School of Information Technology and
Engineering at the
ADA University

School of Engineering and Applied
Science at the
George Washington University

LEXICON-BASED APPROACH TO THE SENTIMENT AND EMOTION ANALYSIS IN AZERBAIJANI LANGUAGE

A Thesis
Presented to the Graduate Program of Computer Science and Data Analytics
of the School of Information Technology and Engineering
ADA University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Computer Science and Data Analytics
ADA University

By
Kheybar Mammadnaghiyev

April, 2022

THESIS ACCEPTANCE

This Thesis by: Kheybar Mammadnaghiyev
Entitled: *Lexicon-based approach Sentiment and Emotion Analysis in Azerbaijani Language*

has been approved as meeting the requirement for the Degree of Master of Science in Computer Science and Data Analytics of the School of Information Technology and Engineering, ADA University.

Approved:

Samir Rustamov	Rustamov S	28.04.2022
(Adviser)		(Date)
Abzatdin Adamov		28.04.2022
(Program Director)		(Date)
Sencer Yeralan		28.04.2022
(Dean)		(Date)

ABSTRACT

Sentiment Analysis of a text and Emotion Analysis of a text in the Azerbaijani language has been recently at the center of attention as a research topic. Although different approaches have been applied to this topic, there are still many approaches exist that can be applied to it. In this research, the lexicon-based approach is applied to both Sentiment and Emotion Analysis. Briefly, sentiment analysis deal with whether a text is positive, negative, or neutral, while emotion analysis detects one of the basic emotions. Note that, in this research basic emotions of Plutchik's Wheel of Emotions are used in order to categorize emotions which are anger, anticipation, disgust, fear, surprise, trust, sadness, and joy. This research paper used 24,000 lines of news in Azerbaijani for training and testing the models and used 66,000 lines of newly created Azerbaijani word dictionary dataset in order to apply a lexicon-based approach. From the implementation point of view, Decision Trees, Naïve Bayes, Support Vector Machine, Neural Network, and BERT machine learning algorithm are applied and compared. Moreover, this research also deals with entity recognition. It detects the corresponding sentiment and list of emotions of the main entity of the text. Note that, in this research, entity recognition is accomplished using a rule-based approach.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Explanation
BOW	Bag Of Words
LD	Lexicon Dictionary
TF-IDF	Term Frequency Inverse Document Frequency
SVM	Support Vector Machine
ANN	Artificial Neural Network
BERT	Bidirectional Encoder Representation for Transformers

1 INTRODUCTION

Technology has advanced so much in recent years that the internet has become an indispensable part of our daily life. People of today cannot imagine their life without technology gadgets as these gadgets are in their use every single day. The internet has given us people so many opportunities that these opportunities have yielded a major change in the way life was designed. These changes are observed in almost every area of life from simple daily habits to education and research areas in a chain-like way. For instance, checking out social media is a simple daily habit for the majority of people where they share how they feel, and what they think about various topics. And this daily habit yielded a big research area which is detecting sentiment and emotion in a given text. Although there is a massive amount of data in social media for detecting sentiment and emotion, this research area is not bounded to only social media data, but any given text. The analysis of the text is performed in the field of sentiment and emotion analysis to detect the general attitude in it. The results of this analysis can be utilized to improve the quality of products or to better understand public opinion on various topics. Such analysis is a big aid for especially decision-making as it fastens to understand the main point of view on a specific topic. For instance, people usually write their feedback after they use a particular service or a particular product from a company, but when there is a vast amount of feedback, it creates a big delay for the company to make a decision by actually reading them to evaluate their service or product. However, using sentiment and emotion analysis, the process of decision-making fastens drastically. Sentiment and Emotion Analysis is one of the research areas that has recently gotten a lot of attention, and many research projects have begun in this field. Although there are recent improvements in the area, just a few sample studies on the Azerbaijani language exist for now. This paper is about research that is based on the news articles dataset in Azerbaijani and a lexicon dictionary dataset in Azerbaijani. In this research different machine learning algorithms such as Decision Trees, Naive Bayes, and Artificial Neural Network, different vectorization methods such as CountVectorizer and TF-IDF vectorizer have been applied. This research covers entity-level-based sentiment analysis and emotion analysis using one of the Bag of Words and Lexicon-based methods.

1.1 Definition of the problem

In this paper, the word problem stands for the sentiment analysis and the emotion analysis in Azerbaijani texts. Although, sentiment analysis and emotion analysis seem to be similar concepts, when looked at in detail, they outweigh sharing differences over similarities. Indeed, the word sentiment refers to a point of view, whereas emotion refers to a feeling based on the mood. This definition already clarifies that emotion analysis has a bigger aspect than sentiment analysis as there can be easily many emotions in a text, but it mainly has a single sentiment.

In general, sentiment analysis is the method of recognizing positive, negative, or neutral sentiment in text. In some cases, research for sentiment analysis is downsized into binary, hence making it to recognize only one of positive or negative. So, in the simplest term, sentiment analysis defines whether an online text is about something good, or something bad, or something

unbiased. Here online text can be a blog, news, application reviews, a written post on a social media platform, and so on. The point is each of these written texts demonstrates its author's perspective, hence making it a necessary piece of information.

Whereas emotion analysis is the method of recognizing the mood in text. Recognizing mood in the text is a broader area and it can be classified into emotions in many different ways. Recognizable emotions for emotion analysis may include happiness, sadness, anger, fear, and many others. Although in sentiment analysis, the result values are easily categorized, this is not the case for emotions as there are so many of them. Throughout history, there have been different types of classifications for emotions. Among them, the following three classifications are the ones with the most prevalent perspectives. Here it includes Ekman's theory of fundamental emotions, Russell's circumplex model, and Plutchik's wheel of emotion.

1.2 The objective of the study

The objective of this study is to use machine learning techniques to detect the sentiment and emotion values of an entity in a text in Azerbaijani. The need for the Sentiment and Emotion Analysis increases as the online text data increases every single day. This also applies to Azerbaijani textual content as the technology has advanced so much in the last decade here and the people of Azerbaijan create online textual content in the Azerbaijani language every day. Thus, analyzing Azerbaijani textual data is in need, and such a tool would become extremely useful. The analysis can be done on any textual data, but it would mainly be used for news, customer feedback, and textual social media posts such as tweets. For instance, by using political news data, it can be detected in seconds that the attitude towards that specific politics is positive, negative, or neutral, and also what are the emotions that people feel about this specific political news. Of course, here politics stands for a mere instance as this analysis applies with no boundary to any type of text.

This study includes several applied machine learning algorithms, sentiment analysis of a text in Azerbaijani, emotion analysis of a text in Azerbaijani, and also entity recognition of a text in Azerbaijani with its corresponding sentiment and emotion values.

1.3 Significance of the problem

The significance of the Sentiment Analysis and Emotion Analysis for the Azerbaijani language is in noticeable need. It is mainly beneficial for the companies, and also the public. Almost every company has some kind of service or product that is open to being evaluated by customers in the form of written feedback. Without machine learning, analyzing this feedback in a short time is not efficient which puts the decision-making tasks for the company in a big delay. However, using Sentiment and Emotion Analysis companies gain insight into customers' reactions and detect the common attitude towards a service or a product of a company in a short period of time. This is an important factor for the integrity of the enterprise in order to be successful. When there is no such kind of analysis, the written feedback might go unnoticed as it is a time-taking task to read them one by one, which would result in the loss of interest in the certain product/service of a company. By identifying the good and bad points, companies can improve their negative elements, and emphasize more on the positive elements.

On the other hand, the public also can benefit from this type of analysis for personal interests. For instance, they can check the latest news on the program and see generally what type of sentiment and mood of things happen right now and so on. Sentiment and Emotion Analysis also offers its users to know exactly what entity is good or bad in the product/service/news, and also what is the entity that makes people feel angry, sad, or any other type of emotion.

So, this paper solves multiple mentioned problems and brings improvements in mentioned particular areas.

1.4 Assumptions and Limitations

Sentiment and Emotion Analysis is a type of research area that has been done in various languages successfully, but it lacks this variety in the Azerbaijani language. Although there are already a few pieces of research on only Sentiment Analysis in the Azerbaijani language, there were none for the Emotion Analysis in the Azerbaijani language until this research. So, this improvement can only enlarge the development of textual data analysis in Azerbaijan and may be applied in areas that are previously mentioned in this paper.

Despite the good assumptions of this research, some limitations may create some kind of barrier along the way. First and foremost, the Azerbaijani language is not as much used in machine learning algorithms as the most vital languages. Although this is not going to be a problem in the near future, it might arise some problems for now. As the Azerbaijani language uses plenty of suffixes, the need for stemming is inevitable in order to get more precise results. And finding the stem of the words is not always correct, because sometimes the stem of words over-collapse which yields to have the wrong stemming for the word. Another limitation mainly applies to emotion analysis which is about different opinions of people. Everyone might agree that one piece of text might have several emotions at the same time. But those emotions might not be exactly the same ones from everyone's point of view. Moreover, the same problem is also valid for entities in a text. Detecting what is the main entity in a text is even sometimes hard for humans, so the machine cannot always find the one main entity, or the list of main entities listed in a given text.

In this research, these assumptions and limitations are considered and applied correctly as much as possible, and they are covered comprehensively later in this paper.

2 REVIEW OF THE LITERATURE

Sentiment Analysis and Emotion Analysis is a type of research area in which there are more than a few types of research that use various aspects. In all of these research methodologies, Sentiment Analysis identifies whether the given word is positive, negative, or neutral using various methods. Sometimes, this process is downsized to binary classification which removes the feature neutral and keeps only the features positive and negative. Emotion Analysis had not been researched as much as Sentiment Analysis, but there are several approaches for classifying emotions in order to later utilize it in Emotion Analysis. These methods are using either unsupervised or supervised learning.

Sentiment Analysis using unsupervised learning detects the sentiment of a text-based on an unsupervised dictionary. An unsupervised dictionary is created using the word-seeding method. It is

also concluded that methods such as Word2Vec perform well when unsupervised learning is applied [1].

Although unsupervised learning is applicable to Sentiment Analysis, it performs even better when supervised learning techniques are used [2]. Unsupervised Learning methods are covered by He et al. [3] using Latent Dirichlet Allocation.

Supervised learning shares various techniques that are applicable to Sentiment and Emotion Analysis. Moreover, it does not make sense to seek the best technique that outweighs every other method because it changes depending on the data, context, and many other parameters. The application of supervised learning methods to Sentiment Analysis is pretty viral among the most-known languages such as English. However, there are only a few experiments in other local languages including Azerbaijani. As for the Emotion Analysis, there is no such deep experiment for the Azerbaijani texts.

The very first step in doing a Sentiment or Emotion Analysis is to identify the desired dataset. Among the existed research for Sentiment Analysis in English, the most popular datasets are movie reviews, tweets, or feedback on a particular service. There are also similar experiments on English news data. Different aspects exist for the way the data is pre-processed and labeled in each of these research methodologies. For instance, Godbole et al. [4] used movie reviews dataset based on its ratings. Pang et al. [5] used movie reviews collected from IMDb. In this research, the Support Vector Machine is applied and considered a successful method for Sentiment Analysis. Support Vector Machine is also examined by Kaya et al. [7] and got 90% accuracy on the dataset of Turkish tweets. As this language is similar to the Azerbaijani language, it is likely that it also predicts good results in the Azerbaijani language. Sentiment Analysis in Twitter data is also examined by Severyn et al. [8] where neural networks techniques are applied. In terms of other types of datasets, Belahur et al. used an English news dataset in order to divide the news into positive and negative groups.

In general terms, there are 3 main approaches for Sentiment and Emotion Analysis of a text [9]:

- Lexicon-based Approach
- Machine Learning Approach
- Hybrid Approach

These approaches might perform better than others on a particular dataset and a particular method applied. Azam et al. [9] analyzed both Sentiment and Emotion Analysis in English and Urdu languages and compared others' work in this field with these particular languages. This research also concludes that the emotion categorization varies in different research methods. For instance, Poria et al. [10] categorize emotions into 6 main categories, namely anger, joy, sadness, disgust, surprise, and fear, while Mohammad et al. [11] included two more categories – anticipation and trust in this analysis. This categorization is decided based on the existed Emotion Wheels and Circumplexes which are covered by Kim et al. [12] thoroughly. They investigated that the most known emotion categorization techniques are the following ones:

- Ekman's Theory of Basic Emotions
- Plutchik's Wheel of Emotions
- Russel's Circumplex Model

Different Emotion Analysis research is mainly based on one of these three categorizations for data labeling, whereas there are those who used other techniques. For instance, Mohammad et al. [11] used the basic 8 emotions of Plutchik's Wheel of Emotions, while Carlo et al. [13] used the basic 6 emotions of Ekman's model which are happiness, sadness, fear, disgust, anger, and surprise.

