REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Data Analytics

SENTIMENT ANALYSIS OF TWEETS ABOUT KARABAKH IN TWITTER BY APPLYING MACHINE LEARNING TECHNIQUES

Sanan QIYASZADE

Master's Thesis

Supervisor

Asst. Prof. Dr. Oğuz KARAN

İstanbul, 2024

SENTIMENT ANALYSIS OF TWEETS ABOUT KARABAKH IN TWITTER BY APPLYING MACHINE LEARNING TECHNIQUES

Sanan QIYASZADE

Data Analytics

Master’s Thesis

ALTINBAŞ UNIVERSITY

2024

The thesis titled SENTIMENT ANALYSIS OF TWEETS ABOUT KARABAKH IN TWITTER BY APPLYING MACHINE LEARNING TECHNIQUES prepared by SANAN QIYASZADE and submitted on 28/06/2024 has been **accepted unanimously** for the degree of Master of Science in Data Analytics.

_____

Asst. Prof. Oğuz KARAN

Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Oğuz KARAN	Department of Software Engineering, Altınbaş University	_____
Assoc. Prof. Dr. Sefer KURNAZ	Department of Computer Engineering, Altınbaş University	_____
Asst. Prof. Dr. Serdar KARGIN	Department of Biomedical Engineering, Istanbul Arel University	_____

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

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

Sanan QIYASZADE

Signature

DEDICATION

I would like to dedicate my thesis to my family, whose support, motivation, and sacrifices have been the foundation of my academic journey. Their endless love, understanding, and trust in me have been the only source of strength throughout this effort.

Also, I am truly thankful to my thesis advisor, Asst. Prof. Oğuz KARAN, whose support, and guidance were very valuable.

ABSTRACT

SENTIMENT ANALYSIS OF TWEETS ABOUT KARABAKH IN TWITTER BY APPLYING MACHINE LEARNING TECHNIQUES

QIYASZADE, Sanan

M.Sc., Data Analytics, Altınbaş University,

Supervisor: Asst. Prof. Dr. Oğuz KARAN

Date: May / 2024

Pages: 43

In recent years, social media platforms have become powerful sources of public sentiment, reflecting the diverse spectrum of emotions and opinions on global events. This thesis delves into the sentiments expressed on Twitter regarding the Karabakh conflict, a longstanding and continuous issue. Through advanced sentiment analysis techniques, this study examines a major corpus of tweets related to Karabakh, aiming to evaluate the prevailing sentiments, patterns, and trends within the discourse. Our analysis categorizes tweets into positive, negative, or neutral sentiments, relying on state-of-the-art sentiment analysis methodologies. Utilizing natural language processing and machine learning algorithms, we were able to compare machine learning models and acquire the top recommended ML model. More than 10.000 tweets were collected and analysed. In this research Count Vectorizer and TF-IDF vectorizers have been applied to convert textual data into numerical vectors. At the end, we concluded that Logistic Regression performed well compared to other ML models, which will be given in proceeding steps. The implications of this research extend beyond academic inquiry, offering a nuanced understanding of how social media platforms reflect and influence public opinion during protracted geopolitical conflicts. This study contributes to the fields of sentiment analysis, social media analytics, and conflict studies, by providing information about the relationship between digital discourse and real-world conflicts.

Keywords: Sentiment Analysis, Machine Learning, Natural Language Processing, Social Media, Public Opinion, Karabakh Conflict.

ÖZET

TWITTER'DA KARABAĞ HAKKINDA ATILAN TWEETLERİN MAKİNE ÖĞRENMESİ TEKNİKLERİ UYGULANARAK DUYGU ANALİZİ

QIYASZADE, Sanan

Yüksek Lisans, Veri Analitiği, Altınbaş Üniversitesi

Danışman: Dr. Öğr. Üyesi Oğuz KARAN

Tarih: Mayıs / 2024

Sayfa: 43

Son yıllarda sosyal medya platformları, küresel olaylara ilişkin farklı duygu ve görüş yelpazesini yansıtan güçlü bir kamuoyu duyarlılığı kaynağı haline geldi. Bu tez, uzun süredir devam eden ve bir sorun olan Karabağ konusuna ilişkin Twitter'da ifade edilen duyguları incelemektedir. Bu çalışma, gelişmiş duygu analizi teknikleri aracılığıyla Karabağ ile ilgili önemli bir tweet yapısını inceleyerek söylemdeki hakim duyguları, kalıpları ve eğilimleri değerlendirmeyi amaçlamaktadır. Analizimiz, son teknoloji duygu analizi metodolojilerine dayanarak tweet'leri olumlu, olumsuz veya tarafsız duygulara göre sınıflandırır. Doğal dil işleme ve makine öğrenimi algoritmalarını kullanarak makine öğrenimi modellerini karşılaştırdık ve en çok önerilen makine öğrenimi modelini elde edebildik. 10.000'den fazla tweet toplandı ve analiz edildi. Bu araştırmada metinsel verileri sayısal vektörlere dönüştürmek için Count Vectorizer ve TF-IDF vektörleştiriciler uygulanmıştır. Sonunda Lojistik Regresyonun, ilerleyen adımlarda anlatılacak olan diğer makine öğrenimi modellerine göre iyi performans gösterdiği sonucuna vardık. Bu araştırmanın sonuçları akademik araştırmanın ötesine geçerek sosyal medya platformlarının uzun süren jeopolitik çatışmalar sırasında kamuoyunu nasıl yansıttığı ve etkilediğine dair incelikli bir anlayış sunuyor. Bu çalışma, dijital söylem ile gerçek dünyadaki çatışmalar arasındaki ilişki hakkında bilgi sağlayarak duygu analizi, sosyal medya analitiği ve çatışma çalışmaları alanlarına katkıda bulunmaktadır.

Anahtar Kelimeler: Duygu Analizi, Makine Öğrenmesi, Doğal Dil İşleme, Sosyal Medya, Kamuoyu, Karabağ Çatışması.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

LR	:	Logistic Regression
ML	:	Machine Learning
NB	:	Naïve Bayes
NLTK	:	Natural Language Toolkit
NLP	:	Natural Language Processing
RF	:	Random Forest
SA	:	Sentiment Analysis
SIA	:	Sentiment Intensity Analyzer
SVM	:	Support Vector Machine
TF-IDF	:	Term Frequency and Inverse Document Frequency

1. INTRODUCTION

In the digital age, social media services have emerged as dynamic hubs for the exchange of information, opinions, and sentiments on a global scale. These platforms have become essential channels for individuals to express their views, discuss contemporary issues, and participate in public discourse. The Karabakh conflict, a long-standing and complex geopolitical issue, is no exception to this trend. Twitter, one of the world's largest microblogging platforms, has witnessed an influx of user-generated content discussing and debating the Karabakh conflict.

The Karabakh conflict, centred around the Nagorno-Karabakh region and involving Armenia, Azerbaijan, and other stakeholders, is a long-running and deeply based conflict that has lasted decades. Over the years, it has gathered significant international attention and media coverage. Twitter, as a real-time platform, has become a rich source of data for understanding the sentiments of individuals across the globe in response to developments related to the conflict.

1.1 WHY TWEETS ABOUT KARABAKH CONFLICT?

In the last years there have been published enormous number of tweets about Karabakh conflict in Twitter. During the second Karabakh war, social media, including Twitter, became a battleground for information discussion. Thus, by using the data, we can analyze the conversations related to Karabakh conflict behind the tweets and offer a thorough framework for understanding the diverse ways that Twitter impacts public perceptions of this ongoing geopolitical war. And, by getting such dynamic data related to this topic, we can deeply analyze the conflict and understand unfairness against the Nagorno-Karabakh.

1.2 RESEARCH AIM

This study aims to evaluate the performance of machine learning techniques by applying them into Twitter data of geopolitical tweets, examining the emotional tone and polarization within digital conversations. Through a comprehensive analysis we can obtain sentiments of the data by investigating the tweets. The primary aim is to comprehensively analyze the role of this social media platform’s function as a dynamic arena for exchange of information, opinion formation, and geopolitical debate during the conflict.

1.3 SIGNIFICANCE AND RELEVANCE OF SENTIMENT ANALYSIS IN SOCIAL MEDIA

Sentiment analysis, an essential component of natural language processing (NLP), has become a key tool for understanding the sentiments, and opinions of people as they express themselves on social media. Sentiment Analysis plays a vital function in understanding public opinion and perception on a variety of subjects. By examining sentiments expressed in tweets, researchers can gain insight about the prevailing opinions, feelings, and sentiments on social media.

