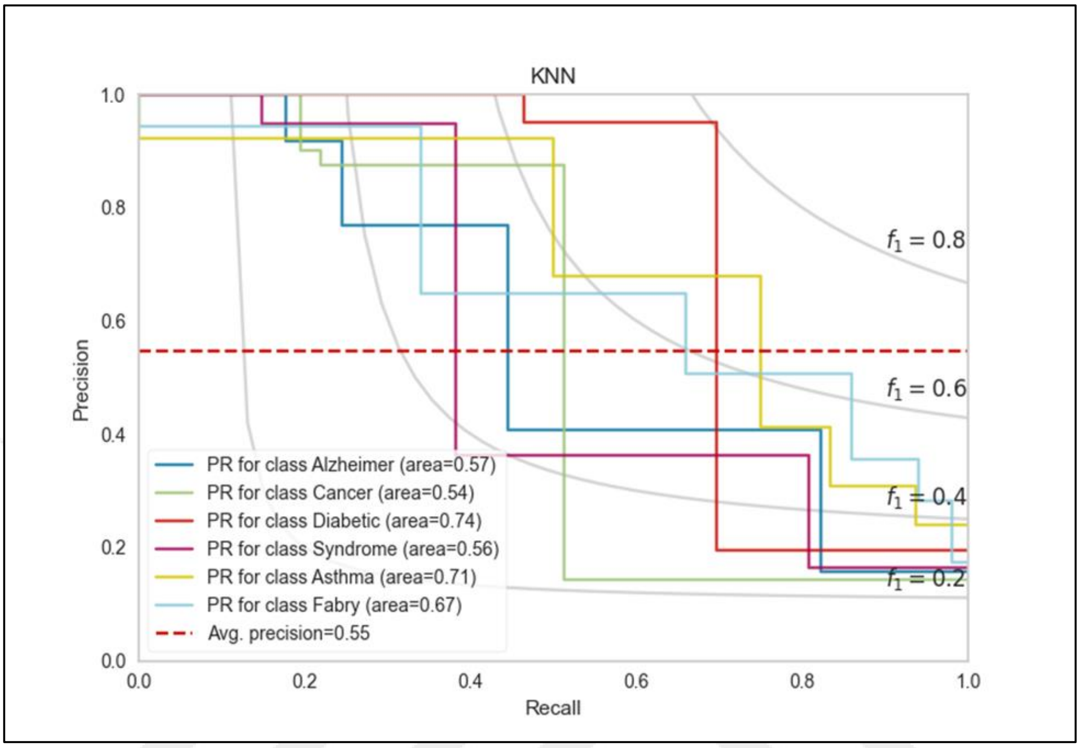

Figure 4.16: PR of KNN Algorithms.

Here, notices the highest value of the PR was in the Diabetic class, which is 0.74, while the lowest value was in the Cancer class, which was 0.54, The lowest value was in the Cancer class, and it was 0.55.

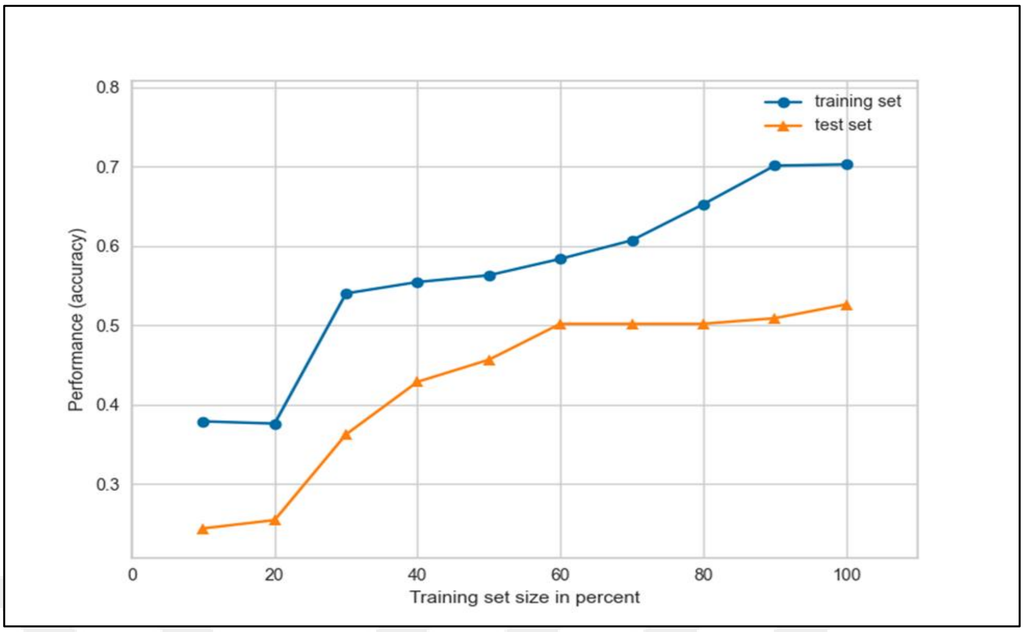

Figure 4.17: Training of KNN Algorithms.

Table 4.8: TF-IDF – 3.

Class	Precession	Recall	F-score	Accuracy
Alzheimer	0.85	0.89	0.87	91.638
Asthma	0.95	1.00	0.98	
Cancer	0.93	0.98	0.96	
Diabetic	0.89	0.85	0.87	
Fabry	0.94	0.92	0.93	
Syndrome	0.93	0.86	0.90	

In this table, the highest value in the Precision was in the Crisis class, which was 0.95, while the lowest value was in the Alzheimer's class, which was 0.85, while in the Recall, the highest value was in the Crisis class, which was 1.00, and the lowest value was in the Syndrome class, which was 0.86, while in the thousand Scores it was 0.86. The highest value was in the crisis class, while the lowest value was in the Alzheimer's and Diabetic classes.

