# TWITTER SENTIMENT ANALYSIS OF IKEA

by

Nahida Muhammad Umer

Submitted to Graduate School of Natural and Applied Sciences

in Partial Fulfillment of the Requirements

for the Degree of Master of Science in

Computer Engineering

Yeditepe University

2023

TWITTER SENTIMENT ANALYSIS OF IKEA

APPROVED BY:

Assist. Prof. Dr. Tacha Serif .....
(Thesis Supervisor)
(Yeditepe University)

Assist. Prof. Dr. Onur Demir .....
(Yeditepe University)

Prof. Dr. Semih Bilgen .....
(Istanbul Okan University)

DATE OF APPROVAL: …../…./2023

I hereby declare that this thesis is my own work and that all information in this thesis has been obtained and presented in accordance with academic rules and ethical conduct. I have fully cited and referenced all material and results as required by these rules and conduct, and this thesis study does not contain any plagiarism. If any material used in the thesis requires copyright, the necessary permissions have been obtained. No material from this thesis has been used for the award of another degree.

I accept all kinds of legal liability that may arise in case contrary to these situations.

Name, Last Name Nahida Muhammad Umer

Signature .....

# ACKNOWLEDGEMENTS

It is with immense gratitude that I acknowledge the support and help of my thesis supervisor Assist. Prof. Dr Tacha Serif. His continuous support and valuable advises had significant impact on the success of the project. Pursuing my thesis under his supervision has been an experience in broadening my understanding by presenting unlimited sources of learning.

I also want to thank the faculty and staff at the Computer Science department of Yeditepe university for their help in expanding my knowledge within the machine learning field.

Finally, I would like to thank my family for their endless love and support. Your kindness and care makes everything more beautiful.

# ABSTRACT


## TWITTER SENTIMENT ANALYSIS OF IKEA

Sentiment analysis is a natural language processing technique to classify textual data. It is one of the main approaches in machine learning that is used to enhance decision-making by gaining knowledge underlying the growing amount of historical online text data. In this thesis project, the sentiment analysis method was applied to analyze the perception of people on IKEA products. As part of this study overall, 60,000 tweets were collected from the Twitter platform. After predicting the polarity of the tweets, they were divided into categories to investigate the polarity of users on IKEA products. Multiple machine learning methods were used to predict the polarity and RoBERTa had the highest accuracy of 0.70 and f1-score of 0.7023. However, Naive Bayes also performed well, with an accuracy of 0.65 and f1-score of 0.65. On the other hand, traditional machine learning algorithms including KNN and BERT show poor performance. The outcomes of this work provide a prototype on future performance analysis considering user attitudes, perceptions, and opinions on IKEA. Towards this goal, an academic approach was followed by performing exploratory data analysis experiments in order to investigate the reason for frequent user mood changes in social media.

# ÖZET

## IKEA'NIN TWITTER DUYGU ANALİZİ

Duygu analizi, metinsel verileri sınıflandırmak için kullanılan doğal bir dil işleme tekniğidir. Artan miktarda tarihsel çevrimiçi metin verilerinin altında yatan bilgileri elde ederek karar vermeyi geliştirmek için kullanılan makine öğrenimindeki ana yaklaşımlardan biridir. Bu tez projesinde, insanların IKEA ürünlerine ilişkin algısını analiz etmek için duygu analizi yöntemi uygulanmıştır. Genel olarak bu çalışmanın bir parçası olarak, Twitter platformundan 60.000 tweet toplandı. Tweetlerin polaritesini tahmin ettikten sonra, kullanıcıların IKEA ürünleri üzerindeki polaritesini araştırmak için kategorilere ayrılmıştır. Polariteyi tahmin etmek için çoklu makine öğrenimi yöntemleri kullanıldı ve RoBERTa 0,70 ile en yüksek doğruluğa ve 0,7023 ile f1 puanına sahipti. Ancak Naive Bayes de iyi bir performans sergiledi. 0,65 doğruluk ve 0,65 f1 skoru ile. Öte yandan, KNN ve BERT gibi geleneksel makine öğrenimi algoritmaları düşük performans gösteriyor. Bu çalışmanın sonuçları, IKEA ile ilgili kullanıcı tutumlarını, algılarını ve görüşlerini dikkate alan gelecekteki performans analizi hakkında bir prototip sunmaktadır. Bu amaca yönelik olarak, sosyal medyada sık görülen kullanıcı ruh hali değişikliklerinin nedenini araştırmak için keşifsel veri analizi deneyleri yapılarak akademik bir yaklaşım izlenmiştir.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF SYMBOLS/ABBREVIATIONS

| | |
|---|---|
| BERT | Bidirectional encoder representations from transformers |
| CNN | Convolutional neural network |
| ELMO | Embeddings from language models |
| GAN | Generative adversarial networks |
| GPT-3 | Generative pre-trained transformer 3 |
| GRU | Gated recurrent unit |
| LSTM | Long short-term memory |
| NLP | Natural language processing |
| NLP-RL | Natural language processing reinforcement learning |
| RL | Reinforcement learning |
| RoBERTa | Robustly optimized BERT pre-training |
| SENT | Sentiment analysis |
| seq2seq | sequence-to-sequence transformer-based models |
| Transformer | Attention-based neural network architecture |
| VAE | Variational autoencoder |
| WORDEMB | Word embeddings |

# 1. INTRODUCTION

Natural language processing techniques are used for analyzing textual data to identify and extract subjective information from source materials [1]. Recently, there has been a growing interest in sentiment analysis to understand the emotions and opinions of social media users. As a result, social media platforms such as Facebook, Twitter, and Instagram, have become essential data sources for companies and organizations looking to gauge public sentiment on various topics [1].