In terms of possible techniques, Patil et al. [14] made comparison-based research for some of the techniques. They made an investigation of the methods such as Naïve Bayes, SVM, Decision Tree, and Semantic Orientation, and concluded with some advantages and disadvantages of using these methods on Sentiment Analysis. They stated the importance of opinion mining and ways to deal with it in this paper.

Behdenna et al. [15] specified their research on document-level sentiment analysis. They have divided the survey into four main aspects which are approach-based, technique-based, data-sources-based, and feature-construction-based surveys. Later Azam et al. [8] decided to categorize sentiment analysis into three main approaches as mentioned previously, Behdenna et al. decided to make four categories by having an additional Deep Learning Approach alongside Machine Learning Approach.

Shirsath et al. [16] covered their sentiment analysis research on news articles in order to determine sentiment in them. Their investigation outstands among others to some extent because they proposed a new ML-based technique. They have compared the performance of SVM and Naïve Bayes methods and concluded that although both of these methods are performing well, Naïve Bayes outperforms SVM by having two more percentage accuracy which is 96%. Jurafsky et al. [17] have made deeper research on the Naïve Bayes method of Sentiment Analysis. In this paper, the Bag of Words model is used and explained comprehensively, also the Naïve Bayes method is explained step by step from the way how to train the data for Naïve Bayes to the ways for optimization. This research also includes test sets, cross-validation, and statistical significance testing including methods such as the Paired Bootstrap Test for making better classifications for sentiment analysis.

Another noticeable survey has been made by Medhat et al. [18]. This survey explores the most known sentiment analysis techniques in a very structured way. This survey includes short information about different techniques for Sentiment and Emotion Analysis. This paper has analyzed the possible techniques without comparison or any specific dataset. It is more of a survey of what has been already done in this field in a nutshell, which is a great paper to refer to when this information is needed.

Pozzi et al. [19] focused on social media datasets using feature-based sentiment analysis. Some researchers conclude the levels of sentiment analysis with three aspects [6] [20], whereas this paper included this fourth one.

As for the work on Azerbaijani texts, there are significant research already made on this area. One research is done by Rustamov et al. [21] who used Neuro-fuzzy and Hidden Markov Models of the text for Sentiment Analysis. They used a movie review as a dataset and used FSC, ANFIS, HMM for the classification. Among these three methods, ANFIS have the most accuracy at 83%. They also combined these methods under the name of Hybrid-1 and Hybrid-2 and got a 0.9% higher result when used Hybrid-2 method. Another noticeable research is done by Hasanli et al. [22]. They applied Naïve Bayes, SVM and Logistic Regression for Sentiment Analysis of a text in Azerbaijani. This

researches focused on the tweets data in Azerbaijani language and get the most accurate result with Naïve Bayes at 93%.

3 RESEARCH APPROACH AND METHODOLOGY

This paper is about entity-level-based Sentiment and Emotion Analysis for the Azerbaijani language. In this research, several machine learning algorithms, several levels of sentiment and emotion analysis, several vectorization methods, and several entity recognition tools are analyzed, and desired ones are applied for both sentiment analysis of the text and also for emotion analysis of the text.

3.1 Levels of Sentiment Analysis

Sentiment Analysis classifies a written text into three main categories using Natural Language Processing. NLP, short for Natural Language Processing, is a field that deals with the classification of text. It is an important part of Machine Learning and is widely used in a range of fields. As for the classification of sentiment analysis, there are three main classification elements which are positive, negative, and neutral. Besides this classification, sentiment analysis can be in one of the following levels:

- Document Level Sentiment Analysis
- Sentence Level Sentiment Analysis
- Word Level Sentiment Analysis
- Feature Level Sentiment Analysis

3.1.1. Document Level Sentiment Analysis

Sentiment Analysis at Document Level analyzes documents as a whole where a document is a set of paragraphs containing sentences. Here the analysis simply neglects all the individual entities in the document and focuses on the text, only in one piece. So, at this stage, the goal is to figure out what the general sentiment of the complete document is. This level of sentiment analysis has been used and applied many times and is considered the simplest level. Although it is mostly used, it has several drawbacks. For instance, if several sentiments are included in one document, this level of sentiment analysis cannot produce several sentiment results as it views the entire document as one topic. Note that, Document Level Sentiment Analysis is also applicable to the Emotion Analysis.

3.1.2. Sentence Level Sentiment Analysis

Sentiment Analysis at Sentence Level analyzes given text by its sentences. The main idea behind this level of analysis is to identify the sentiment of each sentence in the text. As mentioned, one whole document might easily have several sentiments at the same time if there is a contradiction in the document. This problem is eliminated to some extent when sentence-level sentiment analysis is used. Although it is still true that, sometimes one sentence might have several sentiments simultaneously, in most cases it is not the case. This level of Sentiment Analysis, firstly, determines whether the sentence is subjective or objective and, later, determines the sentiment value of the subjective sentences. Sentence Level Sentiment Analysis is sometimes considered as a short form of

Document Level Sentiment Analysis as they share more similarities. Note that Sentence Level Sentiment Analysis is also applicable to the Emotion Analysis.

3.1.3. Word Level Sentiment Analysis

Sentiment Analysis at Word Level analyzes given text by its words or. This level of sentiment analysis is also called Phrase Level Sentiment Analysis. At this stage, the analysis is done for each individual word in a given text. Sometimes Word Level Sentiment Analysis is not considered one of the main levels of sentiment analysis, as it is rarely used. Note that Word Level Sentiment Analysis is also applicable to the Emotion Analysis.

3.1.4. Feature Level Sentiment Analysis

Sentiment Analysis at Feature Level analyzes features of every word in a given text. This level of sentiment analysis is also called Aspect Level Sentiment Analysis. Here the main idea is to look through the possible sentiment values of each word from multiple perspectives by checking its value in different contexts and detecting its overall value. Although the overall value is not the correct answer always for some contexts, it might benefit the result to know the average attitude towards every element in the text. In short, Feature Level Sentiment Analysis categorizes the sentiment values in accordance with certain entity features. This level of Sentiment Analysis, firstly, determines the entities in the text and, later, determines the aspects of those entities in order to have the sentiment results. Note that Feature Level Sentiment Analysis is also applicable to the Emotion Analysis.

3.2. Classification for Emotion Analysis

Emotion Analysis covers a wide range of topics. Unlike sentiment analysis, here the results can be classified in many different ways. In this section of this paper, we will discuss the most popular three approaches for the classification of Emotion Analysis as shown in the following:

- Ekman's Theory about Basic Emotions
- Plutchik's Wheel of Emotions
- Russel's circumplex Model

3.2.1. Ekman's Theory about Basic Emotions

This type of emotion classification is first proposed in the 1960s by Paul Ekman. Ekman's theory about basic emotions is based on observations of facial behavior and it states that emotions are not continuous, rather they are discrete. It claimed that although some emotions are culture-based, some basic emotions are internationally clear to everyone, hence emotions can be categorized discretely. Based on his experiment, there are 6 categories of basic emotions. These are happiness, sadness, fear, disgust, anger, and surprise. After some years of research, he enlarged his categories of basic emotions to 10 by adding the following 4 emotions, namely, pride, shame, embarrassment, and excitement. Thus, these mentioned categories might be used as classification elements in order to do an Emotion Analysis of a text.

3.2.2. Plutchik's Wheel of Emotion

Aside from Ekman's Theory about basic emotions, another powerful model has been proposed by Robert Plutchik after around two decades. Here not only basic emotions are considered, but also each of these basic emotions is categorized into three sub-levels. These sub-level emotions are created from the amalgamation of those basic emotions. One of the interesting facts about Plutchik's Wheel of Emotion is that Plutchik has made his research on animals first, but they are currently more appliable to humans.

These are the 8 basic emotions at the primary level according to Plutchik's Wheel of Emotions: anger, joy, fear, sadness, disgust, trust, surprise, and anticipation. The emotions at the secondary level are a milder form of the first level: annoyance, serenity, apprehension, pensiveness, boredom, acceptance, distraction, and interest. The tertiary level emotions are the ones that occur the least. These are rage, ecstasy, terror, grief, loathing, admiration, amazement, and vigilance. The following is the final emotion classification by Plutchik that is created by combining the two emotions from lower levels: aggressiveness, love, awe, disapproval, remorse, submission, contempt, and optimism.

All of these emotions at different levels are combined as one into Plutchik's Wheel of Emotions in a very structured way. This wheel is shown in Figure 1. In this wheel, the same-colored emotions are meant to have similar meanings, also the brightness of the color of the emotion shows its strength. Moreover, their position is placed in a way that more similar ones in terms of meaning are closer to each other. This means the emotions that have opposite meanings are placed face to face in the wheel such as joy versus sadness, anticipation versus surprise trust versus disgust, and so on. Additionally, their position is also placed in a way that the more common the emotion is the closer it is placed to the center.

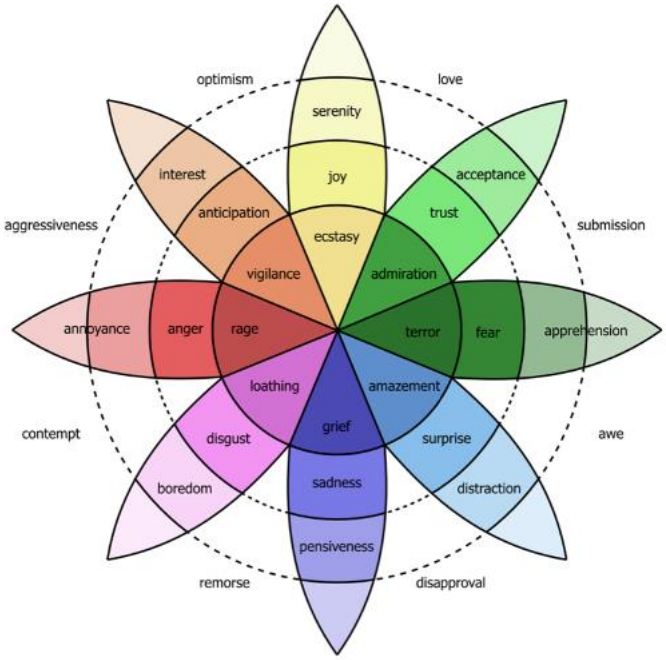

Figure 1. Plutchik's Wheel of Emotions (https://www.6seconds.org/2022/03/13/plutchik-wheel-emotions/)

Plutchik's Wheel of Emotions is widely used, especially for the emotion analysis of a text. Thus, it can be considered one of the best emotion categorizations that have been proposed.

3.2.3. Russel's Circumplex Model

Another noticeable emotion classification has been proposed by James Russel. This model is called a circumplex model because in this model the emotions are shown inside of the circle where the circle is divided into four parts. Russel's Circumplex Model is based on one's subjective feelings and is shown in Figure 2. The main difference in this model is that there are two main dimensions in the circle called valance, which is placed horizontally, and arousal, which is placed vertically and there are numerous emotions in each periphery based on these lines. In the model, the similar-meaning emotions are placed in the same periphery of the circle. As shown in Figure 2, there are 4 main basic emotion categories where two of these categories are negative emotions while the other two are positive emotions which are anger, joy, depression, satisfaction.

Russel's Circumplex Model is used widely in psychological studies, and it might be not well suited for emotion analysis in the text. Because of the way the model was designed, the classification of emotions is continuous instead of discrete. This makes it hard to choose exactly which emotions to consider for the emotion analysis.