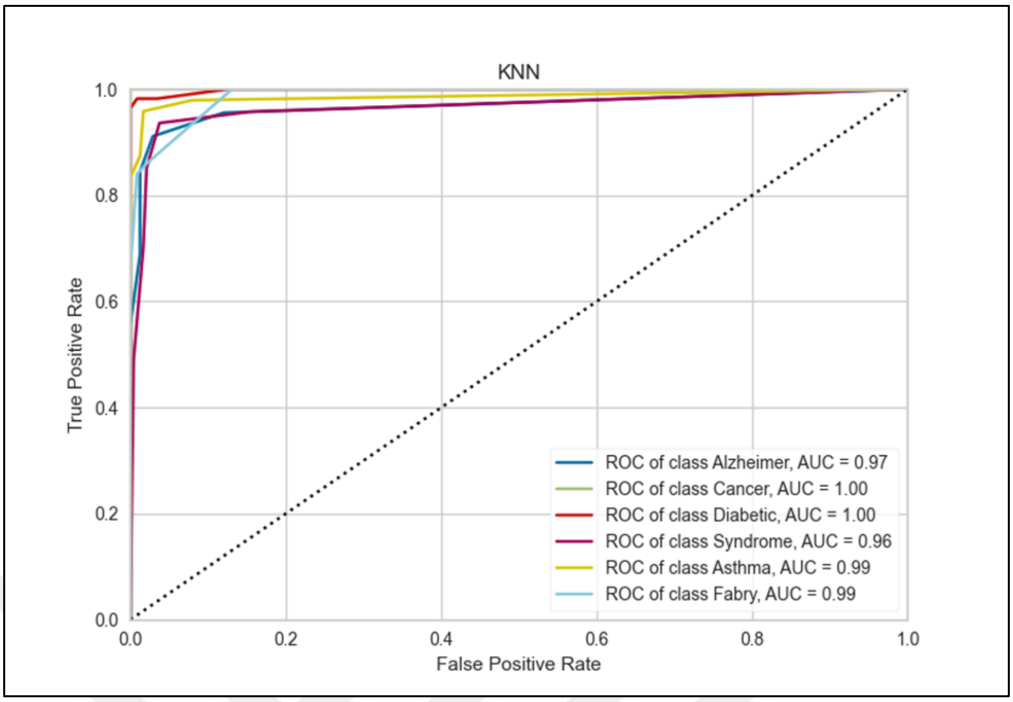

Figure 4.18: ROC of KNN Algorithms.

Here, note that the highest value of the Currency was recorded in the Cancer Class and Diabetic Class, which is 1.00, while the lowest value was recorded in both syndromes, which is 0.96.

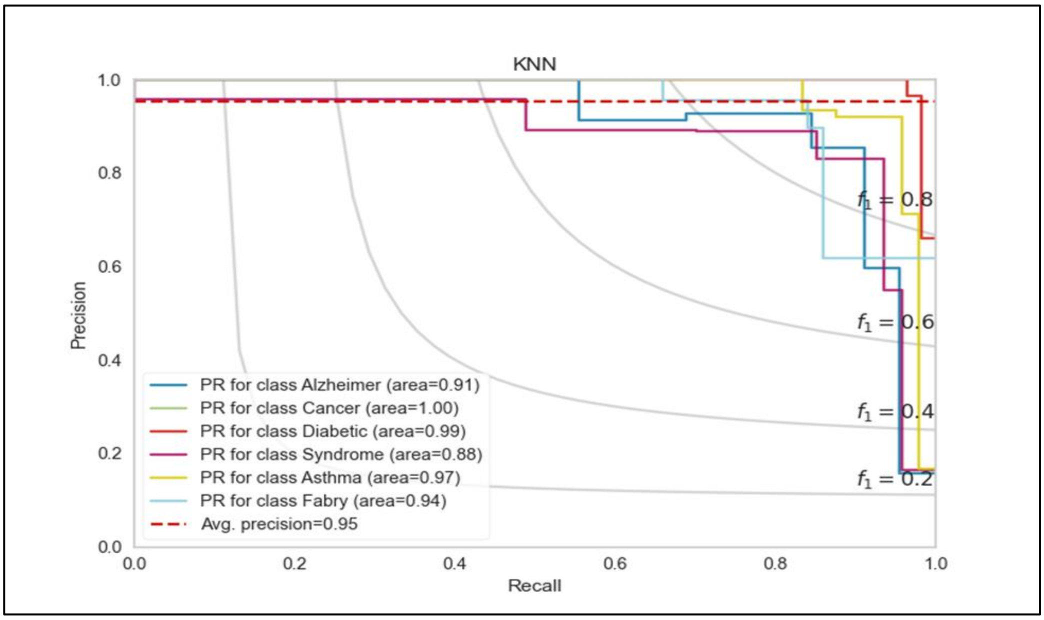

Figure 4.19: PR of KNN Algorithms.

In this drawing, the cancer class recorded the highest value in the PR, which is 1.00, while the lowest value was in the syndrome class, which is 0.94.