2. LITERATURE REVIEW

There have been numerous research articles published on big data analysis, sentiment analysis, and opinion mining. Previous studies on sentiment analysis had been analysed for better analysis. For example, J Singh and P Tripathi applied SVM, Random Forest and decision tree approaches to categorize tweets as positive, negative, and neutral. For cleaning the dataset, the authors used TF-IDF algorithm. Among the algorithms decision tree performed best accuracy by 88.51% [1]. H Parveen and S Pandey applied Hadoop framework for movie dataset from twitter by using naïve bayes classifier [2]. M. Faisal et al applied machine learning techniques to predict football fans emotions during Qatar world cup 2022. The dataset utilized in this study was collected from Twitter API including various Arabic countries. They applied four algorithms in this study: logistic regression, random forest, Naive Bayes classifier, and support vector machine. The best result among these algorithms performed SVM by 93% accuracy [3]. Madan, Anjum, and Udayan analysed sentiments of movie reviews in Hindi language from twitter. In this research they applied Natural Language Processing techniques by using Lexicon based approach and Machine Learning Approach based on supervised learning. Also, to validate the results, the Python Programming Language was applied. In conclusion, machine learning based models performed higher results comparing to other used approaches [4]. Shamrat, F. M. J. M., et al. extracted tweets related to COVID-19 vaccines from twitter to analyse peoples' feelings during pandemic. In this project Twitter API authentication token was used to extract tweets. Also, to process and store the raw data they used NLP techniques and to classify the processed data supervised KNN algorithm was applied [5]. Mary, G. Prema Arokia, et al. [6] proposed to find sentiments from twitter data by using machine learning algorithms such as Logistic Regression, Random Forest Classifier, Linear SVC, Bernoulli NB, Voting Classifier, Decision Tree Classifier and KNN Classifier. In conclusion, Linear SVC and Logistic Regression performed high result comparing to other used algorithms. Gandhi, Usha Devi, et al. proposed sentiment analysis of IMDB film reviews by using deep learning models such as Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM). By using these techniques with natural language processing methods, authors classified the data into positive and negative reviews [7]. Maulana, Fairuz Iqbal, et al. proposed a paper about COVID-19 vaccines where they applied machine learning to detect sentiments from the dataset. Furthermore, to extract the data from given dataset authors used

tweepy python package. Also, for sentiment classification TexBlob was used. As a result, positive, neutral, and negative tweets about vaccines were obtained [8]. Hasib, Khan Md, et al. applied deep learning to classify sentiment analysis of twitter data of us airline service. The work was separated into three major parts in order to end up at the suggested model. In the first part the dataset was processed to adapt and filter the given data. The second part was used to future extraction, where for extracting the feature authors used TF-IDF. In the last stage multiple models were tested to classify the data. In this stage, SVM produced the best results compared to other models [9]. A. S. Neogi et al. extracted tweets on farmers' protests in India in order to analyse sentiments by using g TF-IDF approach. The authors used machine learning models such as Naïve Bayes, Decision Tree, Random Forest and Support Vector Machine to classify the data. According to the authors, the Random Forest classifier outperformed the other three classifiers [10]. Singh, Satyendra, Krishan Kumar, and Brajesh Kumar proposed a paper of sentiment analysis of US airlines dataset from twitter. In this article, they applied TF-IDF technique for extracting the feature and also, machine learning techniques were used for classification and analysis. The proposed model is divided into three stages: pre-processing, feature extraction and classification method. At classification stage researchers applied multiple algorithms such as RF, GB, XGBoost and SVM. The proposed model performed best result when combined with the SVM classifier. Researchers stated that future studies include the use of other testing features [11]. Chiorrini et al. provide a paper by using real-world Twitter datasets, where they evaluated the performance of Bidirectional Encoder Representations from Transformers (BERT) models for sentiment analysis and emotion recognition. A proposed model performed 92% of accuracy for sentiment classification and 90% accuracy for emotion recognition where both the cased and uncased variants of BERT were used [12]. Rathod, Dharmendrasinh, et al. proposed a sentiment analysis study of drug reviews by implementing machine learning techniques. This present research compares the performance of various machine learning models for sentiment analysis of drug reviews. At the end, Decision Tree, Random Forest, Naïve Bayes, and Extra Tree classifiers compared to get the highest accuracy ML model [13].

3. METHODOLOGY

In this study, as we said before, we are going to use different machine learning algorithms to extract and analysis sentiments from twitter data. Sentiment Analysis is a use of Natural Language Processing which is used for text classification. Sentiment analysis classifies text into several sentiments, such as positive, negative, and neutral [14].

As previously said, the objective of this research is to utilize sentiment analysis on textual data from Twitter and perform text polarities on the data. The first phase in this stage begins with gathering the data from Twitter. Then a few essential elements, such as cleaning and preparing the data to get it in a format that can be understood by a machine. Visualization steps are also included in the process. Lastly, the data was tested and trained using the Logistic Regression, Random Forest, Support Vector Machine and Naïve Bayes classifiers and accuracies are compared to see the optimal method of used classifiers. The final findings will be reported in the classification report stage.

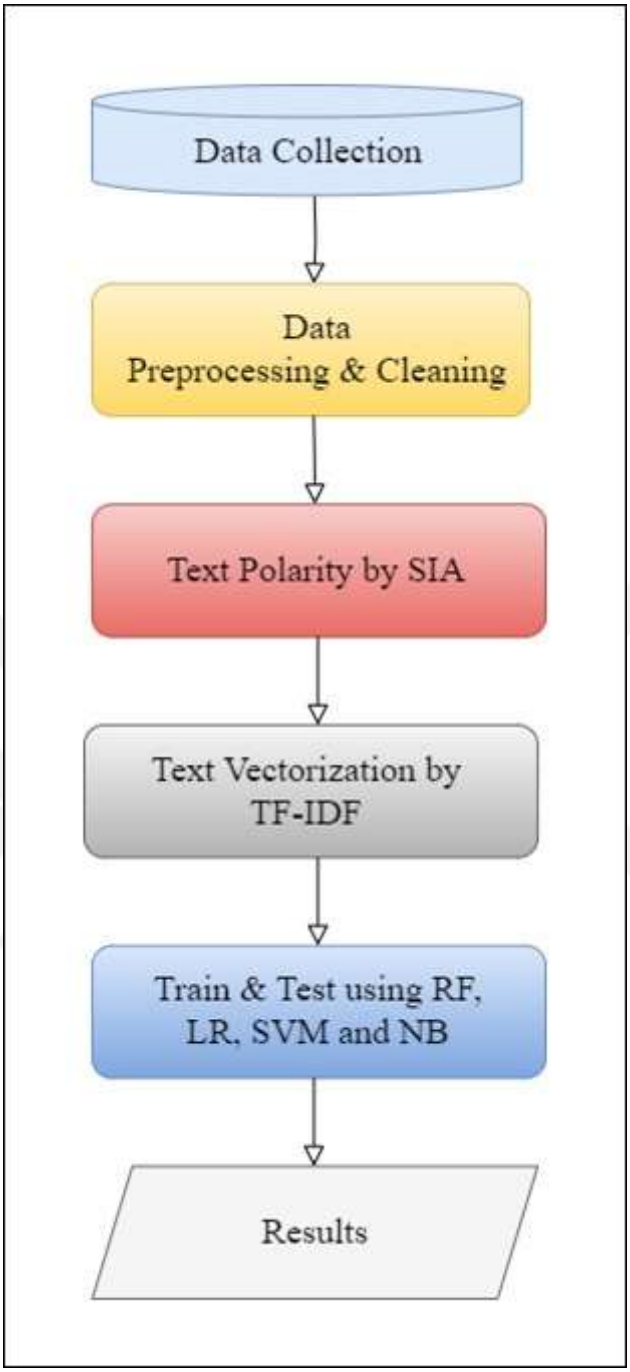

Figure 3.1: Sentiment Analysis Process Flowchart.

To create the thesis framework, readers need to understand the required approaches and procedures. Thus, this chapter covers methodology and principles used in this thesis concerning sentiment analysis. For the first step, we will start with the most required process which is Data Collection.

3.1 DATA COLLECTION

More than 10k of Tweets scraped from Twitter. Here is a sample of top five tweets in our dataset. (*Table 3.1*)

Table 3.1: Sample of The Collected Dataset.