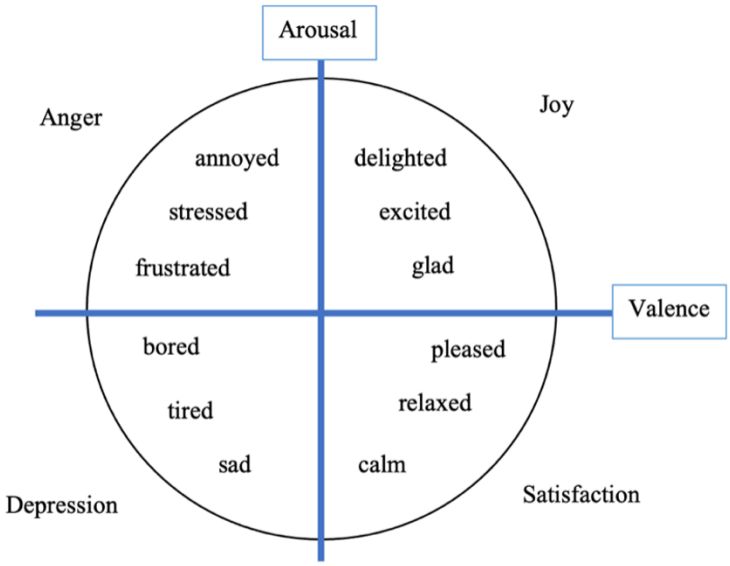

Figure 2. Russel's Circumplex Model

After analyzing these three categories, the primary level emotion categorization of Plutchik's Wheel of Emotion seems to be suited the best for an emotion analysis of a text.

3.3. Data Selection and Preprocessing

Data plays the main role in Machine Learning, hence in Sentiment and Emotion Analysis. Machine Learning plays an extremely critical role in the way today's businesses operate. Although it is used in many areas from finance to social media, the procedures required to train and deploy a model vary for each of them. There are two main forms of Machine learning, namely Supervised Learning, and Unsupervised Learning. Their main difference arises in the training part. Based on the task, one of these machine learning types outweigh the other.

Unsupervised learning uses unlabeled data. The outcome of this type of machine learning requires further explanation to make the results clearer. Indeed, the main distinguishedness between supervised and unsupervised learning is data labeling which belongs to the way the data is trained.

Supervised Learning works with labeled data. Another way to call this type of learning is Supervised Machine Learning. Although it seems tedious work to manually label lines and lines of big data for the machine to learn, this is how it works. There are already many labeled data ready for machine learning, but also there are millions of unlabeled and unstructured data. Also, if a new dataset is being used for machine learning applications, then this dataset must be labeled properly in a manual way. As there's human work involved at some phase of this type of learning, namely in the training phase, this learning is named supervised learning. Generally, supervised learning is a great way to predict some outcome from the input, thus making it suitable for Sentiment and Emotion Analysis. Labeled data is used in supervised learning for the purpose of training models so that they can be used in prediction. In general, supervised learning is a great tool for classification problems and regression problems. Classification problems include classifying test data into some categories

based on the prediction learned by train data. Sentiment and Emotion Analysis are a part of the classification problems of supervised machine learning. Available algorithms and methods may vary here such as neural networks, decision trees, support vector machines, etc. Besides this, there are regression problems that are used for the exploration of relationships among variables. Regression algorithms can be linear, logistic, polynomial, and so on.

Supervised learning splits a labeled dataset into two main sub-datasets, which are the training dataset and the test dataset, sometimes also the validation dataset. As mentioned, the training dataset plays the role of fitting the model. The outcome of this model alters based on the problem. For instance, the prediction outcome of Sentiment Analysis can be positive, negative, or neutral, while Emotion Analysis can give the output of the emotions such as anger, happiness, sadness, and so on depending on which emotions are used in the training process of the model. Meanwhile, the test dataset and the validation dataset are used for measuring the performance and accuracy of the model that is trained by this training dataset. In general, 80% of the dataset is utilized for training, and the remaining 20% is utilized for testing purposes. This splitting way may vary based on the problem, but mostly either 80% or 90% of the data is being trained and 20% or 10% of the data is being tested respectively.

There are three main steps in Supervised Learning programs, hence in Sentiment and Emotion Analysis. The first step is the process of feeding input text data into a previously trained model. After this process, training data is tagged with a predicted output. Here trained model accepts the data as vectors of text and learns the desired labels based on previous human tags that were labeled manually. The last step is also called feeding, but of the data that is not trained. This data has been split beforehand as the test dataset, and it is used in this last stage to see how accurate the model works. Note that training models may refine themselves by being trained over and over because they can explore more of the data in this way. In order to make the trained model predict better, the quality of trained data should be preserved from the earliest stages. The better the trained data is labeled, the better outcome the model can predict. Here better-trained data is a term for more relevant, representative, large enough, and discrete trained data. If the trained data and its labels are not all relevant, there is no meaning to hope that the model will predict properly. Also, if the amount of labeled data is not enough for the model to learn, it can result in undesired outcomes. Moreover, the features of training and test datasets must be the same as the analysis with test data is done on the training data. The quality of the labeling of the training model is in hands of the team of people who labels them manually. So, there should be some kind of agreement and communication between the members of these teams so that they can see similar concepts from the same point of view and not label them differently. Although some tools help to ease the process of manual labeling, this does not apply to Azerbaijani text datasets, hence not applicable to our research.

In Supervised learning, data selection is the very initial thing to do. Depending on the project, the necessary dataset should be decided, found, and then labeled if not already. Especially in Sentiment and Emotion Analysis, the quantity of the data plays an important role as the role of quality of the data. Therefore, the amount of labeled data should be increased as much as possible with quality in mind. As for the data, news articles or written feedback on a certain product/service can be a desired dataset for Sentiment and Emotion Analysis. Note that, in our research, we have

used news articles where we labelled manually over 23.000 news in Azerbaijani. After the selection of data, other tasks arise which are the tasks of Data Preprocessing. These include

- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation

Data Cleaning is a common process when working with big data. Data Cleaning makes the dataset more readable by removing unnecessary characters, correcting missing values, and more. It is almost impossible to find big data that does not require cleaning. Mostly big datasets have some duplicated, corrupted, or improperly formatted lines which influences the quality of training negatively. As the importance and amount of necessary cleaning are purely based on the dataset itself, there is no single one-size-fits-all approach here. However, it is essential to create a structure for the data cleaning procedure to be sure that it is being done in the same desired way each time. Although the steps of data cleaning may vary, the following ones can be considered the common ones that are applied mostly:

- Removing duplicate and unnecessary data
- Fixing typos, syntax errors, and unnecessary uppercase words
- Updating missing data (this is not an efficient method if the dataset is too big)
- Removing stopwords
- Removing unnecessary tags such as HTML tags, XML tags
- Double-checking and validation of data

Data Integration is also a general process in every company that deals with big data today. It is the act of integrating data from several sources in one place in order to be able to view the whole data in a unified way. This process makes the data more accessible, more readable, and more convenient to work on it. Data Integration can benefit a company in many ways including efficiency due to removing manual merging of datasets, quality due to automation, good management due to quality and security, and so on. Although, the process of Data Integration is also vary based on the datasets and the needs of the company, but the general process may include:

- Exploring and collecting data from different sources
- Storing data in a convenient way
- Integrating data with analytics. For instance, if merged datasets include date format, they can be different in the format as in month-day-year and day-month-year. Such cases should be fixed.
- Identifying entities from multiple datasets to understand the relation between them

Data Reduction is needed in order to ease the analysis process. By using data reduction, almost the exact same result can be received in a shorter time, because Data Reduction reduces as much as possible unnecessary data. Data Reduction is helpful in terms of storage. Data Reduction has several methods such as dimensionality reduction, numerosity reduction, and data compression. Dimensionality reduction and numerosity reduction lose no data as they reduce the volume. However, data compression can be both lossy and lossless depending on how it is being used.

Data Transformation is about transforming the data connivance reasons. The noise from the datasets can be decreased or completely removed by the smoothing method of data transformation. Visualization can also be improved by using the normalization method of data transformation. For instance, data can be scaled down to a certain number so that it is more representable. Discretization is another method of data transformation that can play a beneficial role. In case the data size is too large to handle, some unnecessary data can be transformed into another form. For example, discretization can be considered if a one-column-clock-range is used instead of two columns of start time and end time. Aggregation may benefit the quality and quantity of the data which will affect the accuracy directly. Data Normalization is a part of Data Transformation, and it helps to normalize data. Here the main methods are lemmatization and stemming. Both of these methods share a lot of similarities in purpose and most prefer lemmatization over stemming. Stemming finds, as the name suggests, the stem of the word. It is working by applying some set of rules to a word until it reaches the stem. Meanwhile, lemmatization works based on the lemma of a word which is the form of the word exactly as it is in the dictionary.

3.4. Vectorization Methods

Sentiment and Emotion Analysis can be processed in various methods. In all of these methods, the raw text should be converted to a form that the machine can work with it. This process is done by vectorization methods. Machine Learning models cannot understand the input text data without vectorization because machines only understand numbers. Thus, by turning raw text data into a format that Machine Learning supports, it can understand the data and can work on it. Obviously, here the word understand is not the same as what we know as human understanding, but it is termed like this as what it does looks like an understanding if we don't think too deeply.

Vectorization has been around since the dawn of computing, and now is being applied majorly in Natural Language Processing. It can be considered one of the phases in feature extraction.

Several vectorization methods are widely used such as CountVectorizer, TF-IDF Vectorizer, HashingVectorizer, Word2Vec, GloVe, etc.

3.4.1. CountVectorizer

CountVectorizer may be considered one of the most basic approaches available. It is a method that converts a particular set of strings into its frequency form. There are three main steps in this method of vectorization:

- Tokenization
- Vocabulary Definition
- Vector Definition

Tokenization, the very first step of CountVectorizer, is the process of tokenizing the input text into sentences and words. After words are tokenized, the vocabulary is created. At this stage, the unique terms among tokenized words are chosen. In this way, all of the words that are used in the text are collected in one place and then ordered alphabetically. Sorted unique terms are accepted as the created vocabulary. In the final stage of CountVectorizer, the vector is created. This process is being done based on the frequency of the terms in the just created vocabulary.

Vector is a sparse matrix developed using the vocabulary. In this matrix, rows stand for the sentences in the input text. Therefore, the number of columns is as much as the size of the vocabulary. Words in the sentences are appended to the rows of the sparse matrix, called vectors, one by one, which leads to having the words in vectors in the same order as in the sentence. As CountVectorizer simply counts all the words in the input text and extracts features based on it, it is called CountVectorizer. Note that, this whole process is also called the Bag-of-Words model. Bag-of-Words model is considered one of the most popular and efficient methods in order to predict some outcomes. As CountVectorizer works based on counts, it does not have a special formula. One word in the dataset can have many sentiment and emotion values assigned based on the contexts, and CountVectorizer considers them all depending on the frequency of the word while counting. Besides, there are some parameters of CountVectorizer that can be applied in the process of implementation such as lowercase, stopwords, maximum features, and so on.

CountVectorizer can be applied to a program using packages. For instance, in Python, scikit-learn can be used in order to have CountVectorizer. Scikit-learn library is used majorly in the field of Machine Learning ranging from Supervised Learning techniques to Unsupervised Learning techniques. This package is open-source, and it utilizes the following packages; therefore, they need to be imported in order to make scikit-learn available such as SciPy, Matplotlib, and NumPy.

Moreover, sklearn must be imported after the installation of the packages mentioned above. Here SciPy is a Python-based scientific computing package, while NumPy is a Python module for n-dimensional arrays, and Matplotlib is used for a two-dimensional plot and a three-dimensional plot. Downloading these packages makes scikit-learn available for use. Note that, there are already sample datasets that come with the scikit-learn package such as digits and iris. Besides these datasets, several other modules are a part of this library such as clustering, dimensionality reduction, feature selection and extraction, parameter tuning, and so on. The main core of this package is prediction, and it is processed by using the predict() function. Apart from that, in this package, vectorizer has both transform() and fit() functions. CountVectorizer of scikit-learn considers uppercases and punctuations itself, hence making less work on data-preprocessing a little bit.

There are some disadvantages of using CountVectorizer. One of the main disadvantages is that there is no way of determining the most essential word and the least essential word in the text using CountVectorizer because CountVectorizer considers that a word that is used a lot is essential. However, this is always not the case as there are words such as ‘the’, ‘a’, ‘with’ words in English in every language, which are used a lot but cannot be essential words at all. Another drawback of CountVectorizer is the fact that relations between words are ignored here. Thus, CountVectorizer cannot make predictions based on the semblance of the words.

3.4.2. TF-IDF Vectorizer

TF-IDF Vectorizer is another efficient vectorization method used in Natural Language Processing. TF-IDF stands for Term Frequency Inverse Document Frequency. The purpose of the TF-IDF Vectorizer is also to convert a text into a form that the machine can communicate with. The difference between TF-IDF Vectorizer and CountVectorizer is the fact that CountVectorizer extracts the features based on the vocabulary index, whereas TF-IDF Vectorizer evaluates the overall texts

of the weight of words. TF-IDF Vectorizer can provide the importance level of a word with its percentage. Thus, TF-IDF Vectorizer is already a better method than CountVectorizer because it predicts not only based on the word frequency, but also on the fact that how important a word is.