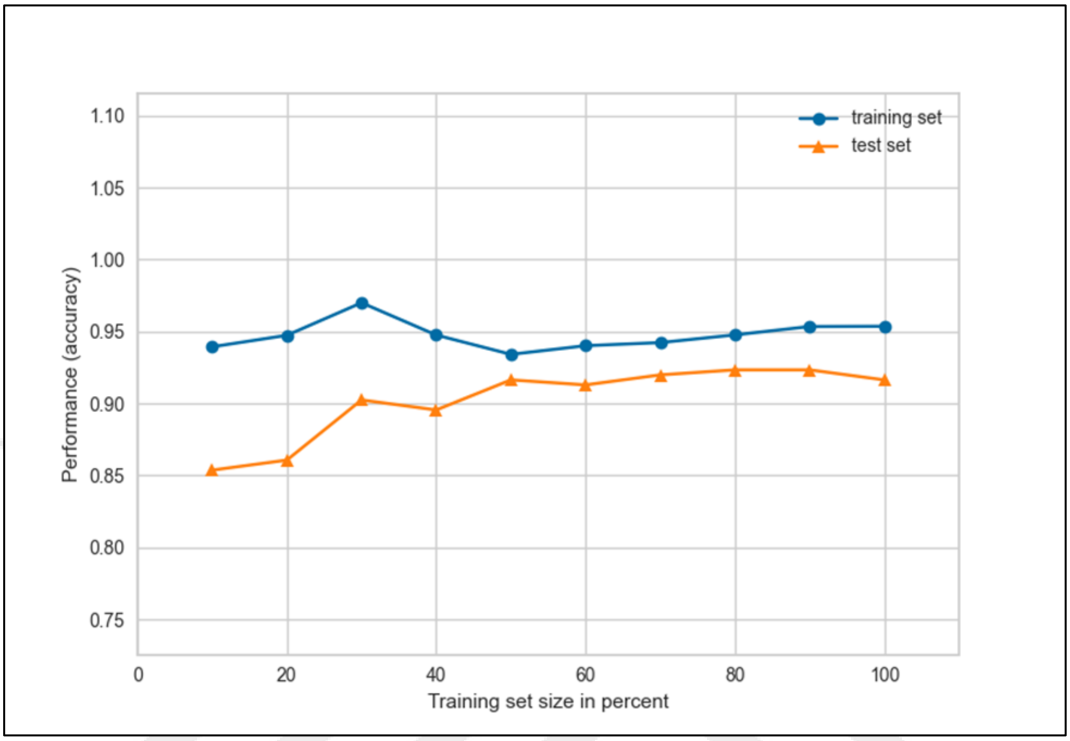

Figure 4.20: Training of KNN Algorithms.

4.5.4 NB

Table 4.9: Term Frequency - 3.

Class	Precession	Recall	F-score	Accuracy
Alzheimer	0.84	0.80	0.82	0.83
Asthma	0.98	1.00	0.99	
Cancer	0.83	0.96	0.89	
Diabetic	0.84	0.77	0.80	
Fabry	0.96	0.54	0.69	
Syndrome	0.61	0.82	0.70	

In this table of the NP algorithm, notice that the highest value for Precision was in the Crisis class, which is 0.98, and the lowest value was in the Cancer class, which is 0.83. In Recall, the highest value was in the Crisis class, which is 1.00. The lowest value was in the Fabry class, which is 0.54; in the Thousands Score, the highest value was in the Crisis class, which is 0.99, and the lowest value was in the Syndrome class, which is 0.70.

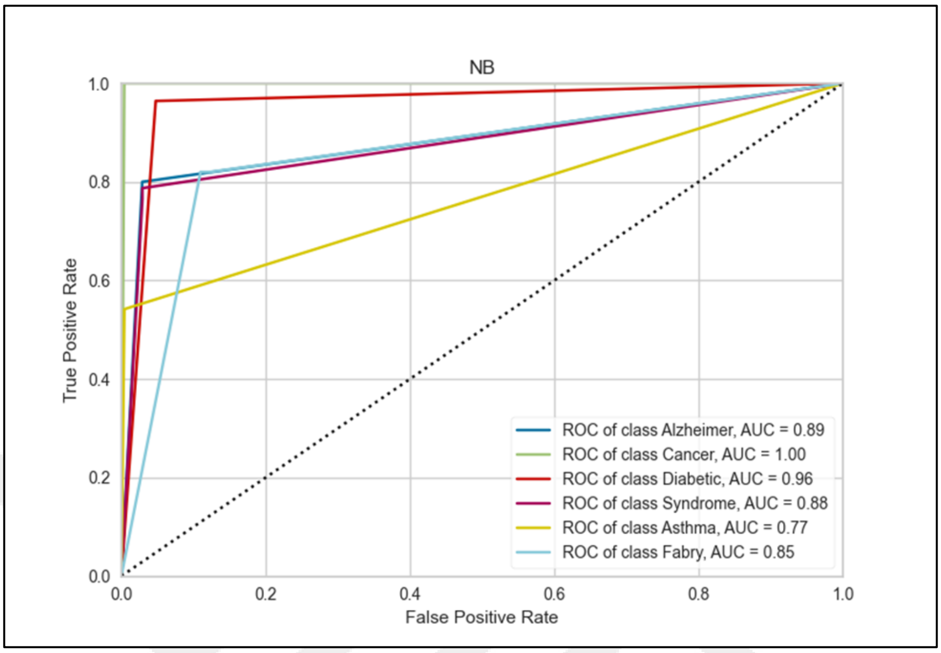

Figure 4.21: ROC of NB Algorithms.