However, one of the main challenges in sentiment analysis of social media is the informal and unstructured nature of the data. Unlike traditional data sources, such as surveys and numerical datasets, social media posts often contain slang words, misspellings, punctuality, and other linguistic variations that make them difficult to analyze [2]. Additionally, the brevity of most social media posts means that the context in which sentiments are expressed often needs to be clarified or completed [3]. Despite these challenges, sentiment analysis of social media has several potential applications. For example, it can monitor brand reputation, detect emerging trends, identify potential risks or opportunities, and inform decision-making processes [4]. One recent development in social media sentiment analysis is the use of machine learning algorithms. These algorithms are designed to provide deeper insights into specific features of the text by associating them with positive and negative polarities [5].

## 1.1. LEVELS OF SENTIMENT ANALYSIS

Textual information is usually written in sentences. However, when multiple sentences are used to express a view, the entire paragraph or document needs to be considered [6]. Thus, based on the format and content of the text, sentiment analysis can be classified into multiple levels. These consists of:

- Document level where sentiment analysis considers the overall sentiment of a document or collection of texts. In this approach, long-term memory approaches are necessary to connect the sentences.
- Sentence level where the sentiment of individual sentences or clauses is analyzed.

Analysis at this level is most common due to the brevity of social media posts [7]. This approach is used in this thesis since the data was collected from Twitter.

- Aspect level where sentiment analysis involves analyzing specific aspects or features of a product or service by connecting multiple words in a sentence rather than one word [8].
- Multi-domain where sentiment is analyzed by considering multiple domains to account for words and phrases that may refer to opposite meanings in different domains [9].
- Multi-modal analysis is conducted to combine text with other information from other types of data such as video and photo [10].

Moreover, sentiment analysis can also be classified based on the algorithms and techniques used. For example, some methods use machine learning algorithms to classify texts as positive, negative, or neutral [11]. Whereas other methods use dictionaries or lexicons of words and phrases associated with positive or negative sentiments [12]. Also, other methods use more advanced techniques, such as deep learning algorithms, to provide more insight into text-specific features associated with different sentiments [13].

## 1.2. ADVANCEMENTS IN SENTIMENT ANALYSIS

Traditionally, sentiment analysis of tweets has relied on text-based methods, such as dictionaries or lexicons of words and phrases associated with positive or negative sentiments. These methods are relatively simple and easy to implement, but they can be limited in their ability to capture more subtle or complex sentiments [14]. On the other hand, ML algorithms can be trained on large datasets of annotated tweets to learn patterns in the language associated with different sentiments [15].

To make more accurate predictions about the sentiment of new unseen tweets, various ML methods are used for sentiment analysis of tweets that include support vector machines (SVMs) and Naive Bayes classifiers [15] as well as deep learning algorithms. Deep learning is particularly well-suited to analyzing text data and it has been shown to achieve high levels of accuracy in sentiment analysis tasks [16]. Convolutional neural networks (CNNs) and

recurrent neural networks (RNNs) that are two types of deep learning algorithms have been successfully applied for twitter sentiment analysis [16]. One of the most widely used algorithms for sentiment analysis of tweets is the BERT (Bidirectional Encoder Representations from Transformers) model. BERT is a transformer-based model trained to predict masked words in sentences based on the context of the other words in the text. BERT, that was developed by Google in 2018 [17]. This was followed by the development of Robustly optimized BERT approach (RoBERTa) by Facebook in 2020. It improved performance of the model by using a more extensive training dataset as well as additional training steps [18]. RoBERTa achieved state-of-the-art results on various natural language processing tasks, including sentiment analysis [19].

In conclusion, several traditional text-based, ML, and DL methods have been applied to conduct sentiment analysis on tweets. While traditional methods are relatively simple and easy to implement, they can be limited in capturing more subtle or complex sentiments. ML and DL classifiers, on the other hand, have the potential to achieve higher levels of accuracy. However, they require more computational resources and they are more challenging to interpret. BERT and RoBERTa are examples of DL algorithms that have been applied to sentiment analysis of tweets and have achieved state-of-the-art results on a variety of natural language processing tasks. Further studies have used these models to analyze sentiments for various products using social media data.

### 1.3. RESEARCH AIM

This study aims to evaluate the performance of various sentiment analysis methods by investigating tweets on IKEA, the Swedish home furnishings company which is a well-known brand worldwide. Extracting and analyzing bi-gram and tri-gram features that are most associated with the obtained sentiments. [20]. With a large customer base and a robust online presence, IKEA is a prime candidate for evaluating sentiment analysis on tweet data. By analyzing the sentiment of tweets about IKEA, various areas of concern or strength can be identified to make changes or improvements [21]. Sentiment analysis of tweets about IKEA can also help the company to identify trends and patterns in customer feedback [22].

## 1.4. RESEARCH QUESTIONS AND CONTRIBUTION

The main objectives of this study are to construct an overview of current algorithms, techniques, and tools for brand sentiment analysis and support further upcoming research in that field. Thus, the main research questions in this thesis are:

- How do different sentiment analysis methods perform on an Ikea dataset of 60,000 tweets? And what are the strengths and limitations of each approach in this context?