TF-IDF Vectorizer is a measuring way of the originality of a word by comparing the number of times it appears in a given text with the number of given texts the word appears in. We can show this in a formula to understand it better.

$$TF_IDF(t) = TF(t,d) \times IDF(t)$$

Here TF(t, d) is the Term Frequency, and IDF(t) is the Inverse Document Frequency. TF(t, d) means how many times the term t appears in document d. Note that, here the term is the synonym for the word 'word', and the document is the synonym for the word 'given text'. This formula is created so that every word can be counted and also assigned to a value based on its frequency. In more detail, this formula has two main parts, starts with the following term frequency:

$$TF(t,d) = \sum_{x \in d} FR(x,t)$$

where FR(x, t) is a simple function defined as:

$$\mathrm{FR}(x,t) = \begin{cases} 1, & if\ x = t \\ 0, & otherwise \end{cases}$$

The outcome of this formula is the number of times the term t is observed in document d. Using these formulas of term frequency, the sparse matrix is created. After this process, inverse document frequency starts its task. Note that document frequency in this vectorizer is called inverse document frequency because the document frequency is in the log denominator. IDF is being calculated by the following formula:

$$IDF(t) = log\frac{|D|}{1 + |d: t \in d|}$$

Here, D depicts the total number of documents in the corpus, $|\mathrm{d}: \mathrm{t} \in \mathrm{d}|$ demonstrates the number of documents where the term t is detected. Note that, the reason why there is an addition of 1 in the denominator of this formula is due to avoid dividing by zero. As the denominator is the number of terms used in the document, the answer can be none, which would yield a division-by-zero. In the end, the result of the TF-IDF vectorizer is calculated by multiplying the result of term frequency and inverse document frequency. This multiplication formula has a significant effect as it covers both term and inverse document frequencies, therefore it results in a high weight even when we have a lower TF or a lower IDF.

TF-IDF Vectorizer detects the most important words in a text by assigning low tf-idf scores to the words that are hardly ever used and the words that are abundantly used. That's why mostly used nonsense words such as 'the' have a low tf-idf score, hence less importance in the text. TF-IDF Vectorizer can be applied to a program using libraries as well. In Python, scikit-learn can be used in

order to have the TfIdfVectorizer. The fit() function and the transform() function is available in this vectorizer of sklearn. TF-IDF Vectorizer is a great tool to use for Sentiment and Emotion Analysis.

TF-IDF Vectorizer shares some of the disadvantages of CountVectorizer. Both of these methods lack understanding of the meaning of the word and its semblance with other words.

3.4.3. Bag of Words Model

The bag of words model is another concept for processing text data into a form that the machine understands. This model is one of the most common methods for text representation in vectors. Here vector simply means a string of numbers representing the text. The very first thing the Bag of Words model does is to count the number of each unique word in a given text. Based on the number of unique words, each element in the data is depicted by this number of dimensions as a vector. Note that, Bag of Words Models can utilize a list of stopwords in order to not to count undesired words.

Bag of Words Models predicts better results when bigram or higher n-grams are used. n-grams are a list of words where a word is represented with n words next to it. Therefore, when unigrams are used, the words next to a word are not known which yields to predicting not many good results. However, in the case of higher n-grams such as bigrams and trigrams, it can predict based on the context. In n-grams n can represent any positive integer, but it is not advised to use higher integers than three, because trigrams already give desired outcomes.

Though the Bag of Words model is pretty common, there is one main disadvantage to it, which is the vector size. As it is mentioned, the size of the vectors depends on the number of unique words in the text which can be easily long, and as not all texts include exactly the same words, most elements will have zero values in this vector. Having a long list of zeros would yield a sparse matrix which would yield a lack of efficiency in the model.

3.5. Sentiment and Emotion Analysis Methods

Sentiment Analysis and Emotion Analysis can be processed by using various methods. In general terms, these methods can be divided into three main groups:

- Machine Learning approach
- Lexicon-based approach
- Hybrid approach

There are several subclasses inside each of these main groups. Some of this information is provided in Figure 3.

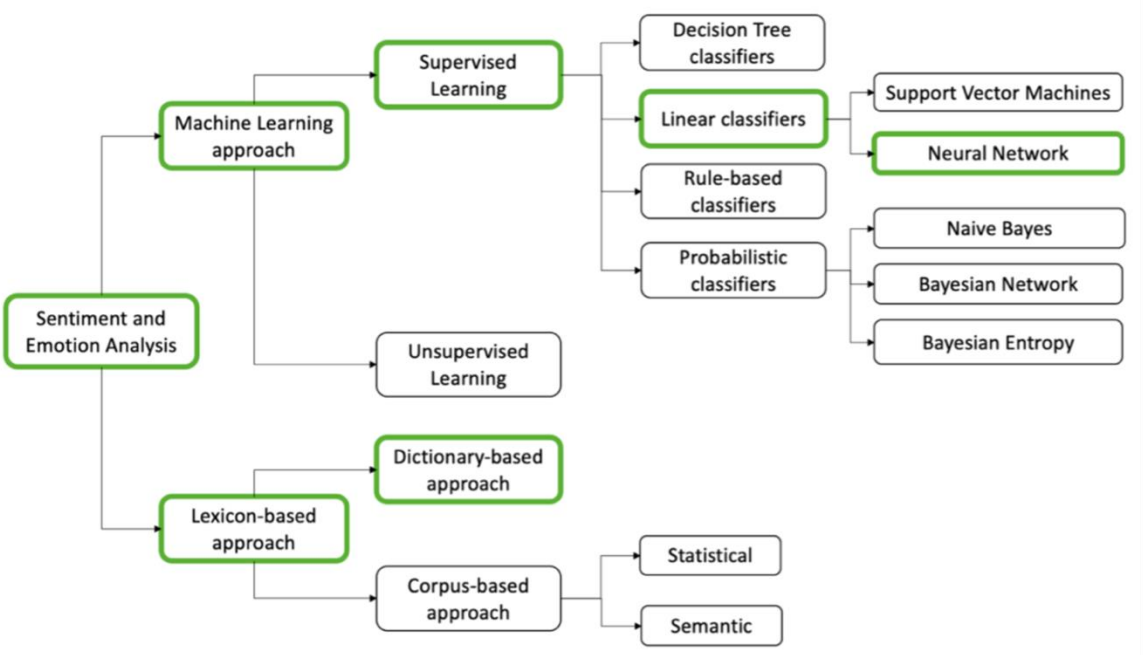

Figure 3. Sentiment Analysis and Emotion Analysis Methods

Machine Learning Approach of Sentiment Analysis and Emotion Analysis can be in one of the following types:

- Unsupervised Machine Learning
- Supervised Machine Learning

Unsupervised Machine Learning works with unlabeled data. Besides these two types, Machine Learning can also be semi-supervised.

Supervised Learning is a more common type of Machine Learning for Sentiment and Emotion Analysis, and it has four main types of classifiers:

- Probabilistic Classifiers
- Decision Tree Classifiers
- Linear Classifiers
- Rule-based Classifiers

Probabilistic Classifiers include several machine learning techniques such as:

- Naïve Bayes
- Bayesian Network
- Maximum Entropy Classifier

Decision Tree Classifiers include Decision Tree as the name suggests.

Linear Classifiers include the following machine learning techniques and others:

- Support Vector Machine (SVM)
- Neural Network

3.5.1. Naïve Bayes

Naïve Bayes is one of the probabilistic classifiers of supervised machine learning that is applicable for Sentiment and Emotion Analysis. This technique is based on one of the probabilistic theorems, namely the Bayes Theorem and it distinguishes for its feature independence.

Bayes Theorem is calculated using the formula below:

$$P(y|X) = \frac{P(X|y) \times P(y)}{P(X)}$$

where P(y | X) denotes the probability of y occurring where the evidence of X has already occurred,

P(X | y) denotes the probability of X occurring where the evidence of y has already occurred,

P(y) denotes the probability of y occurring where y is a stand-in for the theory which are labels and

P(X) denotes the probability of X occurring where X is a stand-in for proof which are features.

The reason why this method is called Naïve Bayes is that the features in this method are independent, therefore the assumptions they gave about the occurrences of features do not influence one another. In simpler terms, Bayes Theorem calculates the posterior by multiplying the likelihood by the division of proposition and its evidence.

X is a set of n features which can be shown as:

$$X = (x_1, x_2, x_3, \dots, x_{n-1}x_n)$$

The formula of Bayes Theorem can also be formed as follows using the equation above:

$$P(y|x_1, \dots, x_n) = \frac{P(x_1|y) \times \dots \times P(x_n|y) \times P(y)}{P(x_1) \times \dots \times P(x_n)}$$

Naïve Bayes uses these formulas after the data is converted into a frequency table alongside a likelihood table.

Naïve Bayes of scikit-learn has several types that can be applied for Sentiment and Emotion Analysis such as:

- Gaussian Naïve Bayes
- Bernoulli Naïve Bayes
- Multinomial Naïve Bayes

Gaussian Naïve Bayes is the first type of Naïve Bayes and its difference from other types is the fact that it follows Gaussian Normal Distribution. In this type of Naïve Bayes, the likelihood of the features is calculated using the formula below:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Bernoulli Naïve Bayes is another type of Naïve Bayes, and it is based on binary values. In this type of Naïve Bayes, all used features of data are converted into binary using Bernoulli Distribution.

Bernoulli Distribution accepts features as 1 if it is a success, and as 0 if it is a failure. Based on this distribution, the formula of Bernoulli Naïve Bayes is formed as follows:

$$P(x_i \mid y) = P(i \mid y)x_i + \big(1 - P(i \mid y)\big)(1 - x_i)$$

Multinomial Naïve Bayes is the most popular type of Naïve Bayes. This type of Naïve Bayes is especially used in document-level sentiment analysis widely.

Naïve Bayes has already been applied to numerous real-world applications related to spam filtering and text categorization including Sentiment Analysis and Emotion Analysis, and it is a quite popular method. The main advantages of using Naïve Bayes are that it is trained faster than other possible methods and uses a lower amount of data for learning. Although these two advantages play an important role in choosing the desired method when looking at the disadvantages, the eagerness to choose this method decreases. Naïve Bayes indeed classifies very well, but it does not predict well. Also, if regression is needed, then this method is not desired as the data cannot contain continuous elements.

Probabilistic classifiers of Supervised Learning also include Bayesian Network and Maximum Entropy Classifiers besides Naïve Bayes. As Naïve Bayes contains only independent features, there was a need for the dependent ones. Bayesian Network is a graph model where the edge of the graph depicts the dependencies which are dependent on some condition. The graph of the Bayesian Network is directed and acyclic and has random values on the nodes. As for the Maximum Entropy Classifier, it deals with features that are labeled with weights. This type of probabilistic classifier is also called the Conditional Exponential Classifier.

3.5.2. Decision Tree

Decision Tree is a method of Decision Tree Classifiers of Supervised Machine Learning. It works not only with discrete data values but also with continuous data values. As the name suggests, the structure of this classifier is tree-based which means it consists of nodes. The predictions are based on these nodes in the tree: it always chooses between two possible choices until it predicts the outcome at the end of the tree.

Decision Trees are popular and good models, but not the best model for Sentiment and Emotion analysis based on their advantages and disadvantages. The advantages of the Decision Tree model include the following:

- It has a good visualization as every step is depicted on a tree structure.
- It is a properly working model for both the data with discrete values and the data with continuous values.
- It does not require data normalization such as lemmatization and stemming.
- It predicts well even when there are flaws in the generated data.

Alongside these advantages, some disadvantages arise when Decision Trees are used. These disadvantages may include the ones below:

- It can cause overfitting if the structure of the tree is too complex.
- It can build completely undesired trees with a minimum change.
- It can be not easy to be understood.

- It can build a non-optimal decision tree.
- It can be expensive if the dataset is complex.
- It cannot predict properly if the dataset is too small compared to other possible methods.