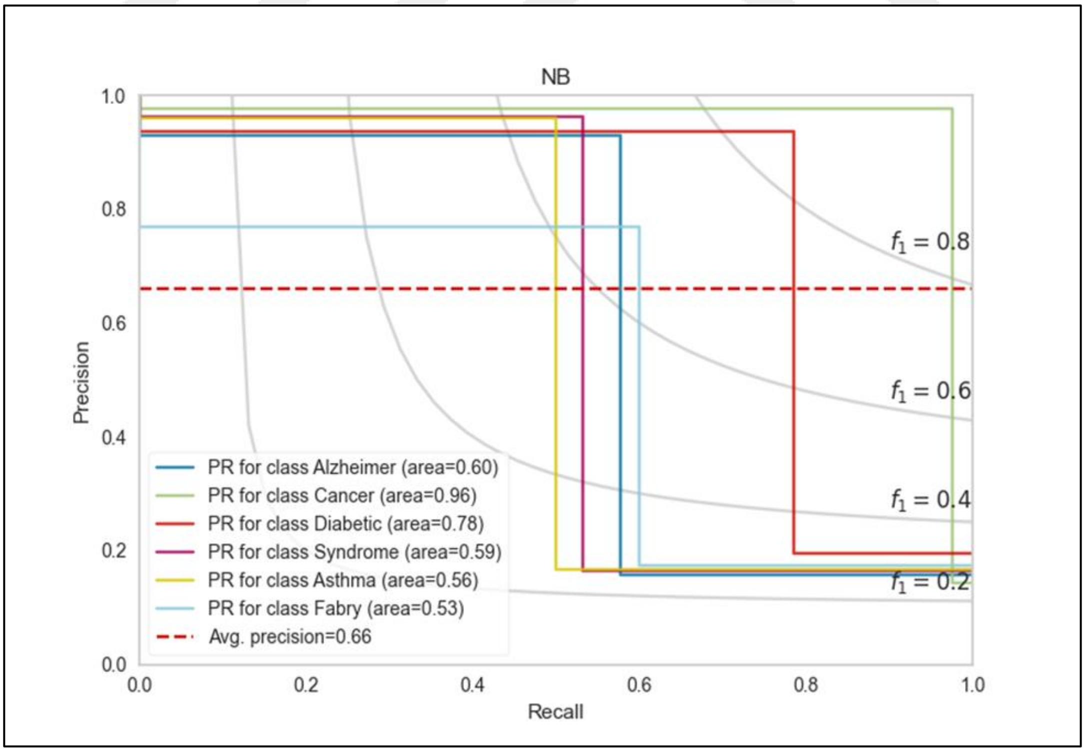

Figure 4.22: PR of NB Algorithms.

Here, in this graph, the PR of the Cancer class recorded the highest value, which is 0.96, and the lowest value was in the Viberian class, which is 0.53, and the rate of precision was 0.66.

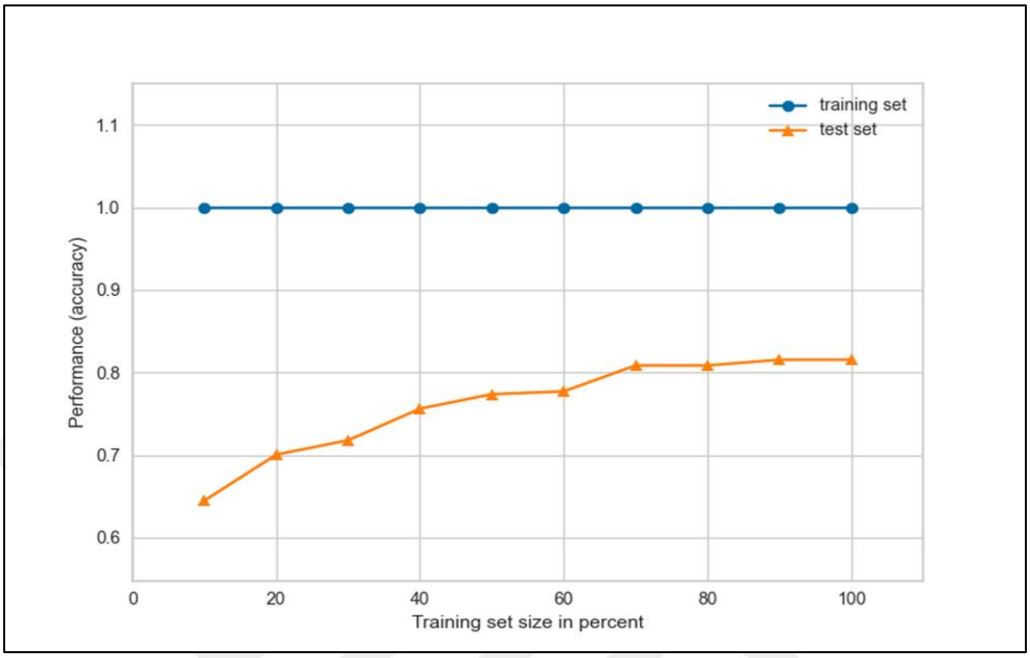

Figure 4.23: Training OF NB Algorithms.

Table 4.10: TF-IDF – 4.

Class	Precession	Recall	F-score	Accuracy
Alzheimer	0.80	0.71	0.75	76.655
Asthma	0.93	0.95	0.94	
Cancer	0.78	0.96	0.86	
Diabetic	0.78	0.68	0.73	
Fabry	0.96	0.48	0.64	
Syndrome	0.56	0.80	0.66	

The Fabry class had the highest precision value of 0.96, while the cancer and diabetic classes had the lowest precision values. In Recall, the highest value was 0.96, which was recorded in the Cancer class. The crisis was at 094, and the lowest value was in the Viberian class, at 0.64.