  For that purpose, the research will compare the performance of different methods and classify the polarity to understand the strengths and limitations of each approach in analyzing consumer sentiment.

- What are the bi-gram and tri-gram features of the IKEA dataset that are most associated with the positive, negative, and neutral sentiments. Also, how do these features differ between different sentiment analysis methods?

  The study will extract and analyze sentiment features using bi-gram and tri-gram. The research will also compare the identified features between different sentiment analysis methods to understand how they differ.

## 1.5. THESIS ORGANIZATION

The thesis is divided into seven chapters: This introduction is followed by Chapter 2 which presents the theoretical background and literature review of sentiment analysis and machine learning. Chapter 3 introduces the algorithms and principles that are used in this thesis. It also presents their strengths and limitations in the context of sentiment analysis. Chapter 4 illustrates the analytical methods and the steps that were applied to conduct sentiment analysis and obtain the results of this thesis. Chapter 5 explains the implementation details including data curation, data cleaning, pre-processing, sentiment prediction, and evaluation processes. Chapter 6 elaborates on the results obtained from the analysis conducted. The last chapter outlines the conclusions and summarizes the main findings and contributions of the research. Finally, recommendations for future works are provided.

# 2. BACKGROUND

The following chapter provides an overview of the current works and studies related to sentiment analysis. It provides a summary of the studies that have applied machine learning classifiers in the field.

## 2.1. SENTIMENT ANALYSIS OVERVIEW

We, as humans, unconsciously tend to categorize feelings and texts into positive and negative polarities. This division is usually based on evaluations of our perceptions that is affected by our emotions. Analysis of feelings has been a general practice even in ancient history. For instance, it is reported that "Greece's sovereign was gauging the fighting spirit of fighters by detecting intrinsic dissent that reflected emotionally loaded opinions" [23].

Sentiment analysis that is also known as emotional analysis, is a process of categorizing a piece of text to determine a person's attitude. The outcome can be summarized into categories such as positive, negative, and neutral or numerical scores to present attraction or acceptance towards a particular topic or product [24]. Sentiment analysis can be defined as a combination of alternative sequences to understand psychology and sociology concepts from a set of given natural language phrases [24].

The current advancements in information technology have provided unprecedented access for people to express their feelings and share their opinions on any product or topic via social media platforms. Users frequently express their feelings publicly by uploading voluminous raw data, including text, photos, video, and audio [24]. Extracting sentiments from these volumes of data provides great opportunities in many sectors. For instance, businesses are now able to understand how customers react to a new product and how to design new products.

Machine learning and deep learning provide the essential tools to achieve this goal by predicting opinion polarity [24]. Applying sentiment analysis techniques to the enormous amounts of online data from social media platforms can have an essential impact on society, including the modern economy [25]. That is especially effective since it has been reported that most consumers buy from the brands they follow, and consumers trust online reviews

more than sellers' statements [26]. Therefore, sentiment analysis has a great impact on the financial target as a powerful resource to facilitate service enhancement. Sentiment analysis improves the map between financial information posts on social media as a key to creating a successful link between brand and consumer [26].

## 2.2. MACHINE LEARNING METHODS FOR SENTIMENT ANALYSIS

Various machine learning algorithms have been applied for sentiment analysis. As examplified below, algorithms such as Naive Bayes, Gradient Boosting, K-Nearest Neighbors, and Logistic regression can be utilized to build the classifier model.

Shafiqi et al. [27] investigated clothes brand popularity on twitter. The work consisted of five stages, extraction of Tweeter data, labeling, pre-processing, sentiment classification, and evaluation. They used twitter-API to curate 24,000 tweets covering clothing brands Data tokenization, normalization, and stemming were implemented during the pre-processing stage. Data filtration was also applied to clean the data from stop words and outliers. They used *Word2Vec* to perform word embedding and applied *TextBlob* for data labeling. Afterwards, five different machine learning methods were applied, including support vector machine, naive bayes, random forest, multilayer perceprton, and logistic regression. The classifiers were evaluated using two splits. The first dataset included 70% training and 30% test data, whereas the second dataset was divided 50 to 50%. In the first case, SVM, and in the second case, logistic regression performed best, highest accuracy compared to the other classifiers. In both cases, the classifier accuracy was above 57%. This study provides the applicability of machine learning methods to predict brand popularity.

Chaudhry et al. [28] implemented a Naive Bayes classifier to investigate the US elections. The aim of the study was to investigate peoples' opinion before, during, and after the elections. They extracted Tweeter data including the location of the users. Data cleaning and preprocessing were performed by tokenization and removal of outliers and stop words. The tweets were clustered into five zones based on location for finding region-wise sentiments. For further preprocessing of the data, they used bigram and TF-IDF to detect sarcasm. To keep the dataset balanced, the tweets were limited per person and day. Furthermore, they

accounted for tweets that had correlated patterns to avoid inclusion of automated posts by Bots. Interestingly, these regorious preprocessing techniques helped the researchers to build a highly accurate model that accuracy of 94.5%, precision of 93%, F1 score 94.8%. This study demonstrate that performing a detailed preprocessing of the data is key to obtain a reliable classifier model. [29]

Windasari and Eridani[30]i applied an SVM classifier for analyzing tourist reviews on travel destinations. The reviews were extracted from Tripadvisor and they manually labeled a trainining dataset. After data preprocessing, they used N-gram and U-gram for weighting, followed by detection of essential terms in the sentences. The trained model had an accuracy of 85% both for positive and negative classes. This work provides a case study on sentiment analysis in the hospitality industry to improve tourist satisfaction.