Date	Tweets	Likes
2020-10-16	#Armenian armed forces launched a #missile attack on the second largest city Ganja …	424
2024-02-19	Azerbaijani President Ilham ALIYEV: "The Second Karabakh War and the anti-terrorist measures we took 5 months ago …	134
2024-03-14	!!□#Artsakh/Nagorno-#Karabakh citizen confirms #Russian peacekeepers sold humanitarian goods entering from #Armenia for a much higher price to the people.	102
2020-10-30	!!□#WarCrimes against #Azerbaijan/i civilians and still countries are silent. Humanitarian organisations should take action to stop this genocide by #Armenia …	220
2023-05-16	"""A legitimate question may arise: could Armenia not recognize the territorial integrity of Azerbaijan? In my opinion, after the 44-day war …	56

In this thesis project Twitter Application Project Interface (API) used to collect large datasets of tweets related to the topic. To get access the API user should create a Twitter and Twitter Development account. To ensure inclusion of diverse perspectives, the dataset will be filtered by relevant hashtags, keywords, and geolocation factors. Data collection will cover the duration of the Karabakh conflict, allowing for an in-depth analysis of tweets from the current conflict. After collecting the data, we will proceed to the next steps.

3.1.1 Twitter Api

In order to collect tweets related to the Karabakh conflict, Twitter Application Programming Interface (API) will be used. In this process we will use Python programming language and its library called Tweepy. There are plenty of functions in this library that require just few lines of script [19].

To install the package in Python, user first should install the library by typing the following code:

'!pip install tweepy' (3.1)

First of all, to extract the data from Twitter user have to set up a Twitter account and then, in order to access the data, user should create a Twitter Developer account. After completing these steps, to connect the data, we need API keys, such as:

a. *Consumer Key*

b. *Consumer Secret Key*

c. *Access Token*

d. *Access Token Secret Key*

3.2 DATA PREPROCESSING AND CLEANING

After completing the data collection, we will start to preprocess the data in order to prepare it for analysis. Data preprocessing is the step of converting the raw data into machine-readable format. To remove irrelevant data from the raw data that doesn't include any important information, preprocessing is required. To start the preprocessing stage first we must remove null values. After completing this step, we select the language in our coding to withdraw irrelevant tweets and work on only on selected language tweets. In our case it is tweets posted in English language. Pre-processing the data is a crucial step that requires the data cleaning procedure, which converts raw data into a format that is readable by machines. In this stage the data collected, including tweets, will be pre-processed for removing duplicate tweets, removing noise, including stopwords, URLs, emojis and special characters.

The aim of the preprocessing stage is to ensure that the data contains only appropriate, valuable, and useful data for sentiment analysis.

Data cleaning, commonly referred to as data cleansing, is an important phase in the data analysis process. Data cleaning contains itself removing unnecessary information to make the data ready for further steps. In this stage, removing stopwords, removing duplicates, also such characters as hashtags and null values removed from the data. Since we have a huge dataset consisting of 10k of tweets, data cleaning is necessary. After applying the steps listed above, the total number of tweets reduced to about 7k. Overall, data cleaning is essential for ensuring the quality and integrity of the dataset before performing further analysis or modelling. It helps researchers or analysts to minimize biases, errors, and inaccuracies that could compromise the validity and reliability of their findings.

3.2.1 Convert to Lower Case

Converting text data to lowercase is an essential step in data cleaning**.** Many language processing tools and libraries operate more effectively when text data is in lowercase. Lowercasing ensures compatibility with these tools and facilitates seamless integration into NLP pipelines and frameworks. To convert the letters in gathered dataset we run the code in next line:

$$data['Tweets'] = data['Tweets'].str.lower() \quad (3.2)$$

3.2.2 Removing Stopwords

Stopwords are the words in any language, which have no semantic meaning. It is an essential preprocessing step across different techniques in NLP. Removing stopwords helps build cleaner dataset for machine learning model. It is simple to create stopword lists to include terms that you want to avoid. Articles and pronouns are typically classed as stop words. These are some of stopwords, collected in data: “we”, “ourself”, “ours”, “in”, “is”, “a”, “the” etc.

It would not be desirable for these terms to take up space in our database or use extra processing time. To remove them easily from the data, we use NLTK (Natural Language Toolkit) library in Python, which stores a list of stopwords in 16 languages.

3.2.3 Lemmatization

Lemmatization is the step of analysing the morphological meaning of the term and returns the base word known as a lemma. Lemmatization uses the dictionary to identify the root word rather than just cutting off suffix and prefix. It is much slower than stemming but much more accurate. Stemming and lemmatization are both techniques used in natural language processing (NLP) to reduce words to their base or root forms. They are employed to normalize text, which helps in tasks such as text analysis, information retrieval, and machine learning. Here is the sample of lemmatization for better understanding [15]:

Example:

a. Word: “running”

b. Stemming: “run”

c. Lemmatization: “run”

Unlike stemming, which simply chops off affixes to derive the root form, lemmatization takes into account the morphological analysis of words based on their dictionary form or lemma.

3.3 SENTIMENT INTENSITY ANALYZER

A sentiment intensity analyzer (SIA) is an NLP tool that rates the sentiments expressed in text according to its intensity. Sentiment intensity analysis goes beyond traditional sentiment analysis by measuring the strength or intensity of the sentiment expressed in the text, instead of to simply classifying text as positive, negative, or neutral. In this stage we calculate the compound score of tweets. In the given sample, readers can easily understand the working principle of compound score.

pattern_sentence = 'I love this flower because this flower is beautiful'

sia.polarity_scores(pattern_sentence) (3.3)

{'neg': 0.0, 'neu': 0.426, 'pos': 0.574, 'compound': 0.8442}

Here we see that the sentence is positive and compound score is 0.8442, which means positive sentiment. Polarity is situated in between [-1, 1], -1 means negative sentiment, 0 neutral and 1 is positive sentiment. In Python programming language first, we create a function to calculate the polarity labels and then we apply it to dataset. Based on the outcome of our experiment, we observed that the distribution of sentiment polarities throughout the 7769 recordings formulated in the following way:

Table 3.2: Sentiment Polarities of The Collected Dataset.

Sentiment Score	**Number of Tweets**	**Percentage**
Positive	2083	27 %
Neutral	2292	30 %
Negative	3394	43 %
Total	**7769**	**100%**

4. WORDCLOUD

A popular method to display the most frequently used terms in a given text is to create a wordcloud visualization. To illustrate the most used words in visual we import WordCloud library in python programming language. A wordcloud is a figure that resembles a cloud and is filled with many words of various sizes and shapes. The size of each word indicates its importance or frequency; larger words are more frequently used [20]. It provides to give a summary of the data in terms of the most frequently occurring words [16].

To start creating visual from the given data first we must count the number of tweets as shown below:

tf = data['Tweets'].apply(lambda x: pd.value_counts(x.split())).sum (axis = 0).reset_index()

(4.1)

tf.columns = ['Word','tf']

After completing counting the words in the dataset we could display the top words by term frequency.

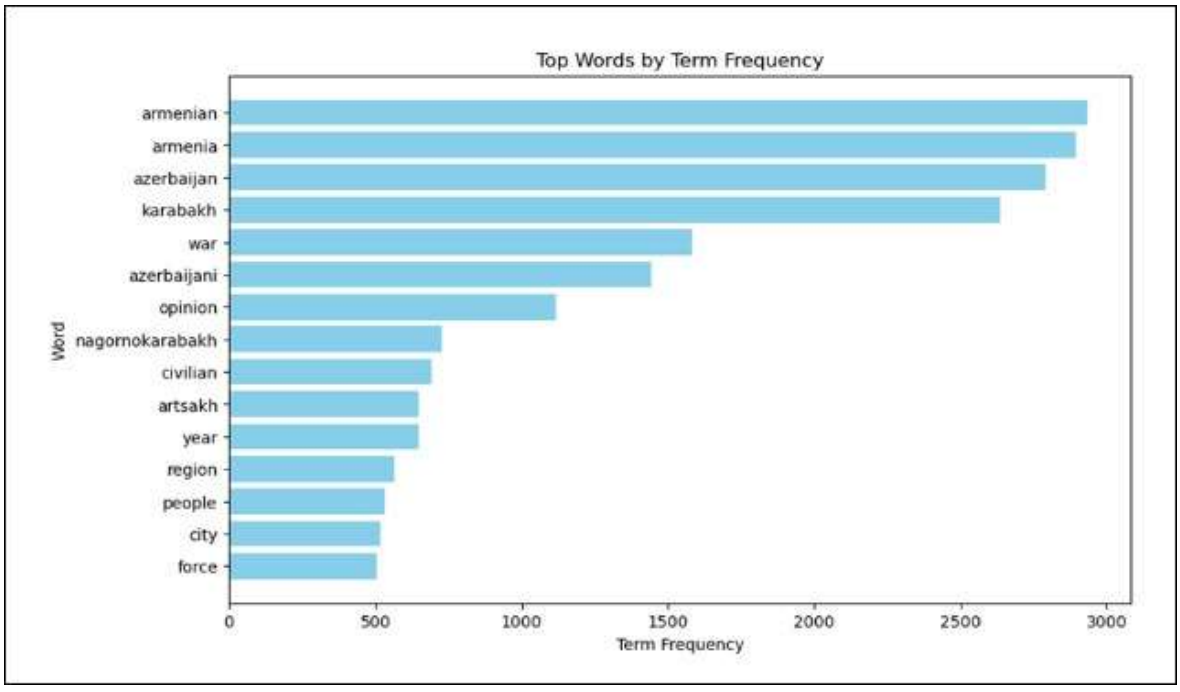

Figure 4.1: Top Words and Their Frequencies in The Corpus.