The structure of the process of Decision Trees is shown in Figure 5 on a sample iris data of the scikit-learn library of Python:

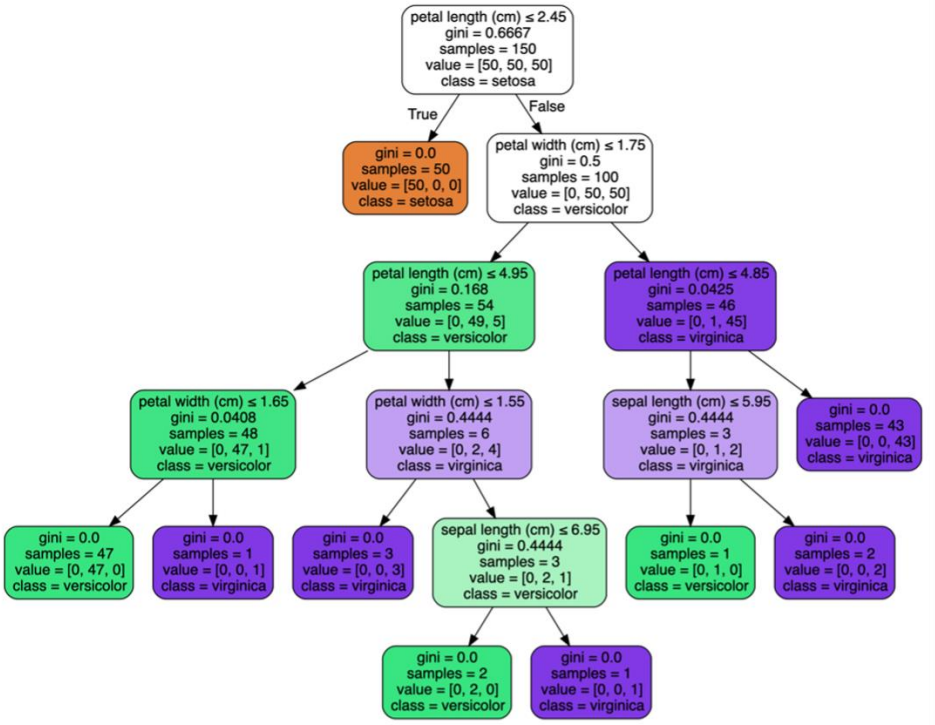

Figure 4. Decision Tree Trained on All Iris Sample Data Features (https://scikit-learn.org/stable/modules/tree.html)

Decision Tree has no parameters and makes predictions based on simple choices. Decision Tree makes this tree of binary choices using entropy. Entropy helps to determine the point where we can make a split efficiently. The entropy of a decision tree is denoted as H(D) and calculated as the following formula:

$$\mathrm{H}(D) = -\sum_{i=1}^{k} P(c_i|D)\, log_2 P(c_i|D)$$

where c stands for the label,
H stands for entropy,
P stands for probability,
D stands for data.

If the entropy is high, it is most likely there are many labels in the data. Note that, entropy can also be trivial when there only one class exists. The splits are done using this entropy, but there are two ways towards improving the way the data splits onto a decision tree. These are:

- Information Gain
- Gini Index

Information Gain is considered the down in the entropy after the split. This information exists in every feature of the data and is calculated by:

$$IG = Entropy\_of\_target - \big((Weighted\_avg \times Entropy\_of\_feature)\big)$$

Information Gain helps to determine the most efficient point to split. More decrease in the entropy yields a better tree split. As it is seen from the formula, Information Gain can easily be detected by finding the entropy of the desired text alongside the entropy of every feature inside it.

Gain Index, on the other hand, uses a technique called CART for measuring the entropy in order to have a better split in the tree. CART is an abbreviation for Classification and Regression Tree. Gain Index calculates this variable using the following formula:

$$CART(D_Y, D_N) = 2 \times \frac{n_Y}{n} \times \frac{n_N}{n} \times \sum_{i=1}^{k} |P(c_i|D_Y) - P(c_i|D_N)|$$

In simpler terms, the Gain Index, a.k.a. Gain Impurity Index, is calculated by finding substitution of the probability of first and second classes' squares which is also substituted from 1.

$$GI = 1 - (Prob_c lass_1 \times Prob_c lass_1) - (Prob_c lass_2 \times Prob_c lass_2)$$

3.5.3. Support Vector Machine (SVM)

SVM, short for Support Vector Machine, is a method of Linear Classifiers of Supervised Machine Learning. Support Vector Machine is a well-known algorithm for especially Sentiment Analysis and Emotion Analysis. In general, it is used for prediction purposes that include regression or classification. There are some terms that Support Vector Machine uses such as:

- Point
- Line
- Plane
- Hyperplane

These terms are related to the number of dimensions that are equal to the number of features. Point is 1-dimensional, Line is 2-dimensional, Plane is 3-dimensional, and Hyperplane is 4-and-more-dimensional. The dimension is calculated using the formula below:

$$w \times x + b = 0$$

This exact formula is used in Support Vector Machine as follows:

$$h(x_i) = \begin{cases} +1 & if\ w \times x_i + b \geq 0 \\ -1 & if\ w \times x_i + b < 0 \end{cases}$$

Another way of representing this formula would be:

$$h(x_i) = sign(w \times x_i + b)$$

The main idea behind Support Vector Machine is to split the data into some classes by either drawing a line or drawing a hyperplane in the dataset. This idea is represented in Figure 5 below:

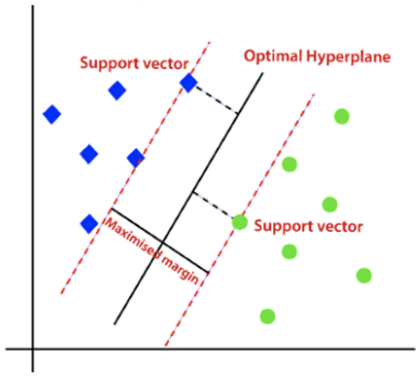

Figure 5. Classified dataset by SVM (https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm)

The data as in Figure 6, can be easily separated into two classes, whereas detecting the optimal points to draw a line in between is not something easy. This process is done by using a Support Vector Machine. It uses the following formula in order to choose between possible hyperplanes that might be more than one, hence it is calculated with the following formula:

$$F = min_{i=1,\dots,m} (y_i \times (w \times x_i + b))$$

The formula above is called a Hyperplane Equation or a Functional Margin Equation. In this equation, the position and place of the hyperplane do not get updated if b in the equation changes. In other terms, Support Vector Machine neglects the scale-invariant b. Therefore, it is necessary to carry along the gained w every time, hence making the formula:

$$M = min_{i=1,\dots,m} (y_i \times (\frac{w}{||w||} \times x + \frac{b}{||w||}))$$

This updated version of this formula is referred to as Geometric Margin. The most optimal hyperplane is chosen based on the comparison of their Geometric Margin.

Generally, a Support Vector Machine has four main tuning parameters. These are:

- Margin
- Kernel
- Regularization
- Gamma

Support Vector Machine contains many beneficial sides in order to be chosen for a classification or regression problem:

- It predicts well if the margin allows easy separation of the data
- It predicts well if there are numerous features, especially if the number of features is more than the number of samples.
- It utilizes less memory.

Support Vector Machine also shares some significant disadvantages:

- It does not predict well if the dataset is too big.
- It does not predict well if the dataset contains a lot of noise.
- It does not perform well if the number of samples is more than the number of features.

3.5.4. Neural Networks

Neural Networks are another method of Linear Classifiers of Supervised Machine Learning. The usage of Neural Networks in the field of Natural Language Processing increases as time goes on. Neural Networks are mainly used for classification and prediction problems. They are also referred to as Artificial Neural Networks, short for ANNs, or Simulated Neural Networks, short for SNNs.

The main characteristic of Neural Networks is that they improve themselves by learning from their past mistakes. Therefore, a good trained Artificial Neural Network can predict good results. Neural Networks contain many neurons inside themselves where each neuron is one unit. These neurons can communicate with their neighbor neurons, and they all work together in order to provide some outcome. Every neuron has some weight associated with them.

Artificial Neural Networks consist of three layers. These layers are:

- Input Layer
- Hidden Layer
- Output Layer

The connection between these layers is depicted in Figure 6 below:

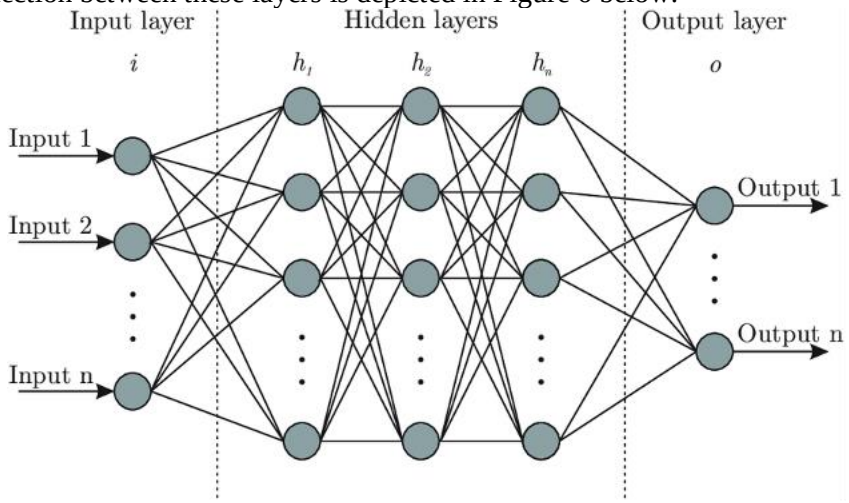

Figure 6. ANN Model (https://www.researchgate.net/figure/Artificial-neural-network-architecture-ANN-i-h-1-h-2-h-n-o_fig1_321259051)

The Input Layer is responsible for supplying the initial data to the neural network. Another way to name this layer is the Input Units.

The Hidden Layer is responsible for building all the connections of the neural network based on the label and the weights of the inputs of the Input Layer. The Hidden Layer does all the necessary processes by itself in the way that they want. Another way to name this layer is the Hidden Units.

The Output Layer is responsible for providing the outcome based on the analysis of the hidden layer. Another way to name this layer is Output Units.

Artificial Neural Networks contain numerous single neurons. A neuron can be considered the very core term of ANNs. Note that another way to name a neuron is a unit or a node. There are neurons in every layer of ANNs. Their working process is shown in Figure 7 below:

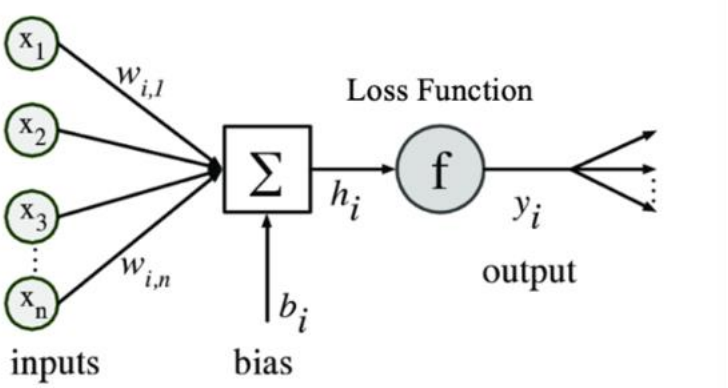

Figure 7. Weights and Bias in ANNs (https://purnasaigudikandula.medium.com/a-beginner-intro-to-neural-networks-543267bda3c8)

They interact with each other when it is needed. For instance, the neurons of the input layer interact with neurons of the hidden layer in order to send the raw data for analysis, and also the neuron of the hidden layer interacts with neurons of the output layer in order to show the predicted result. Neurons of the input layer have a weight associated with them. The weights are assigned based on the essentialness level of the node. These weights added with the biases are applied to a function before it is predicted in the outcome layer. In Figure 7, x denotes a node in the input layer, w denotes the weight assigned to that node, b denotes the bias, f denotes the activation function and y denotes the predicted outcome. There is n number of x inputs in this figure, therefore there are exact n numbers of w weights. Moreover, there is one more line going into the summation function which is dedicated to the bias.