Khan and Urola[31] examined consumer loyalty measurement for airline companies. They analyzed 10,000 tweets from 18 airlines belongin to four continents. Using a random forest classifier the classification accuracy reached 99.05% based on 10-fold cross-validation. The trained model can be a great use for airline companies to predict customer reactions before policy changes.

As a final example in using conventional machine learning methods for sentiment analysis, Padmanayana et al. [32] showcased that XGBoost classifier can be used to predict the stock market trend. They used historical stock data in addition to news headlines and twitter posts. After data cleansing, they trained a Naive Bayes classifier and then they fed the output to the XGBoost algorithm. The study predicted bullish and bearish market terms; where bullish refers to stock market rise, and bearish refers to lower prices in the stock market. The accuracy was 89.8%, which is higher than many other studies. The study shows XGBoost provides a more accurate model.

Recent developments in deepp learning architectures has enabled many studies to implement deep learning models for conducting sentiment analysis. Guner et al. [33], investigated product reviews on Amazon. They used a labeled dataset containing of four million reviews to train three classifiers: Naive bayesian, Support vector machines, and Long short-term memory (LSTM) that is a deep learning technique. LSTM had the highest performance (accuracy of 90%). Also, Bruyne et al. [34] delete this reference please

Hartmann et al. [35] performed an empirical assessment to analyze sentiments using conventional machine learning methods as well as transfer learning models. They gathered 217 publications that contained 1100 experimental results. They found that artificial neural networks had the highest performance, (85.75%) compared to Naïve Bayes, random forest, decision tree, logistic regression, and support vector machines.

## 2.3. OVERALL SUMMARY SECTION

The relevant literature that was presented in the background has paved the way for sentiment analysis using machine learning. However, to gain a better insight, below I provide some of their strengths and weaknesses to improve on.

The amount of labeled data that is used for training has a large effect on building a reliable model, particularly for sentiment analysis due to heterogeneity in text data. Shafiqi et al. has implemented an automated approach to label the data that has greatly helped in constructing the model. However, the amount of data that is used for training is very small that may have a negative impact on the results obtained. In comparison, Chadhry et al. have developed a model for scam detection, greatly focused on high-quality data by keeping the data balanced by filtering the input data to avoid duplicate tweets from the persons in a time-devised manner. They also filtered out tweets from Bots by using a correlation-based pattern. However, their model that used a Naive Bays classifier for the US election might need improved algorithms to receive accurate results.

Moreover, Windasari and Eridani's, implemented a feature extraction algorithm based on N-gram and U-gram for weighting detection of essential terms in the sentences. This method greatly improved the results by facilitating the training model. However, applying additional machine learning models such as deep learning beside their support vector machine model could provide enhanced results.

Khan and Urola, had an impressive accuracy of 99% which is higher than most studies in the same area. However, their lack of more detailed analysis to investigate the reasons for the negative points can be improved by downstream analysis.

Padmanayana et al, integrated different data sources from different structures including Twitter

posts and News headlines and that is a big advantage over other studies that only use one source of data. However, they only trained a naive bayes classifier therefore their results could be improved if they fed the original data in the first round and the supervised algorithm output in the second round to compare the result.

Finally, Guner et al, collected a huge dataset and implemented different algorithms which is a strength key. However, the study needs to extract features to be more powerful since Amazon data has many features. In contrast, Hartmann et al, used both artificial neural networks and supervised learning classifiers but the amount of data that was used for training was limited and that can be considered as a weakness of the study.

# 3. METHODOLOGY

Advances in artificial intelligence and machine learning methods provide great opportunities for conducting sentiment analysis on a large scale. To build the thesis framework, it is important for the reader to get an understanding of the relevant techniques and methods. Thus, this chapter introduces the methods and principles that are applied in this thesis to conduct sentiment analysis.

## 3.1. ARTIFICIAL INTELLIGENCE

Artificial Intelligence (AI) is developed by attempting to mimic the human biological neural network of the brain. AI processes start with collecting structured or unstructured data from the various data sources. The data are then transferred to data consolidation for extracting information. Finally, knowledge is obtained by implementing a series of algorithms to model a system. Moreover, AI architectures must satisfy several nonfunctional requirements including security of the data and computational resources (CPU, GPU, and TPU). Until recently, implementing AI systems were limited due to the low computational power that made it difficult to use it for large amounts of data [36]. However, recent developments in computational technologies has enable A.I. to have exponential growth in changing the fundamentals of data analysis (Scott, 2020, p.1). The availability of high-performance computing capabilities has greatly increased the abilities to get advantage of the large volumes of available data.

## 3.2. DATA MINING

Data mining is the process of discovering hidden patterns and relationships in large datasets through the use of techniques and machine learning algorithms [37]. Data mining aims to extract valuable insights and knowledge from large amounts of data to inform decision-making and drive business strategy. Data mining typically involves the following steps:

- Data cleaning: This step involves identifying and correcting errors, inconsistencies, and missing values, handling the inherent noise, and being uninformative in the dataset.