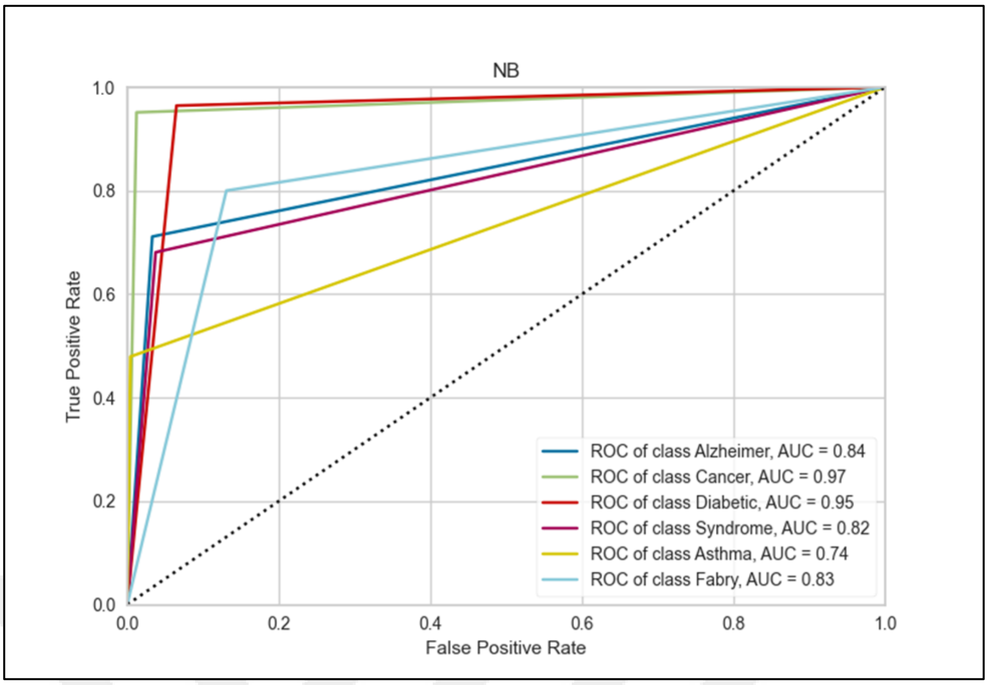

Figure 4.24: ROC OF NB Algorithms.

Currency had the highest value in the dialectic category, at 0.95, and the lowest value in the crisis category, at 0.74.

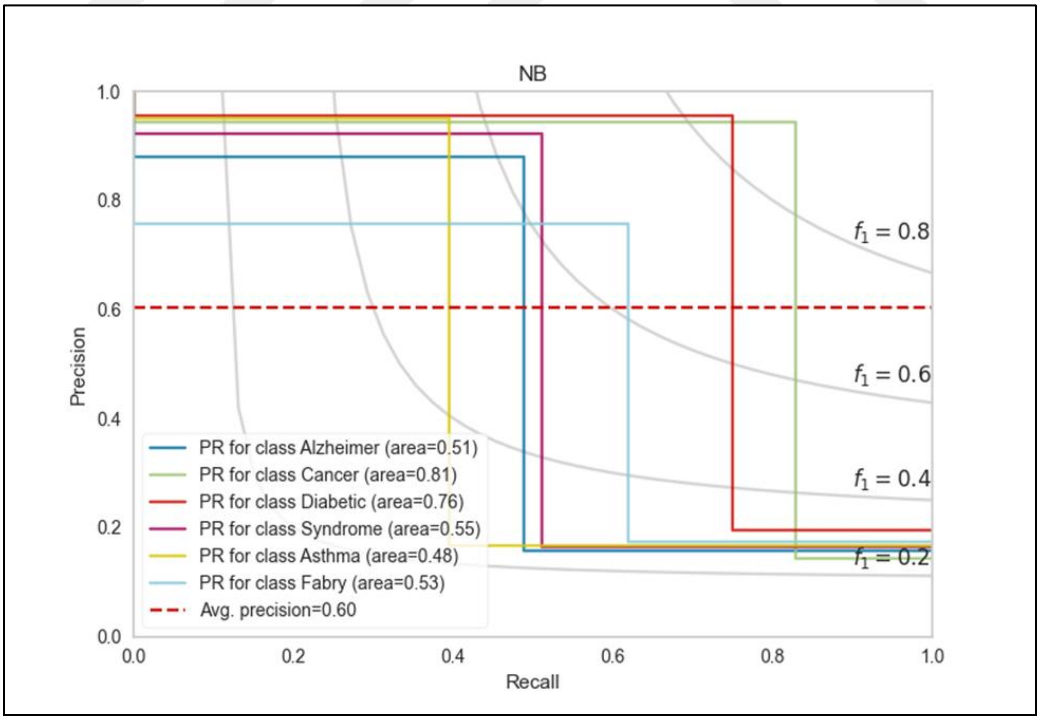

Figure 4.25: PR of NB Algorithms.

Here we note that PR had its highest value in cancer, at 0.81, and the lowest percentage was in crisis class, at 0.48, and the rate of release for precision was 0.60.

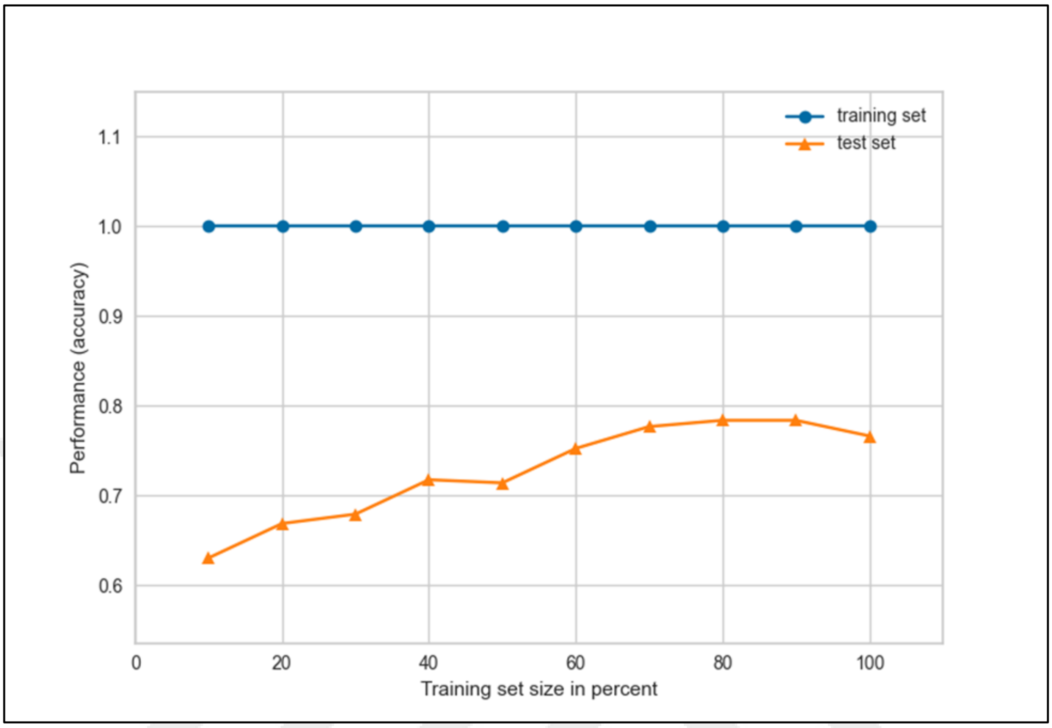

Figure 4.26: Training of NB Algorithms.