This visual displays the most used words and approximate number of times used. Next, we figured out the most common Bigrams. Bigrams is a combination of two words. There are additional Trigrams available. Bigram and trigram are types of n-grams, which are contiguous sequences of n items from a given sample of text, typically words. They are used in natural language processing (NLP) and text analysis to capture relationships and patterns within the text data. In our case we use Bigram to visualize the most common used 2 words in dataset.

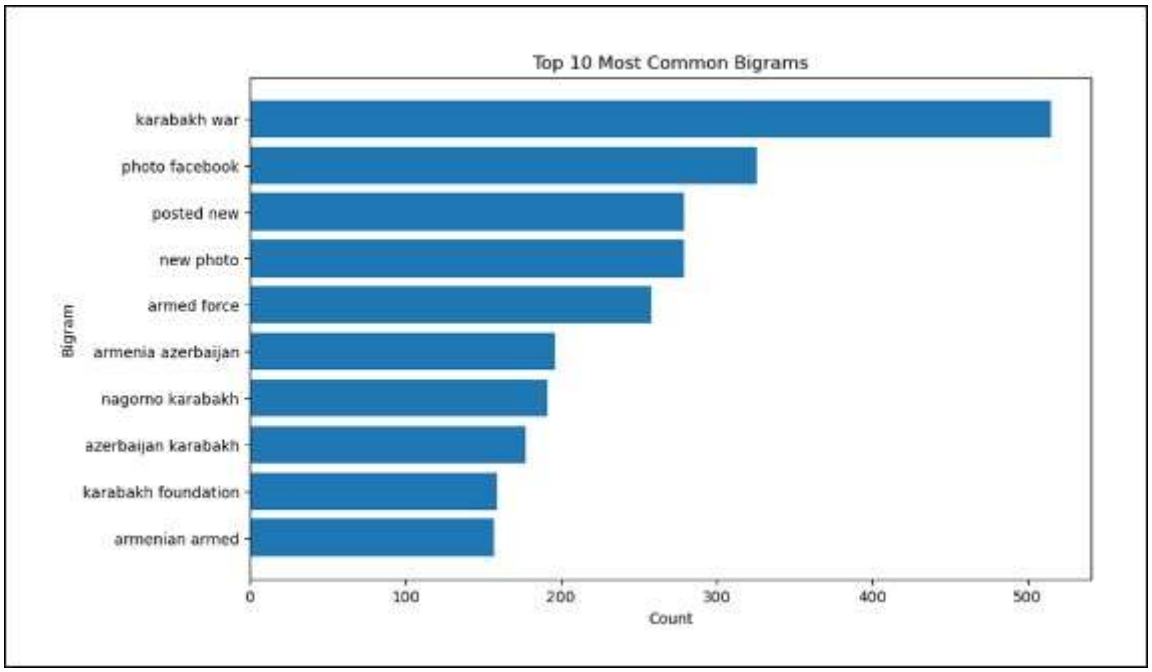

Figure 4.2: Top 10 Common Bigrams.

In this visual we can see that the most common bigram in our dataset is “karabakh war” and it is used more than 500 times. Lastly, we were able to display the main visual called WordCloud. In this stage frequently used words appear in a cloud form and various sizes, which the size of each word is typically determined by its frequency within a given dataset.

Figure 4.3: Wordcloud of The Dataset.

In this visual 50 frequently used words has appeared. This sizing method helps to visually understand the most significant or important words in the text, making it easier for readers to quickly identify key themes or topics.

5. MACHINE LEARNING MODEL

During this stage, the Machine Learning classifiers that will be utilized in this study will be examined. Machine Learning is the main part of this study. Firstly, let's give a brief information about ML and for what we will use it. Machine Learning is a subfield of Artificial Intelligence (AI) that relies on developing algorithms and models that computers can understand and could make predictions and choices without explicit programming. In simple terms, machine learning enables systems to automatically learn and improve through data or experience. There are two different kinds of algorithms in machine learning for both regression and classification. [17].

a. Supervised Learning

b. Unsupervised Learning

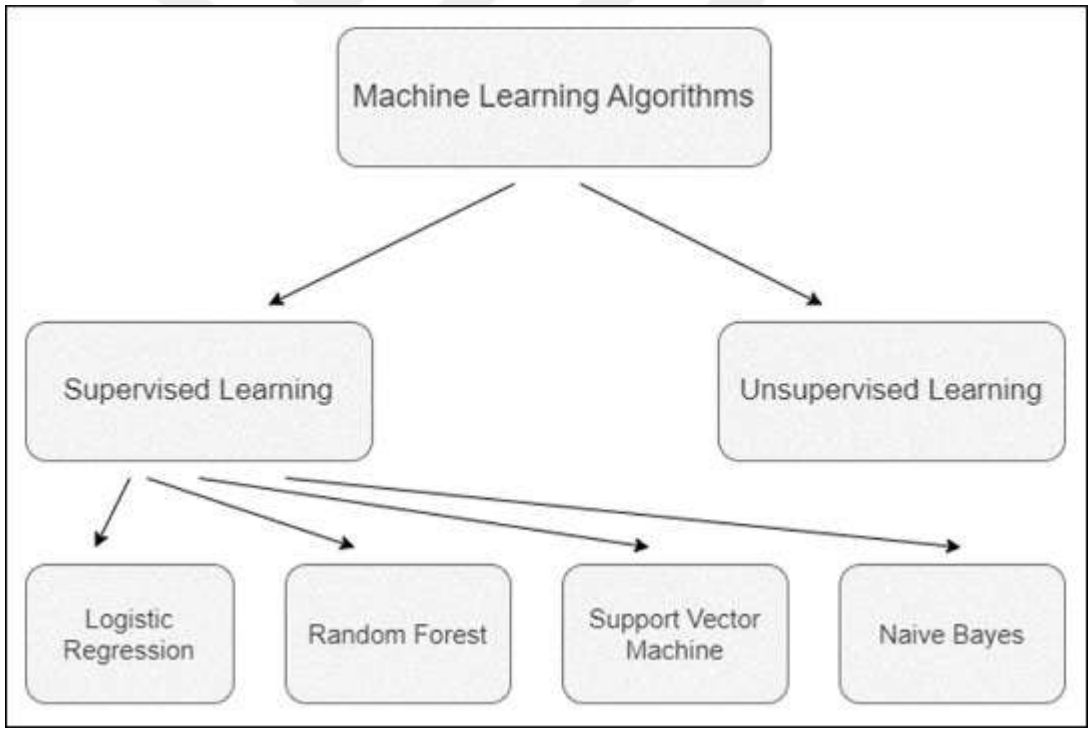

Figure 5.1: Machine Learning Diagram Approach.

As we said before, Logistic Regression, Random Forest, Support Vector Machine (SVM) and Naïve Bayes utilized in this study. At the last step the best performed ML model will be discussed. But before start to classify ML models we have to convert the textual data to numerical in order to machines could understand and analyze the dataset [32]. First, we have

to do a label encoder to convert categorical data into numeric values, by running the code below:

data['Polarity Label'] = LabelEncoder().fit_transform(data['Polarity Label']) (5.1)

Then, X would be defined as “Tweets” and Y as “Polarity Label”. After defining X and Y, next we have to implement vectorization for converting the raw data format into vectors that ML models support. In this study, we applied two essential text vectorization techniques, namely Count Vectorizer and TF-IDF (Term Frequency-Inverse Document Frequency), to process textual data in a numerical format suitable for machine learning algorithms. These techniques play a crucial role in natural language processing (NLP) tasks, providing a basis for feature extraction and text representation.