- Data integration: This step involves combining data from multiple sources into a single dataset.
- Data selection: This step involves selecting a subset of the data relevant to the analysis.
- Data transformation: This step involves converting the data into a more suitable form for analysis. This may include applying statistical techniques such as scaling or normalization.
- Data mining: This step involves applying algorithms and machine learning techniques to identify patterns and relationships in the data. Many powerful algorithms are available for data mining, including decision trees, clustering, and neural networks.
- Pattern evaluation: This step involves evaluating the discovered patterns and determining their significance. This may include using statistical tests or other methods to assess the reliability of the patterns.

Data mining is often used to improve decision-making by uncovering hidden patterns and trends in historical data. Data mining has become an increasingly important tool in various industries and fields, including business, finance, healthcare, and social media. The proliferation of data-generating technologies, such as sensors, mobile devices, and social media platforms, has resulted in an explosion of data that can be analyzed and mined for extracting insights. One example of data mining in the context of social media is the analysis of Twitter data. Twitter data can be mined to identify trends and patterns in user behavior, as well as to track the spread of ideas and opinions across the platform such that data mining could be used to identify the most popular hashtags or topics among Twitter users or to track the spread of a particular message or piece of news across the platform. Data mining could also be used to identify influential users or communities on Twitter or to understand how different groups of users engage with different types of content. Overall, data mining is a powerful tool that can extract valuable insights and knowledge from large datasets, with applications ranging from business strategy to social media analysis.

## 3.3. MACHINE LEARNING

Machine learning is a branch of artificial intelligence that is used to build computer models directly from data. Arthur Samuel, in 1959, defined machine learning as a "Field of study that allows computers to learn without being explicitly programmed" [38]. As explained in the figure 3.1, Machine learning can mainly be divided into two categories:

- **Supervised Learning**: refers to algorithms that try to learn from a labeled or classified dataset. The methods under this category usually require one training dataset and another test dataset. The model is implemented on the training dataset where the outcome is known, and the obtained model is evaluated based on its performance on the test dataset. Thus, supervised algorithms try to accurately map from $X$ to $Y$ where ( $X$ is independent-input, and $Y$ is the dependent label) in the given dataset. Furthermore, the key characteristics or patterns are tightly based on the input labeled data. For instance, in the case of sentiment analysis, the sentiments for a set of tweets should exist for the algorithms to use for training and building a model.
- **Unsupervised Learning**: refers to learning patterns or rules from a set of data points without any knowledge of the actual class or category of the input data. The algorithms in the category try to find some structure or pattern that can distinguish between the data points. Clustering methods are the main algorithms of unsupervised learning where data points in a dataset are grouped together based on the similarity between their features. For instance, in anomaly detection, fraud events in the financial system can be predicted by looking for unusual events. When the number of features is high, dimensionality reduction is applied to compress the dataset while losing as little information as possible

In this thesis, we have only applied supervised learning methods; therefore, below, some of the supervised methods are explained.

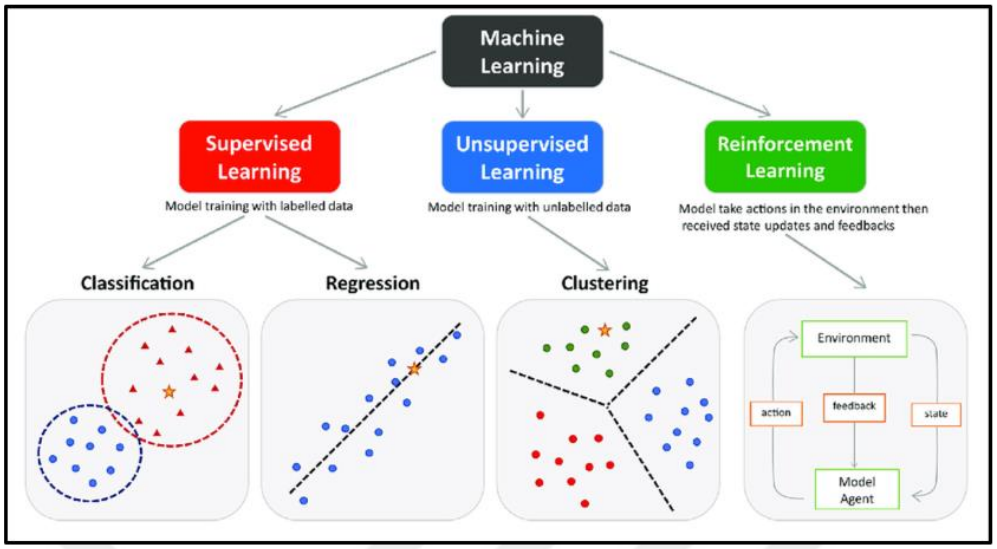


Figure 3.1. Types and examples of machine learning

### 3.3.1. Naive Bayes Classifier

A machine learning algorithm based on the Bayes' Theorem principle. This statistical theorem describes the probability of an event occurring given the likelihood of other related events. In the context of sentiment analysis, a Naive Bayes classifier can be used to predict the sentiment of a piece of text as positive, negative, or neutral; a simple graphical representation of the NB classifier for sentiment analysis is provided in the figure 3.2. The underlying theory behind a Naive Bayes classifier assumes that all of the features in the input data are independent of one another, which is why it is called "naive." In the case of sentiment analysis, the features might be individual words or n-grams (sequences of words) in the input text. Given this assumption of independence, the classifier can then use Bayes' Theorem to calculate the probability that a text has a certain sentiment based on the presence or absence of certain words or n-grams. One of the main benefits of using a Naive Bayes classifier for sentiment analysis is that it is relatively simple and easy to implement. Yet, it can still perform well on many tasks. It is also relatively fast to train and can scale to large datasets. However, there are also some challenges and concerns to remember when using a Naive Bayes classifier for sentiment analysis. One of the main limitations is that it assumes independence among the features, which may not always hold in practice. This can lead to less accurate predictions, particularly when the features are highly correlated. Additionally, a Naive Bayes classifier may be sensitive to