5. CONCLUSIONS AND FUTURE WORKS

5.1 CONCLUSIONS

The findings of this thesis lead to a number of generalizations, which are briefly discussed below.

a. This research found that the KNN algorithm is the best way to figure out which way an article is facing. In comparison to other methods, TF-IDF is the most effective method of feature extraction. As a result, KNN worked closely with TF-IDF on dataset V3.
b. A total of five grammes were used in the research, but the machine learning techniques showed the most promise with a single unit of measurement. This occurs because the chosen samples are relatively small.

c. The difficulties (zero relation, low accuracy, and time consuming) associated with machine learning were successfully addressed by the suggested hybrid method that combines rough set theory and lexicon-based approaches.
d. The research found that three TF and TF-IDF feature extraction vectors were the most effective.
e. Using the lower approximation method with human-based evaluation, this research suggested a method for selecting sentence targets that could pull full sentence targets from Arabic political articles (seed words) using the lower approximation method. On the other hand, the lower approximation technique could only choose sentences that had a seed word in them.

The research found that combining the lower approximation method with the sentence frequency technique increased the accuracy of the selection process.

5.2 FUTURE WORK

This dissertation gives a number of ways to deal with the problems of polarity and target extraction in an Arabic political piece. A collection of corpus versions representing real-world case studies has been successfully applied to the suggested methods. What follows is a synopsis of several proposed elements.

a. In the goal method, words were taken from articles by people (using manually made seed words), so research could focus on making an automatic method that works better and can be used in more situations.
b. This research focused on choosing whole target sentences for the target technique. In the future, this process could be improved by trying to pull out just the target from the sentence.
c. In this study, the parameter was used to improve the way the values for polarity were chosen. Several other methods, like cuckoo search, particle swarm optimization, and the firefly algorithm, can also choose the value on their own, but they take a long time to do so, which makes them hard to use.

REFERENCES

[1] R. Stevens, C. Goble, P. Baker, and A. Brass, "A classification of tasks in bioinformatics," Bioinformatics, vol. 17, no. 2, pp. 180-188, 2001.

[2] G. B. Singh, "Introduction to bioinformatics," in Fundamentals of Bioinformatics and Computational Biology: Springer, 2015, pp. 3-10.

[3] M. A. Hearst, "Text data mining: Issues, techniques, and the relationship to information access," in Presentation notes for UW/MS workshop on data mining, 1997, pp. 112-117.

[4] R. Feldman and I. Dagan, "Knowledge Discovery in Textual Databases (KDT)," in KDD, 1995, vol. 95, pp. 112-117.

[5] T. A. Al-asadi, A. J. Obaid, R. Hidayat, and A. A. Ramli, "A survey on web mining techniques and applications," International Journal on Advanced Science Engineering and Information Technology, vol. 7, pp. 1178-1184, 2017.

[6] M. P. Singh, The practical handbook of internet computing. CRC press, 2004.

[7] A. Akilan, "Text mining: Challenges and future directions," in 2015 2nd International Conference on Electronics and Communication Systems (ICECS), 2015, pp. 1679-1684: IEEE.

[8] M. Allahyari et al., "A brief survey of text mining: Classification, clustering and extraction techniques," arXiv preprint arXiv:1707.02919, 2017.

[9] T. N. Phyu, "Survey of classification techniques in data mining," in Proceedings of the International MultiConference of Engineers and Computer Scientists, 2009, vol. 1, pp. 18-20.

[10] V. Krishnaiah, G. Narsimha, and N. S. Chandra, "Survey of classification techniques in data mining," International Journal of Computer Sciences and Engineering, vol. 2, no. 9, pp. 65-74, 2014.

[11] G. Kesavaraj and S. Sukumaran, "A study on classification techniques in data mining," in 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013, pp. 1-7: IEEE.

[12] H. Bozdogan, Statistical data mining and knowledge discovery. CRC Press, 2003.

[13] P. Giudici, Applied data mining: statistical methods for business and industry. John Wiley & Sons, 2005.

[14] J. Lepping, "Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery," 2018.

[15] R. Mikut and M. Reischl, "Data mining tools," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 1, no. 5, pp. 431-443, 2011.

[16] T. B. Ho, "Knowledge discovery," Knowledge Science: Modeling the Knowledge Creation Process, p. 57, 2011.

[17] M. K. Saad and W. M. Ashour, "Arabic text classification using decision trees," Arabic text classification using decision trees, vol. 2, 2010.

[18] R. Ramya, K. Venugopal, S. Iyengar, and L. Patnaik, "Feature extraction and duplicate detection for text mining: A survey," Global Journal of Computer Science and Technology, 2017.

[19] S. Inzalkar and J. Sharma, "A survey on text mining-techniques and application," International Journal of Research In Science & Engineering, vol. 24, pp. 1-14, 2015.

[20] A. B. Soliman, K. Eissa, and S. R. El-Beltagy, "Aravec: A set of arabic word embedding models for use in arabic nlp," Procedia Computer Science, vol. 117, pp. 256-265, 2017.

[21] R. Gull, U. Shoaib, S. Rasheed, W. Abid, and B. Zahoor, "Pre processing of twitter's data for opinion mining in political context," Procedia Computer Science, vol. 96, pp. 1560-1570, 2016.

[22] A. Al Oudah, K. Al Bassam, H. Kurdi, and S. Al-Megren, "Wajeez: An Extractive Automatic Arabic Text Summarisation System," in International Conference on Human-Computer Interaction, 2019, pp. 3-14: Springer.