The Summation Function is calculated as:

$$\sum_i w_i x_i + b$$

The Activation Function mostly referred to as f, is against the outcome being linear. Its main goal is to reduce the linearity of the results of the nodes, which is the desired behavior. The Activation Functions work based on certain math formulas. Several possible functions can be used as the activation function. The following are the instances for the possible activation functions:

- Sigmoid
- Softmax
- Hyperbolic Tangent
- Rectified Linear Unit

- Leaky Rectified Linear Unit
- Exponential Linear Unit
- Maxout

Sigmoid converts the values to be between 0 and 1. It works based on the following formula:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

The graph of Sigmoid Activation Function is as follows:

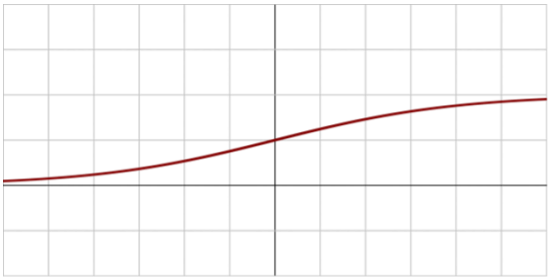

Figure 8. Sigmoid Activation Function (https://en.wikipedia.org/wiki/Activation_function)

Softmax Function converts the values to be a probability between 0 and 1, additionally here the sum of all the values is equal to 1. So, in this activation function, the sum of success probabilities and failure probabilities makes the full percentage. It is calculated using the following formula:

$$\sigma(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$$

Hyperbolic Tangent also referred to as tanh, converts the values to be between -1 and 1. It is calculated using the following formula:

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

The graph of Hyperbolic Tangent Activation Function is as follows:

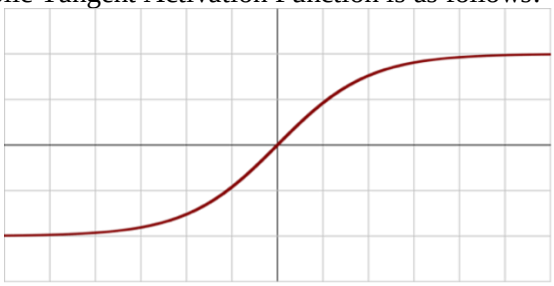

Figure 9. Hyperbolic Tangent Activation Function (https://en.wikipedia.org/wiki/Activation_function)

Rectified Linear Unit, mostly referred to as ReLu, converts all negative values into 0. It is calculated using the following formula:

$$max\{0, x\} = \begin{cases} 0, & if\ x \leq 0 \\ x, & if\ x > 0 \end{cases}$$

The graph Rectified Linear Unit Activation Function is as follows:

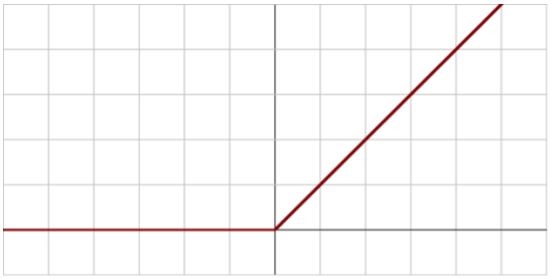

Figure 10. Rectified Linear Unit Activation Function (https://en.wikipedia.org/wiki/Activation_function)

The exponential Linear Unit mostly referred to as ELU, converts all negative values into an exponential value based on the parameter alpha. It is calculated using the following formula:

$$\begin{cases} \alpha(e^x - 1), & if\ x < 0 \\ x, & if\ x \geq 0 \end{cases}$$

The graph Exponential Linear Unit Activation Function is as follows:

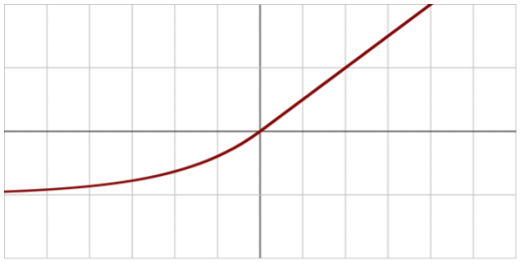

Figure 11. . Exponential Rectified Linear Unit Activation Function (https://en.wikipedia.org/wiki/Activation_function)

There are other several activation functions such as Scaled Exponential Linear Unit, Parametric Rectified Linear Unit, Softplus, Gaussian, and many more.

Artificial Neural Networks consist of two essential steps:

- Feed Forward Propagation
- Backward Propagation

Feed Forward Propagation, also called Forward Propagation is the first main step of neural networks. It works based on the random assignment of weight to every neuron in the neural network. Then, based on these assigned weights, the activation function is calculated. The weights are multiplied by the values and their sum is found. This calculation is applied to all neurons of the hidden layer in a tree-wise way until there are no more neurons in the hidden layer and the result

shows the predicted outcome. The reason why it is called Feed-Forward Propagation is that it passes the data from one layer to the upcoming layer.

Note that, as at the beginning the weights are assigned without any special consideration, the outcome at the end will also be random. Thus, it can easily predict the undesired outcome. Figure 12 below shows the process of Feed-Forward Propagation in Artificial Neural Networks.

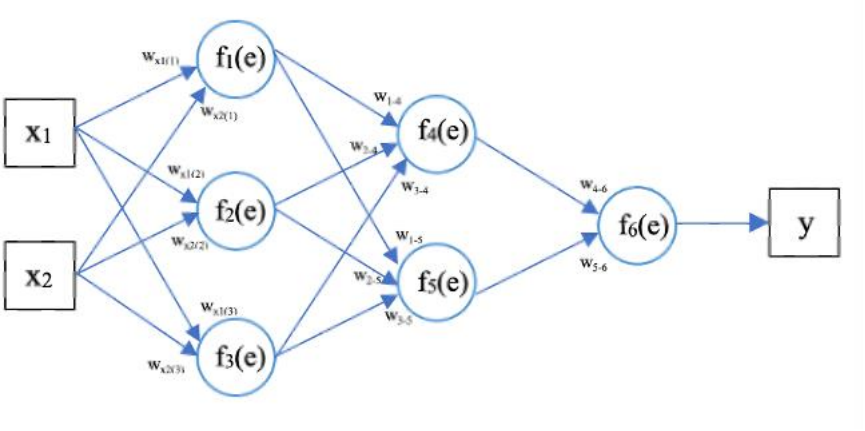

Figure 12. Feed Forward Propagation

Here the weights are calculated one at a time until the end. In the instance of Figure 14, firstly three neurons are calculated as follows:

$$y_1 = f_1(w_{(x1)1} \times x_1 + w_{(x2)1} \times x_2)$$

$$y_2 = f_2(w_{(x1)2} \times x_1 + w_{(x2)2} \times x_2)$$

$$y_3 = f_3(w_{(x1)3} \times x_1 + w_{(x2)3} \times x_2)$$

Afterward, the newly created two neurons are calculated as follows:

$$y_4 = f_4(w_{1-4} \times y_1 + w_{2-4} \times y_2 + w_{3-4} \times y_3)$$

$$y_5 = f_5(w_{1-5} \times y_1 + w_{2-5} \times y_2 + w_{3-5} \times y_3)$$

Lastly, with the combinations of these neurons, y is being predicted.

$$y = f_6(w_{4-6} \times y_4 + w_{5-6} \times y_5)$$

At this step, Feed Forward Propagation completes as the outcome y is already predicted. Now, the error signal is calculated later to apply to Backward Propagation. The error signal is calculated by substituting the desired outcome and the predicted outcome as follows:

$$\delta = z - y$$

where δ stands for the error signal,
z stands for the correct outcome,
and y stands for the predicted outcome.

Backward Propagation, also called Back Propagation, is the second step of neural networks. It cannot be performed first, because what it does is to learn from the mistakes of Feed-Forward Propagation in order to not repeat them. After Feed Forward Propagation predicts an outcome, we as humans can easily detect if the prediction is correct or not. Based on these results, the error signal of the output nodes is calculated and sent back to Backward Propagation for gradient calculations. Gradients of every neuron in the neural network are calculated using the Gradient Descent method. Note that, there are other gradient calculation methods as well. The goal of these methods is to make the model predict better outcomes by decreasing the number of errors they made during Feed Forward Propagation.

The calculated error signal is sent back to each neuron in backpropagation. Figure 13 below shows the process of calculation of error signal after the Feed-Forward Propagation in Artificial Neural Networks.

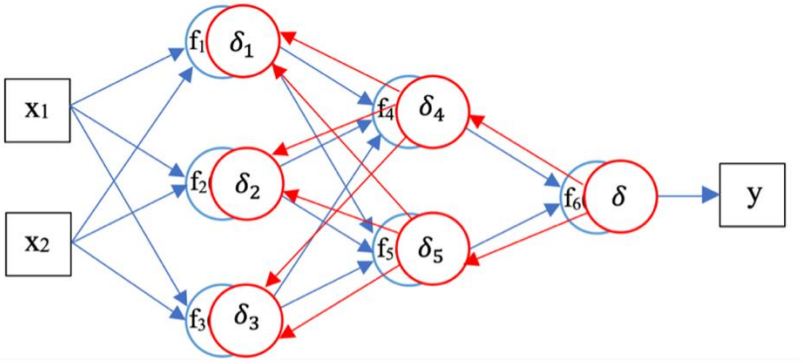

Figure 13. Calculation of Error Signal after Feed-Forward Propagation

The calculation of error signal follows the sequence and the formulas below:

$$\delta = z - y$$

$$\delta_4 = w_{4-6} \times \delta$$

$$\delta_5 = w_{5-6} \times \delta$$

$$\delta_1 = w_{1-4} \times \delta_4 + w_{1-5} \times \delta_5$$

$$\delta_2 = w_{2-4} \times \delta_4 + w_{2-5} \times \delta_5$$

$$\delta_3 = w_{3-4} \times \delta_4 + w_{3-5} \times \delta_5$$

Note that, the weights of this formula through w1-4 to w4-6 are already calculated during the Feed-Forward Propagation and represented in Figure 12.

Backward Propagation starts to apply the results gained from these formulas to every connection of neurons in the network. By this method, the model avoids the wrong results it made during Feed-Forward Propagation as now these weights are combined with the error signals. This combination utilizes the error signal, the previous weights, the previous functions, and a teaching speed parameter. This process is shown in Figure 14 in detail. In this figure, the error signals are represented in blue color to denote that they are already applied to their connections in the neural network.

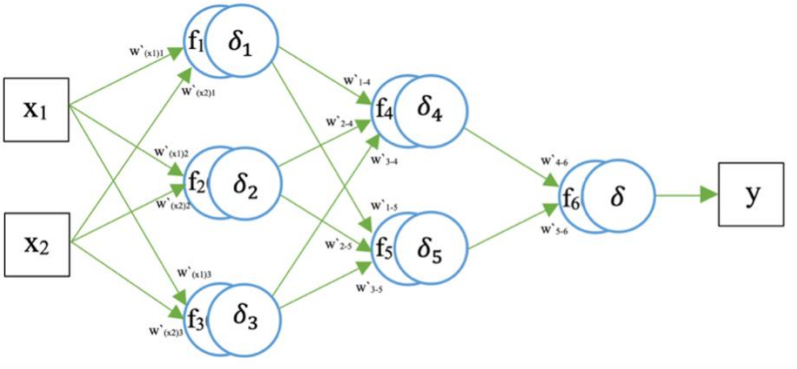

Figure 14. Backward Propagation

The re-calculation of the weights is carried by the following formulas:

$$w'_{(x1)1} = w_{(x1)1} + \eta \times \delta_1 \times \frac{df_1(e)}{de} \times x_1$$

...

$$w'_{(x2)3} = w_{(x2)3} + \eta \times \delta_3 \times \frac{df_3(e)}{de} \times x_2$$

The re-calculation of weights for the last neuron group of the hidden layer is calculated as follows:

$$w'_{1-4} = w_{1-4} + \eta \times \delta_4 \times \frac{df_4(e)}{de} \times y_1$$

...

$$w'_{3-5} = w_{3-5} + \eta \times \delta_5 \times \frac{df_5(e)}{de} \times y_3$$

The re-calculations of weights for the output layer is calculated as follows:

$$w'_{4-6} = w_{4-6} + \eta \times \delta \times \frac{df_6(e)}{de} \times y_4$$

$$w'_{5-6} = w_{5-6} + \eta \times \delta \times \frac{df_6(e)}{de} \times y_5$$

These formulas are shown step by step for every neuron for the smallest possible neural network for clearance. Note that, η in the formulas above, stands for the teaching speed of the neural network. This variable can be assigned either to a high value so that it decreases during the teaching process, or to a low value so that it first increases and then decreases during the teaching process.

After backward propagation, the model will produce better outcomes than it used to. But still, there can be errors remaining due to the poor dataset labeling or other factors. Thus, a well-trained Artificial Neural Network using forward and backward propagation can predict really well outcomes as it corrects the mistakes it made in Feed-Forward Propagation.

Artificial Neural Network has a lot of advantages, especially for Sentiment and Emotion Analysis, compared to other classification techniques:

- It predicts good results with large datasets.
- It stores the data in the whole neural network which is a plus in terms of data loss.
- It works with non-complete data.
- It can work in a parallel environment.
- It learns based on its previous experiences.