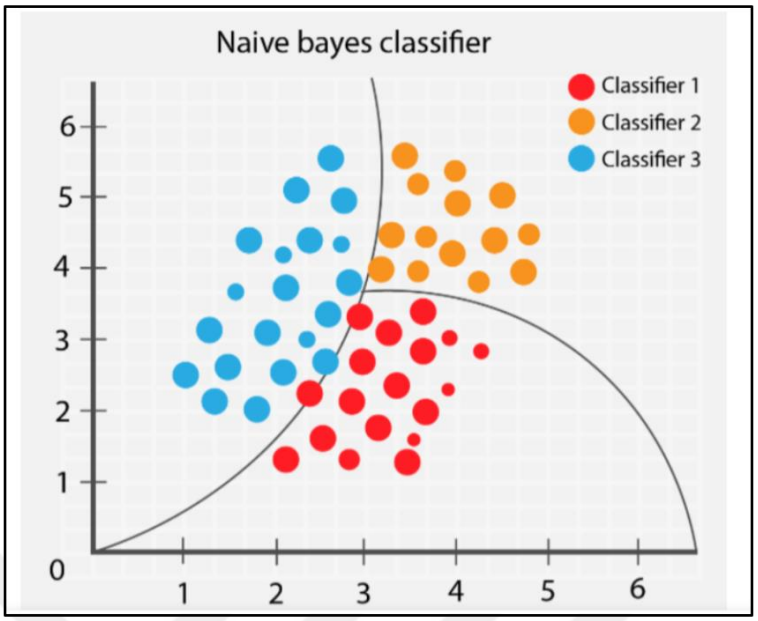


Figure 3.2. A graphical illustration of naive bayes classifier for sentiment analysis

irrelevant or noise features in the input data, which can negatively impact its performance.

In terms of the specific work procedure for using a Naive Bayes classifier for sentiment analysis, the process would involve the following steps:

- Collect and preprocess a dataset of labeled text for the sentiment analysis task. This may involve cleaning the text data and extracting relevant features such as individual words or n-grams.
- Train the classifier on the labeled dataset, using techniques such as cross-validation to evaluate its performance.
- Fine-tune the classifier by adjusting its parameters, such as the smoothing factor or the type of features used, to optimize its performance on the task.
- Use the trained and fine-tuned classifier to make predictions on new, unseen input text. The classifier accepts a piece of text and outputs a prediction of the sentiment. This can then be used for various purposes, such as sentiment analysis of social media posts or customer reviews.

Another benefit of using a Naive Bayes classifier for sentiment analysis is that it can handle large amounts of data efficiently due to its simple and fast training procedure. It is also

relatively robust to the presence of irrelevant or noisy features in the input data, as it only considers the presence or absence of features rather than their exact values.

There are also some ethical considerations to keep in mind when using a Naive Bayes classifier for sentiment analysis. One concern is the presence of potential biases in the training data that will have unwanted impacts on the classifier model used for predictions. For example, suppose the training data contains a disproportionate number of examples from a certain demographic or geographic region. In that case, the model may be biased towards predicting more common sentiments in that group. It is important to ensure that the training data is diverse and representative of various demographics and populations in order for the model to minimize the risk of bias. Overall, a Naive Bayes classifier is a simple yet effective tool for sentiment analysis based on the Bayes' Theorem principle and assumes independence among the input features. It can be used to predict the sentiment of a piece of text as positive, negative, or neutral and has the benefits of simplicity, speed, and robustness. However, it is important to consider the ethical implications of using a Naive Bayes classifier and to ensure that it is used responsibly and appropriately, taking into account any potential biases in the training data and the model's limitations. It is also important to be transparent about the model's limitations and to carefully evaluate its performance to ensure that it makes accurate and reliable predictions.

### 3.3.2. Logistic Regression

Logistic regression is a machine learning algorithm used to predict binary outcomes, such as positive or negative sentiment of a piece of text. It is based on using a linear function to predict the probability of an event occurring and then applying a threshold to determine the outcome. The underlying theory behind logistic regression is that it models the relationship between a set of input features (such as the words or n-grams in a piece of text) and the probability of a binary outcome (such as positive or negative sentiment). The model is trained using a labeled dataset of input features and binary outcomes, and the goal is to learn the weights or coefficients for each input feature that best predicts the outcome. Figure 3.3 shows a pictorial representation of Logistic Regression.