[23] M. A. Alhanjouri, "Pre Processing Techniques for Arabic Documents Clustering," Pre Processing Techniques for Arabic Documents Clustering, vol. 7, no. 2, 2017.

[24] Z. Yao and C. Ze-wen, "Research on the construction and filter method of stop-word list in text preprocessing," in 2011 Fourth International Conference on Intelligent Computation Technology and Automation, 2011, vol. 1, pp. 217-221: IEEE.

[25] D. D. Palmer, "Tokenisation and sentence segmentation," Handbook of natural language processing, pp. 11-35, 2000.

[26] Y. A. Alhaj, M. A. Al-qaness, A. Dahou, M. A. Elaziz, D. Zhao, and J. Xiang, "Effects of Light Stemming on Feature Extraction and Selection for Arabic Documents Classification," in Recent Advances in NLP: The Case of Arabic Language: Springer, 2020, pp. 59-79.

[27] L. Carstens and F. Toni, "Using argumentation to improve classification in natural language problems," ACM Transactions on Internet Technology (TOIT), vol. 17, no. 3, pp. 1-23, 2017.

[28] L. S. Larkey, L. Ballesteros, and M. E. Connell, "Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis," in Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, 2002, pp. 275-282.

[29] A. Oussous, A. A. Lahcen, and S. Belfkih, "Impact of text pre-processing and ensemble learning on Arabic sentiment analysis," in Proceedings of the 2nd International Conference on Networking, Information Systems & Security, 2019, pp. 1-9.

[30] D. H. Abd, A. T. Sadiq, and A. R. Abbas, "Political articles categorization based on different naïve bayes models," in International Conference on Applied Computing to Support Industry: Innovation and Technology, 2019, pp. 286-301: Springer.

[31] N. Shah and S. Mahajan, "Document clustering: a detailed review," International Journal of Applied Information Systems, vol. 4, no. 5, pp. 30-38, 2012.

[32] I. A. El-Khair, "Effects of stop words elimination for Arabic information retrieval: a comparative study," arXiv preprint arXiv:1702.01925, 2017.

[33] P. Schäuble, Multimedia information retrieval: content-based information retrieval from large text and audio databases. Springer Science & Business Media, 2012.

[34] N. Hardeniya, J. Perkins, D. Chopra, N. Joshi, and I. Mathur, Natural Language Processing: Python and NLTK. Packt Publishing Ltd, 2016.

[35] J. Singh and V. Gupta, "Text stemming: Approaches, applications, and challenges," ACM Computing Surveys (CSUR), vol. 49, no. 3, pp. 1-46, 2016.

[36] F. Harrag, E. El-Qawasmah, and A. M. S. Al-Salman, "Stemming as a feature reduction technique for arabic text categorization," in 2011 10th International Symposium on Programming and Systems, 2011, pp. 128-133: IEEE.

[37] F. S. Al-Anzi and D. AbuZeina, "Stemming impact on Arabic text categorization performance: A survey," in 2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA), 2015, pp. 1-7: IEEE.

[38] R. Z. Al-Abdallah and A. T. Al-Taani, "Arabic text summarization using firefly algorithm," in 2019 Amity International Conference on Artificial Intelligence (AICAI), 2019, pp. 61-65: IEEE.

[39] N. Mansour, R. A. Haraty, W. Daher, and M. Houri, "An auto-indexing method for Arabic text," Information processing & management, vol. 44, no. 4, pp. 1538-1545, 2008.

[40] M. R. A. Utomo and Y. Sibaroni, "Text Classification of British English and American English Using Support Vector Machine," in 2019 7th International Conference on Information and Communication Technology (ICoICT), 2019, pp. 1-6: IEEE.

[41] J. Savoy, "A stemming procedure and stopword list for general French corpora," Journal of the American Society for Information Science, vol. 50, no. 10, pp. 944-952, 1999.

[42] V. Gupta, N. Joshi, and I. Mathur, "Advanced Machine Learning Techniques in Natural Language Processing for Indian Languages," in Smart Techniques for a Smarter Planet: Springer, 2019, pp. 117-144.

[43] M. N. Kassim, S. H. M. Jali, M. A. Maarof, and A. Zainal, "Towards stemming error reduction for Malay texts," in Computational Science and Technology: Springer, 2019, pp. 13-23.

[44] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," International Journal of Machine Learning and Cybernetics, vol. 1, no. 1-4, pp. 43-52, 2010.

[45] D. Sarkar, "Text Analytics with Python," 2016.

[46] J. C. Martineau and T. Finin, "Delta tfidf: An improved feature space for sentiment analysis," in Third international AAAI conference on weblogs and social media, 2009.

[47] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2, 2012, pp. 90-94: Association for Computational Linguistics.

[48] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," Journal of documentation, 1972.

[49] H. Ren, Z. Zeng, Y. Cai, Q. Du, Q. Li, and H. Xie, "A weighted word embedding model for text classification," in International Conference on Database Systems for Advanced Applications, 2019, pp. 419-434: Springer.

[50] L. Khreisat, "Arabic Text Classification Using N-Gram Frequency Statistics A Comparative Study," DMIN, vol. 2006, pp. 78-82, 2006.

[51] N. Al-Twairesh, H. Al-Khalifa, and A. Al-Salman, "Subjectivity and sentiment analysis of Arabic: trends and challenges," in 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA), 2014, pp. 148-155: IEEE.

[52] S. ChandraKala and C. Sindhu, "Opinion mining and sentiment classification: A survey," ICTACT journal on soft computing, vol. 3, no. 1, pp. 420-425, 2012.

[53] F. Hemmatian and M. K. Sohrabi, "A survey on classification techniques for opinion mining and sentiment analysis," Artificial Intelligence Review, pp. 1-51, 2017.

[54] Z. Yan, X. Jing, and W. Pedrycz, "Fusing and mining opinions for reputation generation," Information Fusion, vol. 36, pp. 172-184, 2017.