5.1 COUNT VECTORIZER

Count Vectorizer is a simple but effective technique that generates a collection of written text documents into a token count matrix. Every single row in the generated matrix corresponds to a document, while every column indicates a distinct word from the corpus. The cell's values represent the rate of every word in the relevant documents. This approach assigns a value of "1" when the word appears in the sentence, and "0" otherwise [18]. By using Count Vectorizer, we may efficiently handle big vocabularies and text datasets by obtaining a sparse representation of the textual data.

Simply, Count Vectorizer connects NLP techniques to ML. Let's demonstrate how the Count Vectorizer works on the provided sample text:

“This is a dog, This is a cat”

This vocabulary consists of four words which are: ['cat' 'dog' 'is' 'this']. Therefore, the vectors concerning the previous sentence, are shown below:

[[0, 1, 1, 1], [1, 0, 1, 1]].

The first row of the vectorized representation [0 1 1 1] corresponds to the first document 'This is a dog'. It has a count of 0 for 'cat', 1 for 'dog', 1 for 'is', and 1 for 'this'. The second row [1 0 1 1] corresponds to the second document 'This is a cat'. It has a count of 1 for 'cat', 0 for 'dog', 1 for 'is', and 1 for 'this'.

Next, we calculate vectors of N-grams Count Vectorizer. N-gram Count Vectorizer is a nethod used in Natural Language Processing (NLP) to generate feature vectors from text data that include both individual words (unigrams) and word sequences (n-grams). The N-grams Count Vectorizer functions similarly to the classic Count Vectorizer, but it covers n-grams as features rather than individual words.

In this step, we evaluate vectors of combination of two and three words. The vocabulary consists of five combinations of words: ['is cat' 'is dog' 'this is' 'this is cat' 'this is dog']. The vectors are like:

([[0, 1, 1, 0, 1], [1, 0, 1, 1, 0]])

The vectorized representation reflects the number of bigrams and trigrams in each document.

5.2 TF-IDF VECTORIZER

TF-IDF (Term Frequency - Invers Document Frequency) is a numerical statistic in NLP that calculates the significance of a term in a document in relation to a collection of documents. This is a mathematical-statistical method, which determines a word's importance to a text document's corpus [21]. Another aspect of this method is that it considers the corpus, which involves identifying similar words to the keywords found [1]. TF-IDF is a merging of TF and IDF. The weight is measured by multiplying the IDF by the TF, shown in eq 5.1 [22]. Term Frequency (TF) indicates how often a word appears in a single document (In longer documents, a word's frequency can be higher than in shorter documents, whereas its significance or importance may be inversely related to its frequency). The mathematical formula can be defined as:

$$TF(t,d) = \frac{N(t,d)}{T} \tag{5.2}$$

Here, TF is the term frequency of *t* in document *d*, *N* is the total number of times term *t* appears in document *d*, *T* is the total number of terms in document *d*.

Inverse Document Frequency (IDF) measures the importance of a term across a collection of documents. It is calculated using the formula:

$$IDF(t,D) = log(\frac{N}{DF(t,D)}) \tag{5.3}$$

In eq 5.3 *N* is the total number of documents in the corpus, *DF(t, D)* is the document frequency of term *t.* Once we have measured the TF and IDF scores, the TF-IDF score given by:

$$TF - IDF = TF \times IDF \tag{5.4}$$

Equation 5.3 measures the TF-IDF score.

After finishing the vectorization process, the next phase is to build ML models. As we mentioned before, ML classifiers such as Logistic Regression, Random Forest, Support Vector Machine and Naïve Bayes have been utilized in this study.

5.3 LOGISTIC REGRESSION

Logistic Regression is one of the ML algorithms which is used in this thesis project. Logistic regression is a linear model that use the logistic function to estimate the probability of a binary outcome. In this model we have a set of input features x_1, x_2 and a target variable y that we want to predict. In logistic regression model, the dependent variable is a binary variable with insights represented as 1 (YES) or 0 (NO), positive or negative. It calculates probabilities to discrete possibilities applying the Sigmoid function [23]. The logistic function returns the probability that the target variable (*y*) is 1 based on the input feature *x*.

The LR classifier has 3 types of regression [24]:

a. Binary Logistic Regression

b. Multinominal Logistic Regression

c. Ordinal Logistic Regression

In this present study, Binary Logistic Regression is utilized for classifying the data into positive, negative and neutral sentiments.

5.4 RANDOM FOREST

Random Forest is also a supervised ML model [10]. It can be implemented to solve both classification and regression problems in machine learning. It tries to reach single output by combining the multiple decisions. Simply, a random forest is a collection of decision trees,

where each of them differs slightly from the others. Each tree is trained independently on a random portion of the training dataset.

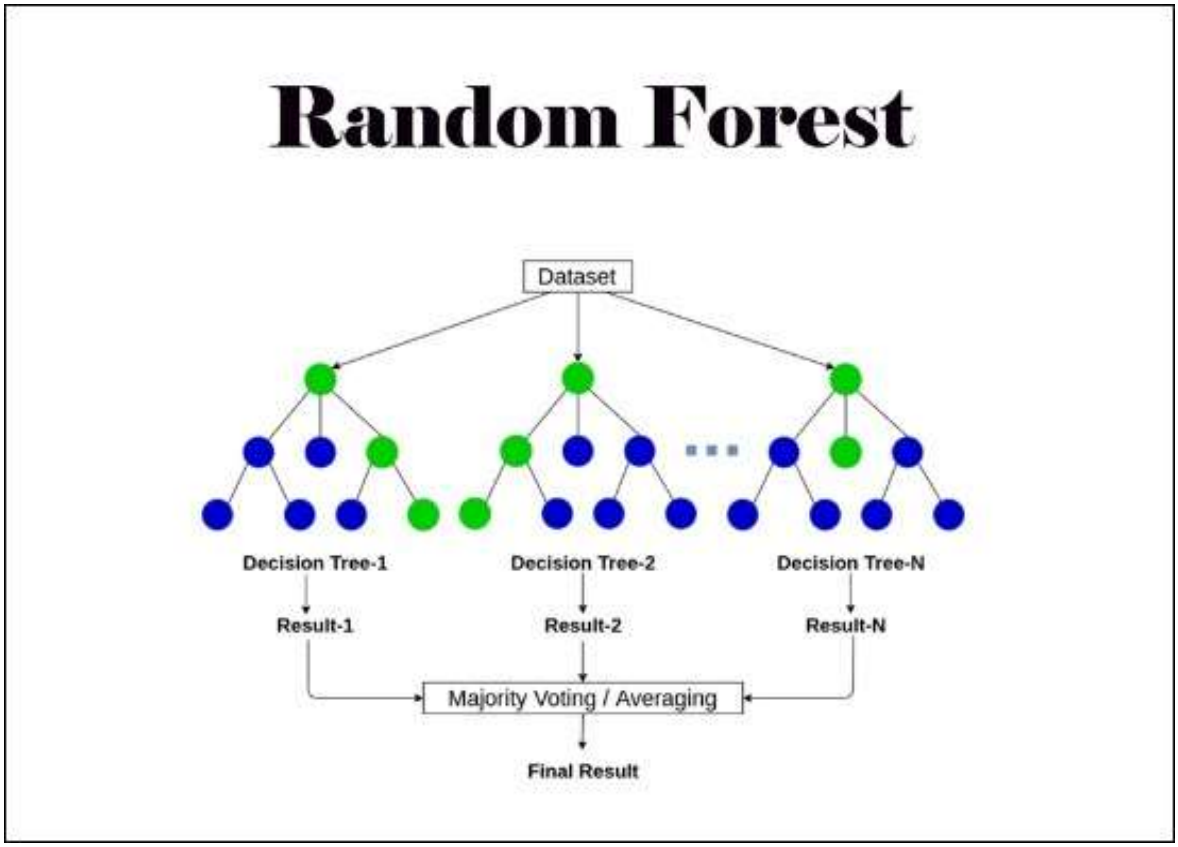

Figure 5.2: Random Forest Scheme.