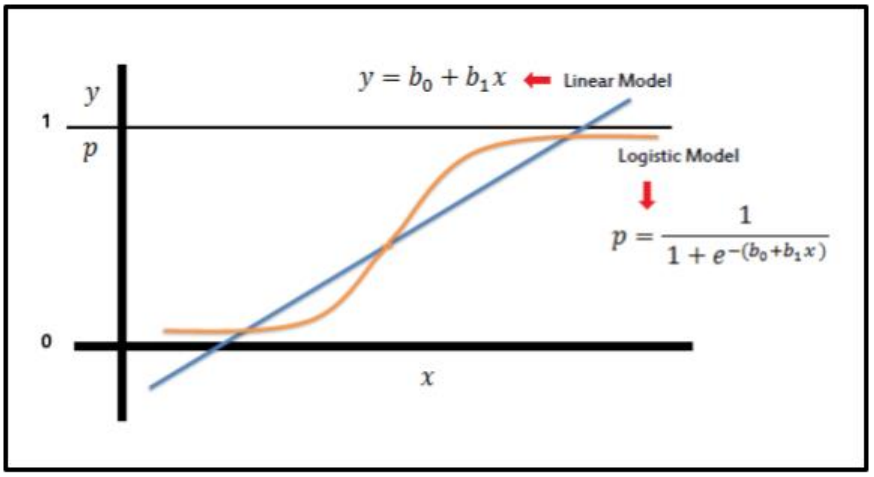


Figure 3.3. A graphical representation of logistic regression

One of the main benefits of using logistic regression for sentiment analysis is that it is relatively simple and easy to implement. Yet, it can still achieve good performance on many tasks. It is also relatively fast to train and can scale to large datasets. However, there are also some challenges and concerns to remember when using logistic regression for sentiment analysis. One of the main limitations is that it assumes a linear relationship between the input features and the outcome, which does not hold in practice especially in textual data used for sentiment analysis. This can lead to less accurate predictions, particularly when the connection is more complex. Additionally, logistic regression may be sensitive to irrelevant or noisy features in the input data, which can negatively impact its performance.

- In terms of the specific work procedure for using logistic regression for sentiment analysis, the process would involve the following steps:

- Collect and preprocess a dataset of labeled text for the sentiment analysis task. This may involve cleaning the text data and extracting relevant features such as individual words or n-grams.

- Train the model on the labeled dataset, using techniques such as cross-validation to evaluate its performance.

- Fine-tune the model by adjusting its parameters, such as the regularization strength or the type of features used, to optimize its performance on the task.

- Use the trained and fine-tuned model to predict new, unseen input text. The model

will take in a piece of text and output a prediction of the sentiment, which can then be used for various purposes, such as sentiment analysis of social media posts or customer reviews.

In summary, logistic regression is a simple and effective sentiment analysis tool based on the principle of using a linear function to predict the probability of a binary outcome. While it has some limitations, it can still achieve good performance on many tasks and is relatively easy to implement.

### 3.3.3. Gradient Boosting Classifier

Gradient Boosting is useful for both classification and regression problems. Gradient Boosting can be used for sentiment analysis in Twitter data by treating the task as a binary classification problem, where the tweets are classified as either positive or negative.

One benefit of using Gradient Boosting for sentiment analysis is that it can handle high-dimensional and sparse data, which is common in textual data. It can also handle missing values and outliers well. There are several tools available for implementing Gradient Boosting for sentiment analysis, including XGBoost and LightGBM. Some challenges and concerns to consider when using Gradient Boosting for sentiment analysis include ensuring that the data is sufficiently diverse and representative and that the model is not overfitting to the training data. It is also important to consider ethical considerations, such as the potential for biased model outcomes if the training data is not diverse or representative.

## 3.4. NEURAL NETWORKS

Neural networks in machine learning are inspired by the brain in both mission and structure. It involves using artificial neural networks that are algorithms designed to recognize patterns in data and make decisions.

Usually, neural networks consist of multiple layers of interconnected nodes, or "neurons," which process and transmit information. Each neuron receives inputs, processes them through a linear combination, and then applies a nonlinear activation function to produce an output.

The output of one neuron is typically passed to multiple neurons in the next layer, forming a directed graph of computations. The input layer of a neural network receives a set of feature vectors, typically numeric values representing some characteristics of an object or phenomenon being studied. These feature vectors are typically stored in a matrix $X$ with dimensions $(n, d)$, where $n$ is the number of samples and d is the number of features. The input layer then passes the feature vectors to the network's hidden layers, which are transformed through a series of computations known as "activations". Each hidden layer $j$ in the network is characterized by weight parameters $W^j$ and biases $b^j$. The activation $a^j$ at a given layer is computed as follows:

$$a^j = g(W^j a^{j-1} + b^j) \tag{3.1}$$

where $g$ is the activation function, $W^j$ is a matrix of weights with dimensions $(m^j, m^{j-1})$,

$a^{j-1}$ is the activation of the previous layer with dimensions $(m^{j-1}, n)$, and $b^j$ is a vector of biases with dimensions $(m^j, 1)$. The activation function $g$ can be any differentiable function, The structure of a brain neuron and artificial neuron is shown in figure 3.4.