[55] S. L. Lo, E. Cambria, R. Chiong, and D. Cornforth, "Multilingual sentiment analysis: from formal to informal and scarce resource languages," Artificial Intelligence Review, vol. 48, no. 4, pp. 499-527, 2017.

[56] J. Sun, G. Wang, X. Cheng, and Y. Fu, "Mining affective text to improve social media item recommendation," Information Processing & Management, vol. 51, no. 4, pp. 444-457, 2015.

[57] R. Xia, F. Xu, J. Yu, Y. Qi, and E. Cambria, "Polarity shift detection, elimination and ensemble: A three-stage model for document-level sentiment analysis," Information Processing & Management, vol. 52, no. 1, pp. 36-45, 2016.

[58] D. Marcheggiani, O. Täckström, A. Esuli, and F. Sebastiani, "Hierarchical multi-label conditional random fields for aspect-oriented opinion mining," in European conference on information retrieval, 2014, pp. 273-285: Springer.

[59] B. Yang and C. Cardie, "Context-aware learning for sentence-level sentiment analysis with posterior regularization," in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2014, pp. 325-335.

[60] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, 2002, pp. 79-86: Association for Computational Linguistics.

[61] R. Moraes, J. F. Valiati, and W. P. G. Neto, "Document-level sentiment classification: An empirical comparison between SVM and ANN," Expert Systems with Applications, vol. 40, no. 2, pp. 621-633, 2013.

[62] P. Gowtamreddy, "Opinion mining of online customer reviews," 2014.

[63] K. Yang, N. Yu, H. Zhang, and A. Valerio, "WIDIT in TREC-2006 Blog track," 2006.

[64] D. Maynard and A. Funk, "Automatic detection of political opinions in tweets," in Extended Semantic Web Conference, 2011, pp. 88-99: Springer.

[65] G. Li and F. Liu, "Sentiment analysis based on clustering: a framework in improving accuracy and recognizing neutral opinions," Applied intelligence, vol. 40, no. 3, pp. 441-452, 2014.

[66] V. Sindhwani and P. Melville, "Document-word co-regularization for semi-supervised sentiment analysis," in 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 1025-1030: IEEE.

[67] X. Wu and V. Kumar, The top ten algorithms in data mining. CRC press, 2009.

[68] D. Zhang, "Bayesian Classification," in Fundamentals of Image Data Mining: Springer, 2019, pp. 161-178.

[69] S. Raschka, "Naive bayes and text classification i-introduction and theory," arXiv preprint arXiv:1410.5329, 2014.

[70] S. Geetha and R. Maniyosai, "An Improved Naive Bayes Classifier on Imbalanced Attributes," International Journal of Organizational and Collective Intelligence (IJOCI), vol. 9, no. 2, pp. 1-15, 2019.

[71] S. Xu, "Bayesian Naïve Bayes classifiers to text classification," Journal of Information Science, vol. 44, no. 1, pp. 48-59, 2018.

[72] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in AAAI-98 workshop on learning for text categorization, 1998, vol. 752, no. 1, pp. 41-48: Citeseer.

[73] N. Sharma and M. Singh, "Modifying Naive Bayes classifier for multinomial text classification," in 2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE), 2016, pp. 1-7: IEEE.

[74] J. D. Rennie, L. Shih, J. Teevan, and D. R. Karger, "Tackling the poor assumptions of naive bayes text classifiers," in Proceedings of the 20th international conference on machine learning (ICML-03), 2003, pp. 616-623.

[75] B. Liu, Web data mining: exploring hyperlinks, contents, and usage data. Springer Science & Business Media, 2007.

[76] N. Deng and Y. Tian, "Support Vector Machines: A New Method in Data Mining," Science Press, 2004.

[77] I. S. Al-Mejibli, D. H. Abd, J. K. Alwan, and A. J. Rabash, "Performance evaluation of kernels in support vector machine," in 2018 1st Annual International Conference on Information and Sciences (AiCIS), 2018, pp. 96-101: IEEE.

[78] A. A. Christopher and S. A. alias Balamurugan, "Prediction of warning level in aircraft accidents using data mining techniques," The Aeronautical Journal, vol. 118, no. 1206, pp. 935-952, 2014.

[79] C. Shahabi, M. R. Kolahdouzan, and M. Sharifzadeh, "A Road Network Embedding Technique for K-Nearest Neighbor Search in Moving Object Databases," GeoInformatica, journal article vol. 7, no. 3, pp. 255-273, September 01 2003.

[80] H. Parvin, H. Alizadeh, and B. Minaei-Bidgoli, "MKNN: Modified k-nearest neighbor," in Proceedings of the World Congress on Engineering and Computer Science, 2008, vol. 1: Citeseer.

[81] P. A. Jaskowiak and R. Campello, "Comparing correlation coefficients as dissimilarity measures for cancer classification in gene expression data," in Proceedings of the Brazilian Symposium on Bioinformatics, 2011, pp. 1-8: Brasília.

[82] N. A. Nnamoko, F. N. Arshad, D. England, and J. Vora, "Meta-classification model for diabetes onset forecast: A proof of concept," in Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on, 2014, pp. 50-56: IEEE.

[83] H. M. Tran, S. Van Nguyen, S. T. Le, and Q. T. Vu, "Fault data analytics using decision tree for fault detection," in International Conference on Future Data and Security Engineering, 2015, pp. 57-71: Springer.

[84] P. Nevlud, M. Bures, L. Kapicak, and J. Zdrálek, "Anomaly-based network intrusion detection methods," Advances in Electrical and Electronic Engineering, vol. 11, no. 6, p. 468, 2013.

[85] S. Sayad, "Comparing Different Classification Techniques in Credit Scoring," [Online], Available:https://www.sas.com/content/dam/SAS/en_ca/User%20Group%20Presentations/Toronto-Data-Mining-Forum/SaedSayad-Credit%20ScoringiSmartsoft.pdf Accessed [09 Jun 2018].