Even decision trees face the issue with overfitting, Random Forest is an option for labelling and solving this issue. In Figure 5.2 the scheme of Logistic Regression is shown [25]. The main idea of the algorithm is each tree can do a good forecast, but it is possible that some of the data will be overfitted. Instead of depending on a single decision tree, the RF classifier forecasts the final result depending on the most predictions from each tree.

The RF classifier historically referred to an American scientist in 1995 considering the idea of random decision forests. In 2001 Leo Breiman invented the phrase Random Forest [20].

5.5 SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is an advanced version of Logistic Regression classifier. It can be applied to tasks involving regression as well as classification [26]. SVM aims to find a hyperplane in N-dimensional space (N is the number of features) that optimally divides the

two classes. The main goal of SVM is to find the possible line, that divides the data points into multiple data classes. In other terms, it is a straight line, that separates the plane into two portions, with each collection of data on one side of the line.

In SVM, we try to find the points that are closest to the line from both classes, these points are known as *support vectors*. Next, the margin will be calculated as the distance from the line and support vectors. The reason of using SVM is to increase the margin, as illustrated in the image below [27].

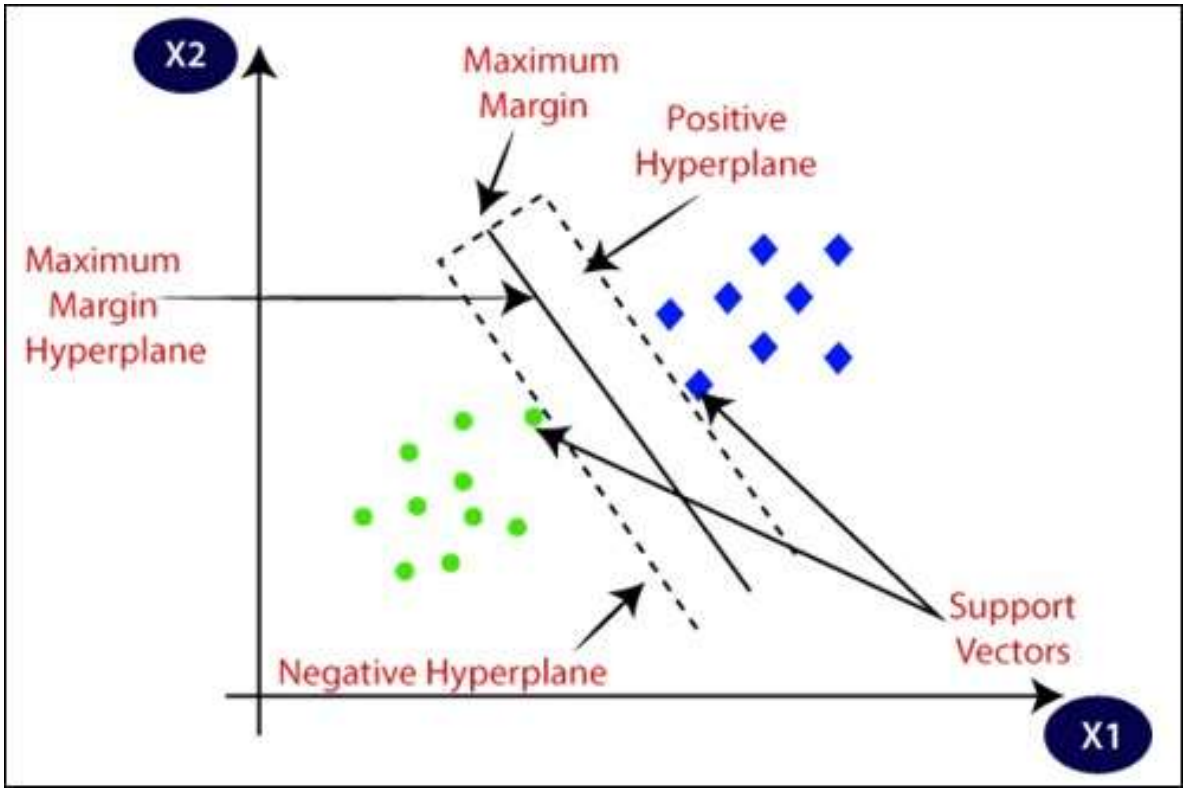

Figure 5.3: Support Vector Machine (SVM) Optimal Margin.

The dimensions of hyperplane are defined by the number of features. If the input has two features, then output is just one line. If there are three input features, the hyperplane comes to be a two-dimensional plane [28].

Support Vector Machines (SVMs) typically involve two tuning parameters, namely C and Gamma. A C parameter manages the trade-off between achieving a higher margin and avoiding classification errors on the training data. A smaller C number provides for a wider

margin but may result in more misclassifications. The Gamma variable in Linear SVM has been bypassed [29].

5.6 NAÏVE BAYES

The Naïve Bayes classifier is a probabilistic ML algorithm which is used for classification tasks based on Bayes' theorem. The goal of a Naive Bayes classifier is to split data into predefined categories or classes depending on the features available in the data [23]. The key objective of classifier is to produce accurate forecasts or classifications. It is commonly used in machine learning for classification tasks, especially in text classification and spam filtering.

As we said before, this model is determined by Bayes' theorem. Here is how Naïve Bayes model works:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \tag{5.5}$$

Where:

a. P(c|x) is the probability of class (c) given predictor (x).

b. P(x|c) is the probability of predictor (x) given class (c).

c. P(c) is the probability of given class (c).

d. P(x) is the probability of predictor (x).

There are three types of Naïve Bayes models:

Gaussian Naïve Bayes: Assumes features have a normal distribution. Suitable for categorization using continuous characteristics.

Bernoulli Naïve Bayes: This model works only with binary values. That is quite a simple model among the other Bayesian models. The most common example is when we check if each value indicates whether a word appears in a document or not.

Multinomial Naïve Bayes: The Multinomial Naive Bayes classifier is a variant of the Naive Bayes algorithm which is used to calculate the probability of textual data. In this thesis project Multinomial Naive Bayes has been used to calculate accuracies of Twitter data.

By creating ML models listed above we were able to find accuracy and F1 scores to proceed to the model hypertuning section with the higher accuracy score models.

Table 5.1: Accuracy and F1-ratings of ML Models.

Model	**Accuracy**	**F1 score**
Logistic Regression	0.7855607240024497	0.7712690309366601
Random Forest	0.7439860244753999	0.7066653354804666
SVM	0.7850450118962675	0.772609494068428
Naïve Bayes	0.7053627263543555	0.6869936188212826

Based on the table provided, it is evident that the accuracy and F1 scores of both Support Vector Machine (SVM) and Logistic Regression models meet the criteria for hyperparameter tuning.

6. MODEL HYPERTUNING

Hyperparameter tuning in machine learning is the process of determining the best hyperparameters for a given machine learning algorithm. As we said before, SVM and Logistic Regression performed well for this stage. First, the dataset is splitted into training and testing subsets, then the hyperparameter grid defined for Logistic Regression as shown below:

```
param_grid = {
'C': [0.001, 0.01, 0.1, 1, 10, 100],
'penalty': ['l1', 'l2']    }
```
(6.1)

In parameter C values means regularization. Smaller values represent stronger regularization. Penalty parameters L1 (lasso) and L2 (ridge) are the methods of regularization for linear regression models that includes a penalty term into the loss function for preventing overfitting. L1 (lasso) adds the coefficients' absolute values to the loss function, encouraging model sparsity. L2 (ridge) regression implements a comparable constraint on the coefficients via a penalty factor.

Following the grid search with 5-fold cross-validation, the results are evaluated individually for the TF-IDF and Count vectorizers.

Table 6.1: Results of LR for Count Vectorizer.

Best Parameters	**Test Accuracy**	**Test F1 Score**
{'C': 10, 'penalty': 'l2'}	0.9099099099099099	0.9019964507004533

Table 6.2: Results of LR for TF-IDF.

Best Parameters	**Test Accuracy**	**Test F1 Score**
{'C': 100, 'penalty': 'l2'}	0.9028314028314028	0.8943716402317049

Next, the results are assessed for Support Vector Machine (SVM). As the previous step, first we divided the dataset into training and testing sets, then define hyperparameter distribution for Randomized Search.

```
param_dist = {
'C': uniform(0.01, 100),
'kernel': ['linear', 'rbf', 'poly'],    }
```
(6.2)

The code written above describes a parameter distribution for usage in randomized search. The C specifies the hyperparameter for a SVM model. Uniform formulates the random numbers between 0.01 and 100. The Kernel function specifies the options for the kernel function that will be used in the model. The parameters provided here are 'linear', 'rbf' (radial basis function), and 'poly' (polynomial kernel).