Today ANNs are applied in various areas such as:

- Pattern Recognition
- Categorization (inc. Sentiment Analysis and Emotion Analysis)
- Image Identification
- Speech Recognition

Some types of artificial neural networks exist such as:

- Multilayer Perceptron (MLP)
- Convolutional Neural Networks (CNNs)
- Recurring Neural Networks (RNNs)

Multilayer Perceptron is a class of feedforward artificial neural networks. MLPs are the most commonly used neural networks and they are widely applied in many areas. Note that, sklearn library in Python includes Multilayer Perceptron through MLPClassifier.

3.5.5. BERT Model

BERT Model is one of the latest methods of Natural Language Processing. The word BERT stands for Bidirectional Encoder Representation for Transformers. It is an open-source method that does really good predictions depending on the context. What makes BERT so much stand out is the fact that it uses pre-training. Most of the techniques perform poorly because of inadequate data training. Models tend to perform better when they are trained with so much data, so having pre-trained data is a plus for the accuracy and performance of the model. BERT is applicable to many areas of Natural Language Processing including Sentiment Analysis and Emotion Analysis.

BERT uses bidirectionality in training as its name suggests different than other methods which are generally based on sequentially trained data. Bidirectionality gives a broader chance in terms of training the dataset better. This is done using the method called MLM, short-form Masked LM.

Masked ML checks not only the next word in order to learn, but also it considers the previous word. In this technique, the previous word, and the next word are considered simultaneously. Note that, in other methods that use trains sequentially, check only the next word for training and they have direction.

There are two possible versions of pre-training models:

- Context-free pre-training
- Context-based pre-training

Context-free pre-training neglects the context in the text, which means it only creates one word embedding. However, almost in every language, there are some words that are represented the same, but have different meanings in different contexts. Therefore, the context-free pre-training model is not as good as context-based pre-training.

Context-based pre-training uses either one-side direction or two-sides direction in order to cover the context alongside the word. Figure 15 below demonstrates how the BERT model works in general:

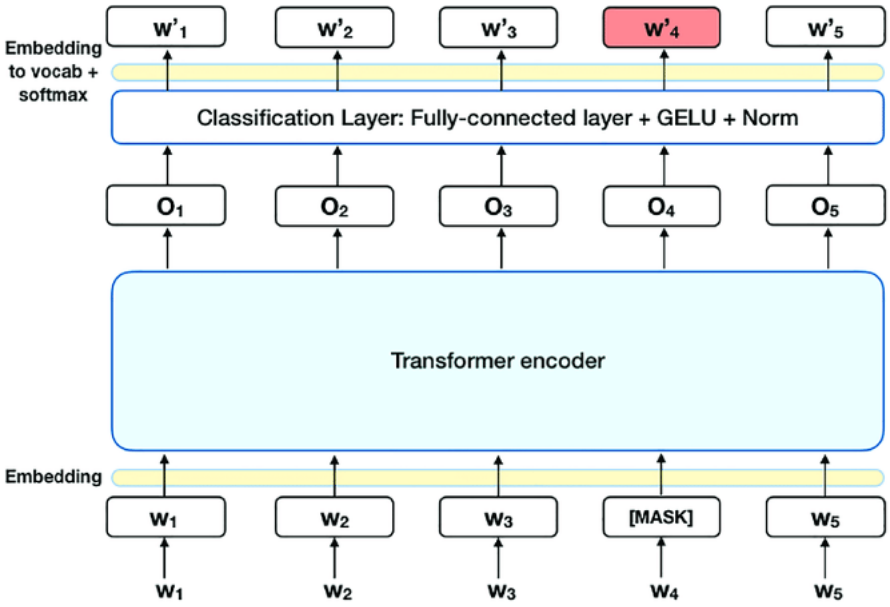

Figure 15. BERT Model

Transformers are at the core of the BERT Model. Transformers have two main parts: encoder and decoder. BERT Model specifically uses the encoder part of Transformers. The encoder is where the input text is read. BERT Model stores the tokens in the encoder of the transformers. There are 3 main processes that happen in this encoder as the following:

- Embedding of Tokens
- Embedding of Segment
- Embedding of Position

After some input text is provided, then every word in every sentence of this input text goes through these mentioned embeddings. Initially, the embedding process is done for the CLS, then it is followed by every word and at the end of every sentence, it is done for the SEP.

BERT Model utilizes the following two methods besides transformers:

- Masked LM
- Next Sentence Prediction

Masked LM utilizes masking little percentage of the input. Note that the selection of masked words is processed in a random way. After the encoder part, the masked words are predicted as there are more unmasked words for training. Note that, mostly 85% of the tokens remain unmasked in the data in order to Masked LM to perform well.

Next Sentence Prediction, short for NSP, is another core method that is used by the BERT Model. Masked LM plays an important role for the model to learn the relations between the words in a sentence, while Next Sentence Prediction is applied for learning the relations between the sentences in the input text. As mentioned before, BERT Model does embed on the SEP after every sentence in a text. By learning the relations between sentences, BERT Model can predict the upcoming sentences.

3.6. Performance Measures

Performance Measures are an important step in order to acknowledge how well the model works. Performance Measures are done using the Confusing Matrices which have four types:

- True Positive (TP)
- False Positive (FP)
- True Negative (TN)
- False Negative (FN)

True Positive means the model predicts as expected where the expected prediction is a positive result. True Negative means the model predicts as expected where the expected prediction is a negative result. Meanwhile, False Positive means the model does not predict as the expected where the expected prediction is a positive result, and False Negative means the model does not predict as the expected where the expected prediction is a negative result.

Performance is measured based on the following factors:

- Accuracy
- Precision
- Recall
- F1-Score

Accuracy is a percentage number that represents how accurate the model works. The calculation of accuracy is pretty straightforward. It is calculated by dividing the number of true predictions by the number of all predictions as follows:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

The sum of True Positive values and True Negative values gives the value of correctly predicted results. Hence dividing this number by the sum of all predictions gives the accuracy.

Precision is a percentage number that is based on positive result values. It is calculated similarly to accuracy, but this performance measurement method neglects True Negative values and False-negative values. The main idea behind precision is to detect the percentage of correctly predicted positive results. It is calculated as:

$$precision = \frac{TP}{TP + FP}$$

Recall is another performance measurement method that is also based on positive result values. The main idea behind the recall is to identify the percentage of positive values among predictions. It is calculated as follows:

$$recall = \frac{TP}{TP + FN}$$

F1-Score is the last performance measurement method which is a combination of Precision and Recall. The difference between precision and recall lies in the fact that they work with False Positive values, and with False Negative values, respectively. That's why it makes more sense to calculate precision, recall, and F-1 Score when the value of False Positive and False Negative differs noticeably. F1-Score is calculated by using the following formula.

$$F1\ score = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}$$

In other terms:

$$F1\ score = 2 \times \frac{precision \times recall}{precision + recal}$$

3.7. Entity Recognition

Entity Recognition, also called Named Entity Recognition (NER), covers one of the sections of Natural Language Processing. The purpose of entity recognition is to recognize and specify certain entities in a given text. An entity itself can be a person, a place, an event, an organization, a date, and many more depending on the context of the text. Note that, another way to name Named Entity Recognition is Entity Extraction and Entity Identification.

Entity Recognition has these three following steps:

- Identifying entities
- Classifying entities
- Double-checking the entities

Identifying the nouns in a text is the first step toward identifying entities because the majority of the time the part of speech for the entities of a text is identified as nouns. In the next step, the entities are classified based on the detected nouns. And, lastly, there might be a need for double-checking so

that the model would be improved if necessary. There are several Python packages for Entity Recognition that works for the English texts such as NLTK, SpaCy, Flair, and so on.

Some methods exist for entity identification of the text. These are:

- Ontology-based Entity Identification
- Deep Learning Entity Identification
- Rule-based Entity Identification

Ontology-based Entity Identification is one of the types of entity identification that uses machine learning. Ontology is a term for the knowledge about the existence and relations of the words in the data. Ontology-based Entity Identification itself can be at various levels. At the simplest level, it recognizes the data with the least or no categorization, whereas at the highest level, it uses every possible category to learn better. Ontology-based Entity Recognition performs well even when the data is not structured data.

Deep Learning Entity Identification is another machine-learning-based entity identification that is better at identification. The presence of this type of entity identification is the fact that it utilizes word embedding. Deep Learning Entity Identification is better at recognizing entities properly than other types of Entity Identification entities as it is trained in different contexts. Thus, the relations between the words are learned by the model in a deeper way.

Rule-based Entity Identification is also a way of identifying entities. Although it is not applied much, it might predict proper results for some languages depending on the grammar of the language. If the grammar of the language is manageable in a way that the entities of a text are identifiable to some extent, it might be used.

4 RESEARCH RESULTS AND ANALYSIS OF RESULTS

This research compares the results for Sentiment Analysis and Emotion Analysis in Azerbaijani Language of various methods, namely Decision Trees, Naïve Bayes, SVM, ANN, and BERT models. This research includes the Bag of Words model and Lexicon-based approach for achieving the results. Furthermore, this research is investigated toward single entity recognition of a text in the Azerbaijani language.

In this research, the Azerbaijani news dataset is used for both Sentiment Analysis and Emotion Analysis of a text. We have increased the amount of news in this dataset to 24,000 news. Note that, in previous studies which used this dataset, this amount was 16,000 news. We have labeled this dataset for both Sentiment Analysis and Emotion Analysis simultaneously. This dataset includes 12 different columns inside. The columns of this dataset start with the column ‘content’ which is the actual news. It is followed by 3 more columns which are in use for Sentiment Analysis: positive, negative, and neutral. The last 8 columns of this dataset are used for Emotion Analysis which are anger, anticipation, disgust, fear, joy, sadness, surprise, and trust. The categorization of these emotions is selected based on the main emotions of Plutchik’s Wheel of Emotions. Every single news in this dataset is labeled manually to later use for training and testing purposes.

This research includes the BOW approach and the Lexicon-based approach. In order to apply lexicon-based approach to Sentiment and Emotion Analysis, we have created from scratch a new dictionary dataset in the Azerbaijani language with corresponding labels. We call this dataset the

Lexicon Dictionary or LD. LD dataset includes 12 columns as our news dataset which has the same label headers. The difference here is the context of the 'content' column. Here the 'content' column includes individual words rather than news and the rest of the columns such as positive, negative, anger, joy, and others include the corresponding label of that individual word. We have labeled 66,000 words properly in this dataset. The content data for the LD dataset is chosen from Azerbaijani Dictionary Dataset. The key idea behind the lexicon-based approach is to replace each individual word in the news dataset with its corresponding label in the Lexicon Dictionary dataset. Therefore, it is better if the Lexicon Dictionary is the individual words representations of the words in the news dataset. In the lexicon-based approach, the words in the news dataset are replaced with their individual labels in the lexicon dictionary, and then the Bag of Words model has applied to this just created dataset.

Table 1 below shows one example line from News Dataset:

Table 1. News Dataset Sample

content	positive	negative	neutral	anger	anticipation	disgust	fear	joy	sadness	surprise	trust
Sabah Bakıda hava şəraiti əsasən yağmursuz keçəcək.	pos				ant			joy			tru

Table 2 below depicts one example from Lexicon Dictionary:

Table 2. Lexicon Dictionary Dataset Sample

content	positive	negative	neutral	anger	anticipation	disgust	fear	joy	sadness	surprise	trust
yağmursuz	Pos				ant			joy			

Note that in both of these datasets, every element may have only and only one sentiment element, which is either positive, negative, or neutral, while they may have multiple emotions at the same time. Another thing to consider about these datasets is the way they are labeled. If a word or a news item does not share a feature, that intersection either will be left empty, or will be written 0. However, if a word or a news item shares a feature, then the first three letters of the name of the features are written into that intersection. For instance, pos for positive, ang for anger, sur for surprise, etc. This labeling can also be replaced with 1 for simplicity, but this is the way it is currently labeled for easier visualization.

In this research, we have examined both CountVectorizer and TF-IDF Vectorizer and concluded that TD-IDF Vectorizer gives slightly better outcomes than CountVectorizer.

As for Entity Recognition, we used a rule-based approach in order to detect the entity in a text based on some grammar rules of Azerbaijani. It detects the objective in the sentence by checking the suffixes of the words, and if it does not find anything, it detects the subject in the sentence.