By performing randomized search with 5-fold cross-validation, the obtained results are as follows:

Table 6.3: Results of SVM for Count Vectorizer.

Best Parameters (SVM)	**Test Accuracy**	**Test F1 Score**
{'C': 46.45121224689047, 'kernel': 'linear'}	0.9073359073359073	0.9002262270241514

Table 6.4: Results of SVM for Count Vectorizer.

Best Parameters (SVM)	**Test Accuracy**	**Test F1 Score**
{'C': 21.407876717426145, 'kernel': 'rbf'}	0.8944658944658944	0.884141117383945

From the outcomes shown in Table 6.1 and Table 6.2, we can see that Logistic Regression achieved great scores for hypertuning compared to SVM.

7. RESULTS AND DISCUSSION

In this section the results of ML model will be proposed and discussed. As we demonstrated before, during the preceding steps Logistic Regression performed high accuracy scores by eliminating Random Forest, SVM and Naïve Bayes classifiers. In this part of thesis study, experimental results of the classification report and confusion matrix have been investigated and evaluated.

The classification report is a text-based summary report and is a crucial stage in machine learning. It is a thorough evaluation tool that determines the performance of a classification model. The classification report visualizer provides precision, f1-score, recall and support results, as well as weighted average of these indicators also included for the model. These are fundamental metrics in classification tasks.

Accuracy is a measure of correctness of the model across all the dataset. It shows the proportion of correctly expected cases to total instances. Precision works on our table's vertical lines (columns). The formula of accuracy is:

$$Accuracy = \frac{True\ Positive\ (TP) + \ True\ Negative\ (TN)}{All\ (TP + TN + FN + FP)} \tag{7.1}$$

Precision is a measure of positive prediction accuracy that indicates the proportion of true positive predictions created by the model. In other words, precision indicates the number of percent of correct predictions.

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)} \tag{7.2}$$

Recall, in the context of classification, measures the classifier's capability to identify all positive instances correctly. Theoretically, recall is determined as the ratio of true positives (properly predicted positive cases) to the sum of true positives and false negatives (positive instances incorrectly expected as negative):

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)} \tag{7.3}$$

F1-score is the balance between precision and recall, in other words it is harmonic mean of precision and recall, providing a balanced measure that takes into consideration both precision and recall. The F1-score is used when the data is imbalanced. F1-score levels between 0 and 1, where 1 represents complete precision and recall, while 0 indicate neither perfect precision nor recall.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7.4}$$

Support refers to the number of actual instances of each class in the dataset. It is essentially the number of true examples for each class, giving context to the precision, recall, and F1-score measures. It doesn't vary between models, instead it diagnoses the evaluation process.

In summary, classification report of Logistic Regression classifier has been done and main measurements such as precision, recall, f1-score, and support scores are presented below. The following results present the outcomes of the Count Vectorizer and TF-IDF approaches.

```
report_lr = classification_report(test_y, y_pred)
print(report_lr)
```

```
              precision    recall  f1-score   support

           0       0.91      0.94      0.92       881
           1       0.91      0.88      0.89       558
           2       0.86      0.83      0.84       504

    accuracy                           0.89      1943
   macro avg       0.89      0.88      0.89      1943
weighted avg       0.89      0.89      0.89      1943
```

Figure 7.1: Classification Report of Logistic Regression for TF-IDF.

```
report_lr = classification_report(test_y, y_pred)
print(report_lr)
```

```
              precision    recall  f1-score   support

           0       0.92      0.93      0.93       881
           1       0.90      0.91      0.90       558
           2       0.86      0.84      0.85       504

    accuracy                           0.90      1943
   macro avg       0.89      0.89      0.89      1943
weighted avg       0.90      0.90      0.90      1943
```

Figure 7.2: Classification Report of Logistic Regression for Count Vectorizer.

The other way to visualize the predictions is confusion matrix. A confusion matrix is a structure of table that displays how well an ML model classifies the actual and expected outcomes. It is a comparison of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) of each classification class.

There are simply four approaches to determine whether the predictions are right or wrong [30]:

a. True Negative (TN): The output is negative, and the predicted is negative.

b. True Positive (TP): The output is positive, and the predicted is positive.

c. False Negative (FN): The output is positive, and the predicted is negative.

d. False Positive (FP): The output is negative, and the predicted is positive.

The confusion matrix is usually demonstrated as a 2x2 matrix, but it can be enlarged to support multiple classes. In the Figure 7.3 a 2x2 confusion matrix is displayed [31].

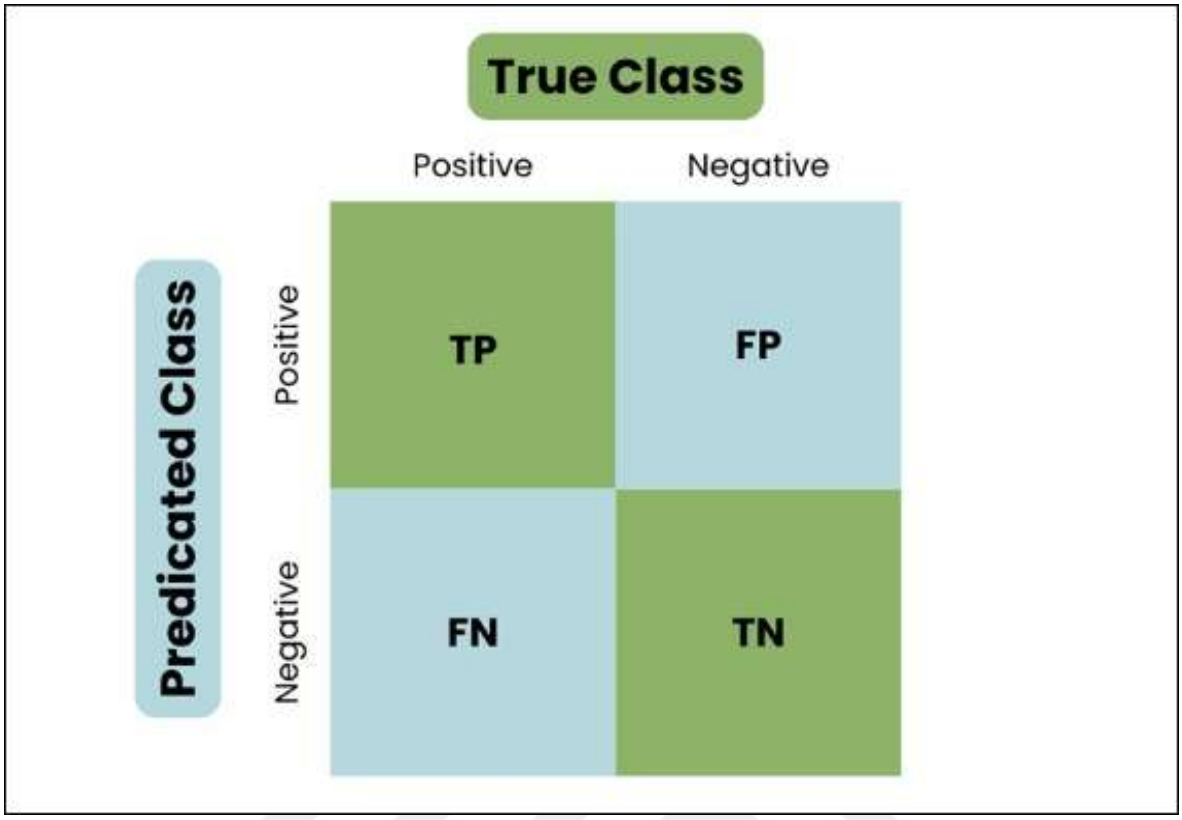

Figure 7.3: Confusion Matrix Table.

At the end, both the confusion matrix and classification report are essential tools for evaluating the effectiveness of machine learning classification models. The main difference between a classification report and a confusion matrix is that they serve different functions for distinct audiences. The confusion matrix proves valuable when comparing classifications, identifying misclassifications between classes, or displaying results in a tabular format. At the same time, the classification report is more appropriate when measuring precision, recall, and F1-score for each class, or when presenting results with numerical values or detailed text.

8. CONCLUSION

In today’s interconnected digital world, sentiment analysis plays crucial role for understanding the public's views, consumer behaviour, and social changes. Sentiment analysis not only allows organizations to monitor and manage their online reputation, but it also helps them find new opportunities, avoid risks, and make data-driven decisions in real time. Furthermore, sentiment analysis has significance role in areas such as political analysis, public health monitoring, and disaster response, where determining public mood can provide with policymaking, crisis management, and resource allocation.

In proposed thesis paper, sentiment analysis as an application of NLP and ML have been implemented on tweets related to the Karabakh conflict subject. The primary function of this research is making Twitter-resistant sentiment analysis framework based on the sentiments expressed in Twitter regarding the outcome of the Karabakh conflict.

The present study is started by data collection stage by following the most important steps like data preprocessing and cleaning, which includes removing duplicates, converting to lowercase, removing stopwords etc. These procedures are necessary to prepare data for vectorization. After the data preparation the SIA applied to calculate the polarity label for each tweet. According to SIA each tweet in collected dataset is assigned as positive, negative, and neutral. For binary classification Count Vectorizer and TF-IDF vectorizer applied for the extraction of features. In this stage TF-IDF vectorizer performed high accuracy scores compared to Count Vectorizer. In ML models Logistic Regression, Random Forest, Support Vector Machine and Naïve Bayes classifiers have been implemented in this study. Through the various steps, Logistic Regression classifier is done with high accuracy rating scores by eliminating RF, SVM, and NB models. Other relevant findings of classification report and confusion matrix have been given in results section.

REFERENCES

[1] J. Singh and P. Tripathi, "Sentiment analysis of Twitter data by making use of SVM, Random Forest and Decision Tree algorithm," in *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, 2021, pp. 193–198.

[2] H. Parveen and S. Pandey, "Sentiment analysis on Twitter Data-set using Naive Bayes algorithm," *in 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, 2016, pp. 416–419.

[3] M. Faisal, Z. Abouelhassan, F. Alotaibi, R. Alsaeedi, F. Alazmi, and S. Alkanadari, "Sentiment analysis using machine learning model for Qatar world cup 2022 among different Arabic countries using twitter API," *in 2023 IEEE World AI IoT Congress (AIIoT)*, 2023, pp. 0222–0228.

[4] A. Madan and U. Ghose, "Sentiment analysis for twitter data in the Hindi language," *in 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2021, pp. 784–789.

[5] F. M. J. Mehedi Shamrat et al., "Sentiment analysis on twitter tweets about COVID-19 vaccines usi ng NLP and supervised KNN classification algorithm*," Indones. J. Electr. Eng. Comput. Sci.*, vol. 23, no. 1, p. 463, 2021.

[6] G. Prema Arokia Mary, M. S. Hema, R. Maheshprabhu, and M. Nageswara Guptha, "Sentimental analysis of twitter data using machine learning algorithms," *in 2021 International Conference on Forensics, Analytics, Big Data, Security (FABS)*, 2021, vol. 1, pp. 1–5.

[7] U. D. Gandhi, P. Malarvizhi Kumar, G. Chandra Babu, and G. Karthick, "Sentiment analysis on twitter data by using convolutional neural network (CNN) and long short term memory (LSTM)," *Wirel. Pers. Commun.*, 2021.

[8] F. I. Maulana, P. D. P. Adi, D. Lestari, A. Purnomo, and S. Y. Prihatin, "Twitter data sentiment analysis of COVID-19 vaccination using machine learning," *in 2022 5th*

International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2022, pp. 582–587.

[9] K. M. Hasib, M. A. Habib, N. A. Towhid, and M. I. H. Showrov, "A novel deep learning based sentiment analysis of twitter data for US airline service," *in 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, 2021, pp. 450–455.

[10] A. S. Neogi, K. A. Garg, R. K. Mishra, and Y. K. Dwivedi, "Sentiment analysis and classification of Indian farmers' protest using twitter data*," International Journal of Information Management Data Insights*, vol. 1, no. 2, p. 100019, 2021.

[11] S. Singh, K. Kumar, and B. Kumar, "Sentiment analysis of twitter data using TF-IDF and machine learning techniques," *in 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, 2022, vol. 1, pp. 252–255.

[12] A. Chiorrini, C. Diamantini, A. Mircoli, and D. Potena, "Emotion and sentiment analysis of tweets using BERT," *EDBT/ICDT Workshops*, 2021.

[13] D. Rathod, K. Patel, A. J. Goswami, S. Degadwala, and D. Vyas, "Exploring drug sentiment analysis with machine learning techniques*," in 2023 International Conference on Inventive Computation Technologies (ICICT)*, 2023, pp. 9–12.

[14] K. S. Madhu, B. C. Reddy, C. H. Damarukanadhan, M. Polireddy, and N. Ravinder, "Real Time Sentimental Analysis on Twitter," *in 2021 6th International Conference on Inventive Computation Technologies (ICICT)*, 2021, pp. 1030–1034.

[15] A. S. Gillis, "What is lemmatization?," Enterprise AI, 13-Mar-2023. [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/lemmatization. [Accessed: 18-May-2024].

[16] Wang, J., Zhao, J., Guo, S., North, C., & Ramakrishnan, N.. ReCloud: semantics-based word cloud visualization of user reviews. *In Graphics Interface 2014*, 2020, pp. 151-158.

[17] A. Géron, Hands-on machine learning with Scikit-Learn, keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. Heidelberg, Germany: O'Reilly, 2019.

[18] O. Rakhmanov, "A comparative study on vectorization and classification techniques in sentiment analysis to classify student-lecturer comments," *Procedia Comput. Sci.*, vol. 178, pp. 194–204, 2020.

[19] Shibab, F. F. S. Sentiment Analysis and Tweet Classification Using Machine Learning, Ph.D. dissertation, Computer Engineering, Karabuk University, Karabuk, 2022.

[20] A. Gupta, A. Singh, I. Pandita, and H. Parashar, "Sentiment Analysis of Twitter posts using machine learning algorithms," *in 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom),* 2019, pp. 980–983.

[21] Y. Yang, "Research and realization of internet public opinion analysis based on improved TF - IDF algorithm," *in 2017 16th International Symposium on Distributed Computing and Applications to Business*, Engineering and Science (DCABES), 2017, pp. 80–83.

[22] B. Ray, A. Garain, and R. Sarkar, "An ensemble-based hotel recommender system using sentiment analysis and aspect categorization of hotel reviews," *Appl. Soft Comput.*, vol. 98, no. 106935, p. 106935, 2021.

[23] A. Poornima and K. S. Priya, "A comparative sentiment analysis of sentence embedding using machine learning techniques," *in 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS),* 2020, pp. 493–496.

[24] *Logistic Regression — Detailed Overview*, Medium, March 2018 [Online] Available: https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc

[25] Alkurdi, A. A. (2023). Enhancing Heart Disease Diagnosis Using Machine Learning Classifiers. *Fusion: Practice and Applications*, 13(1), 08-18.

[26] S. Zahoor and R. Rohilla, “Twitter sentiment analysis using machine learning algorithms: A case study,” in *2020 International Conference on Advances in Computing, Communication & Materials (ICACCM)*, 2020, pp. 194–199.

[27] *Support Vector Machine (with numerical example),* Medium, January 2023, [Online] Available: https://medium.com/@balajicena1995/support-vector-machine-with-numerical-example-8dfe81eae4f0

[28] *Rohith Gandhi (2018, June 7), Support Vector Machine — Introduction to Machine Learning Algorithms,* Medium, June 2018, [Online] Available: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[29] C.-Z. Liu, Y.-X. Sheng, Z.-Q. Wei, and Y.-Q. Yang, “Research of text classification based on improved TF-IDF algorithm,” *in 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE),* 2018, pp. 218–222.

[30] S. Dey, S. Wasif, D. S. Tonmoy, S. Sultana, J. Sarkar, and M. Dey, “A comparative study of support vector machine and naive Bayes classifier for sentiment analysis on Amazon product reviews,” *in 2020 International Conference on Contemporary Computing and Applications (IC3A),* 2020, pp. 217–220.

[31] *Stuck in life?…Here is a way to declutter your life and get moving again,* Medium, March 2024, [Online] Available: https://medium.com/@bleed.inink/stuck-in-life-here-is-a-way-to-declutter-your-life-and-get-moving-again-4ec7fd3b4be9

[32] A. S. Neogi, K. A. Garg, R. K. Mishra, and Y. K. Dwivedi, “Sentiment analysis and classification of Indian farmers’ protest using twitter data,” *International Journal of Information Management Data Insights,* vol. 1, no. 2, p. 100019, 2021.