Let's see the results of this research from the point of each methodology:

4.1. Decision Trees

Decision Trees Classifier with CountVectorizer performed the least successfully among others. Table 3 demonstrates these results in a detailed way:

Table 3. Results of Decision Trees on Sentiment Analysis

	unigram	**bigram**	**trigram**
Precision	pos = 85 neg = 90	pos = 86 neg = 88	pos = 84 neg = 85
Recall	pos = 82 neg = 91	pos = 80 neg = 92	pos = 73 neg = 91
F1-Score	pos = 84 neg = 90	pos = 83 neg = 91	pos = 78 neg = 87
Accuracy	87.09%	88.1%	83.94%
CV Score	0.87	0.86	0.85

4.2. Naïve Bayes

The results of the Naïve Bayes Classifier with CountVectorizer are shown in Table 4 below:

Table 4. Results of Naïve Bayes on Sentiment Analysis

	unigram	**bigram**	**trigram**
Precision	pos = 88 neg = 92	pos = 89 neg = 92	pos = 90 neg = 93
Recall	pos = 90 neg = 92	pos = 89 neg = 90	pos = 89 neg = 91
F1-Score	pos = 90 neg = 92	pos = 89 neg = 91	pos = 89 neg = 90
Accuracy	90.2%	91.11%	90.88%
CV Score	0.89	0.90	0.87

4.3. SVM

The results of the SVM Classifier with CountVectorizer are shown in Table 5 below:

Table 5. Results of SVM on Sentiment Analysis

	unigram	**bigram**	**trigram**

Precision	pos = 92 neg = 93	pos = 91 neg = 91	pos = 88 neg = 92
Recall	pos = 91 neg = 93	pos = 88 neg = 94	pos = 81 neg = 94
F1-Score	pos = 91 neg = 94	pos = 91 neg = 92	pos = 86 neg = 91
Accuracy	93.28%	92.05%	89.7%
CV Score	0.93	0.92	0.89

4.4. ANN

The results of the ANN Classifier with TF-IDF Vectorizer are shown in Table 6 and Table 7 below. Here MLP Classifier of sklearn of Python library is used which is a feed-forward Artificial Neural Network and a lexicon-based approach is applied.

Table 6. Results of ANN on Sentiment Analysis using Lexicon-based Approach

Accuracy: 81.72%	**positive**	**negative**
Precision	91	93
Recall	87	95
F1-Score	89	94

Table 7. Results of ANN on Emotion Analysis

Accuracy: 71.2%	**anger**	**anticipation**	**disgust**	**fear**	**joy**	**sadness**	**surprise**	**trust**
Precision	19	50	2	7	55	59	31	8
Recall	21	44	2	7	57	62	30	8
F1-Score	20	47	2	7	56	61	31	8

4.5. BERT

Table 8. Results of BERT on Sentiment Analysis

	Epoch-1	**Epoch-2**	**Epoch-3**	**Epoch-4**	**Epoch-5**
Accuracy	64	72	77	80	83

5 DISCUSSION AND CONCLUSIONS

This research has studied Sentiment Analysis and Emotion Analysis in Azerbaijani Language. As explained in detail, news articles and lexicon dictionary datasets are used. We have created the

model by using TF-IDF Vectorizer and MLPClassifier. We have developed a Python GUI as follows in Figure 16:

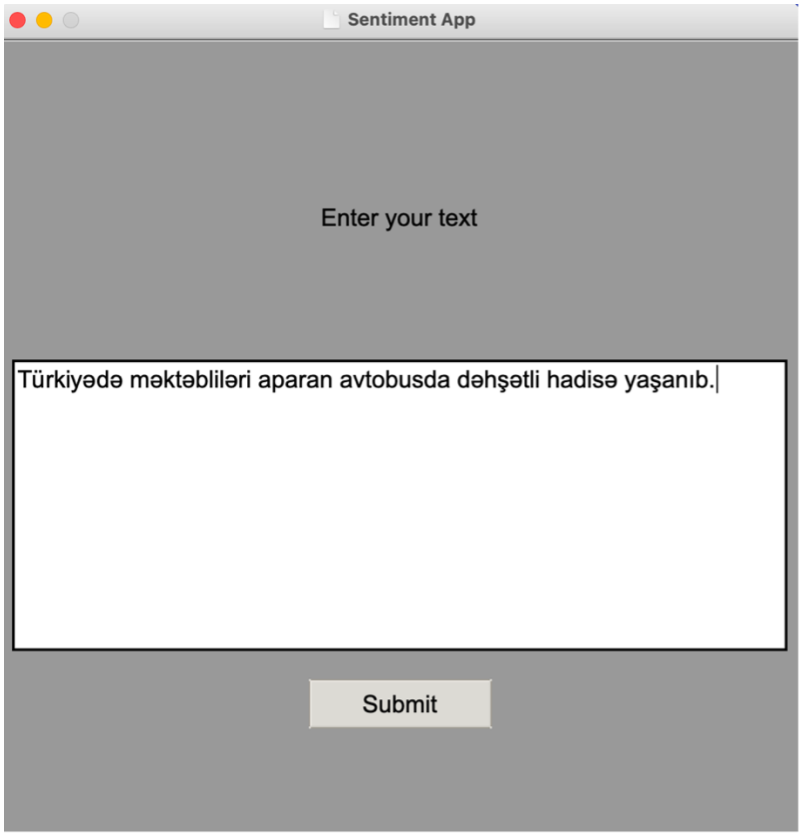

Figure 16. Sample from Program (Before Submit)

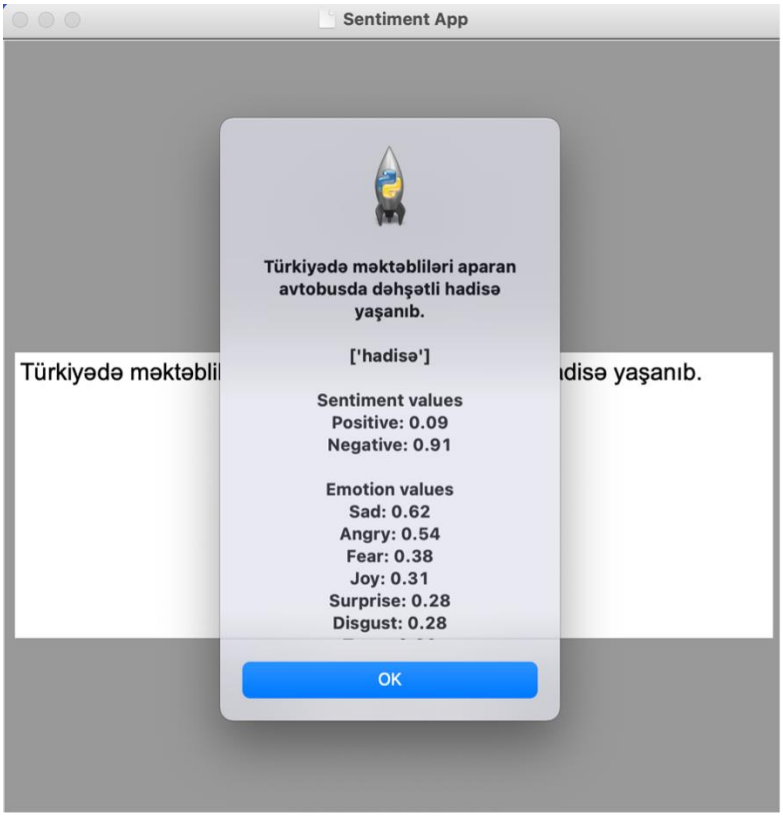

Figure 17. Sample from Program (After Submit)

In conclusion, we have done research on Sentiment Analysis and Emotion Analysis of a text in the Azerbaijani language. We get the accuracy of 82% for Sentiment Analysis and 71% for Emotion Analysis. We have involved lexicon-based approach into Sentiment and Emotion Analysis, its results could be improved by trying other various methods out.

REFERENCES

[1] Chen, Qufei & Sokolova, Marina. (2019). Unsupervised Sentiment Analysis of Objective Texts. 10.1007/978-3-030-18305-9_45.

[2] H. Zhang, Z. Yu, M. Xu, Y. Shi Feature level sentiment analysis for Chinese product reviews Proceedings of the 3rd International Conference on Computer Research and Development (2011), pp. 135-140

[3] Y. He, "Incorporating Sentiment Prior Knowledge for Weakly Supervised Sentiment Analysis," ACM TALIP, 2012.

[4] N. Godbole, M. Srinivasaiah, and S. Skiena. 2007. Large-scale sentiment analysis for news and blogs. In Proceedings of the International Conference on Weblogs and social media (ICWSM).

[5] Alexandra Belahur and Ralf Steinberger. 2009. Rethinking Sentiment Analysis in the news: from Theory to Practice and back. WOMSA'09, pages: 1-12

[6] Bo Pang and Lillian Lee (2008), "Opinion Mining and Sentiment Analysis", Foundations and Trends® in Information Retrieva: Vol. 2: No. 1–2, pp 1-135. http://dxdoi.org/10.1561/1500000011

[7] Kaya M (2013) Sentiment analysis of Turkish political columns with transfer learning. PhD thesis, Middle East Technical University, Ankara

[8] Severyn A, Moschitti A (2015) UNITN: training deep convolutional neural network for Twitter sentiment classification. In: Proceedings of SEMEV AL, Denver, CO, pp 464–469

[9] Azam, Nazish & Tahir, Bilal & Mehmood, Amir. (2020). Sentiment and Emotion Analysis of Text: A Survey on Approaches and Resources.

[10] Soujanya Poria, Alexander Gelbukh, Amir Hussain, Newton Howard, Dipankar Das, and Sivaji Bandyopadhyay. Enhanced sentiment with affective labels for concept-based opinion mining. Journal of IEEE Intelligent Systems, 28(2):31–38, 2013.

[11] Saif M Mohammad and Peter D Turney. Crowdsourcing a word–emotion association lexicon. Journal of Computational Intelligence, 29(3):436–465, 2013.

[12] Kim, Evgeny & Klinger, Roman. (2018). A Survey on Sentiment and Emotion Analysis for Computational Literary Studies.

[13] Carlo Strapparava and Rada Mihalcea. Semeval- 2007 task 14: Affective text. In Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), pages 70–74, 2007.

[14] Patil, Priyanka, and Pratibha Yalagi. "Sentiment analysis levels and techniques: a survey." International Journal of Innovations in Engineering and Technology 6.4 (2016): 523-528.

[15] Behdenna, Salima & Barigou, Fatiha & Belalem, Ghalem. (2018). Document Level Sentiment Analysis: A survey. EAI Endorsed Transactions on Context-aware Systems and Applications. 4. 154339. 10.4108/eai.14-3-2018.154339.

[16] Shirsath, Vishal & Jagdale, Rajkumar & Shende, Kanchan & Deshmukh, Sachin & Kawale, Sunil. (2019). Sentence Level Sentiment Analysis from News Articles and Blogs using Machine Learning Techniques. International Journal of Computer Sciences and Engineering. 7. 1-6. 10.26438/ijcse/v7i5.16.

[17] Jurafsky, Daniel, and J. Martin. "Naive bayes and sentiment classification." Speech and language processing (2017): 74-91.

[18] Medhat, Walaa, Ahmed Hassan, and Hoda Korashy. "Sentiment analysis algorithms and applications: A survey." Ain Shams engineering journal 5.4 (2014): 1093-1113.

[19] F.A. Pozzi, E. Fersini, E. Messina, D. Blanc Enhance polarity classification on social media through sentiment-based feature expansion. Proceedings of the 14th Workshop "From Objects to Agents" — 13th Conference of the Italian Association for Artificial Intelligence (2013), pp. 78-84

[20] D. Maccagnola, F.A. Pozzi, E. Fersini, E. Messina Enhance user-level sentiment analysis on microblogs with approval relations. Proceedings of the 13th Conference of the Italian Association for Artificial Intelligence (2013), pp. 133-144

[21] S. Rustamov, E. Mustafayev and M. A. Clements, "Sentiment analysis using Neuro-Fuzzy and Hidden Markov models of text," 2013 Proceedings of IEEE Southeastcon, 2013, pp. 1-6, doi: 10.1109/SECON.2013.6567382.

[22] Hasanli, Huseyn & Rustamov, Samir. (2019). Sentiment Analysis of Azerbaijani twits Using Logistic Regression, Naive Bayes and SVM. 1-7. 10.1109/AICT47866.2019.8981793.