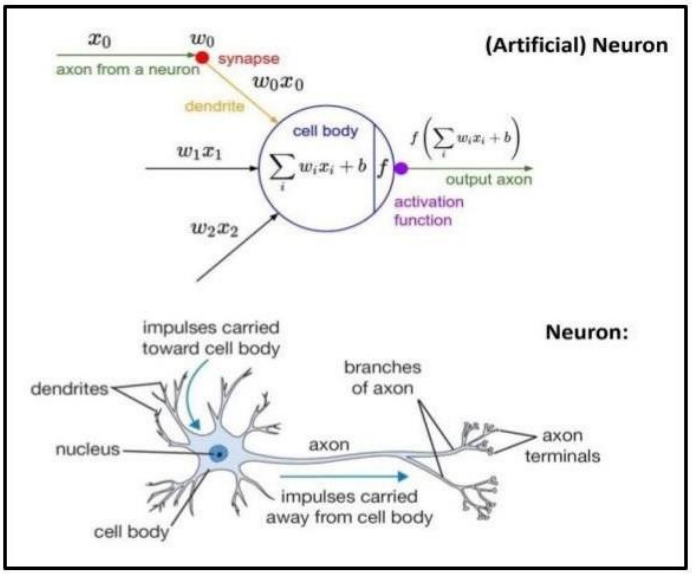


Figure 3.4. The structure of a brain neuron and artificial neuron

The output layer of the neural network is computed similarly, except that it produces the network's prediction or classification. For example, in a binary classification problem, the output layer might consist of a single neuron with a sigmoid activation function. In contrast, in a multi-class classification problem, the output layer might consist of multiple

neurons with a softmax activation function. The predicted class or value is then obtained by applying a suitable loss function $\mathcal{L}$ to the output activations and the true labels or values. The weights and biases of a neural network are typically learned from training data through an optimization process known as backpropagation. Given an initial set of parameters, the network is presented with a batch of training samples and makes predictions based on the current parameters. The loss is computed as a measure of the discrepancy between the predictions and the true labels. The gradients of the loss function for the parameters are then calculated using the chain rule and are used to update the parameters in the opposite direction of the gradient. This process is repeated for multiple epochs until the loss converges to a minimum or a maximum number of epochs is reached. In general, deeper and wider neural networks can learn more complex patterns in the data, but they also require more computational resources and are more prone to overfitting. Various regularization techniques can be applied to mitigate the risk of over-fitting, such as weight decay, dropout, and early stopping. Neural networks can also be trained in a distributed fashion using multiple GPUs or clusters, which allows for faster training and larger model capacity.

### 3.4.1. Deep Learning

Deep learning algorithms can be considered as complex neural network models that use more neurons, parameters and layers. they are appropriate for functions involving huge amounts of unstructured data, such as images, audio, and text. They can learn from the data without being explicitly programmed to perform the job, and they can continue to improve their performance as they are exposed to more data. Using deep learning networks in machine learning. However, it wasn't until the development of more advanced computing technologies in the late 20th and early 21st centuries that deep learning became practical. The rise of big data and the availability of powerful graphics processing units (GPUs) enabled the training of much larger and more complex neural networks, leading to significant advances in the field.

Designing of multiple neural network layers has given rise to many deep learning architectures. The layers in these architectures can be divided into three types:

- Input Layer: The input data is transformed into vectors and matrices to be able to apply

matrix operations.

- Learning layer: this may consist of many interconnected layers. The goal is to extract information from the data in multiple dimensions. These layers are hidden in the model since they are applied between the input and the output layers.

- Output Layer or classification layer: the results are transformed to obtain a neuron for each class that represents the probability of the outcome.

One example of deep learning in social media is Twitter data analysis; deep learning algorithms could be used to classify tweets by sentiment (positive, negative, neutral) or to identify the most popular hashtags or topics among Twitter users. Today, deep learning is used in many applications, including natural language processing, image and video analysis, and medical diagnosis. It has also been used to achieve state-of-the-art results in many other fields, including game playing, drug discovery, and financial forecasting.

## 3.5. NATURAL LANGUAGE PROCESSING

Natural language processing (NLP) is the intersection of artificial intelligence and linguistics to achieve interaction between computers and human (natural) languages. The goal of NLP is to develop algorithms and systems that can understand and interpret human language in a useful and meaningful way. The study of NLP can be traced back to the 1960s, with the development of early natural languages processing systems such as ELIZA, which was capable of basic conversation and pattern recognition. Since then, NLP has evolved significantly, driven by advances in machine learning, linguistics, and computer science. NLP has many applications, including language translation, text summarization, chatbot development, and sentiment analysis. To perform these tasks, NLP algorithms must be able to analyze and understand the structure and meaning of human language, which is a complex and nuanced task. Many techniques and approaches are used in NLP, including rule-based systems, machine learning, and deep learning. Rule-based systems rely on explicit rules and dictionaries to process language. In contrast, machine learning approaches involve using algorithms that can learn from examples and improve their performance over time. Whereas deep learning approaches involve using artificial neural networks to analyze and interconnect

different parts of a language based on text data. One of the key challenges in NLP is the diversity and complexity of human languages. Different languages have their own unique grammatical and syntactical rules, as well as their vocabularies and idioms. In addition, the meaning of words and phrases can depend on their context and the intentions of the speaker or writer. As a result, NLP algorithms must be able to handle a wide range of variations and ambiguities to be effective. Also, large amounts of data are needed so that the algorithms are able to take a wider range of variations into consideration.

For NLP tasks, machine learning algorithms must be able to perform several tasks, including:

- Tokenization: splitting the text into individual words and symbols
- Part-of-speech tagging: identifying the roles that words play in a sentence
- Parsing: analyzing the grammatical structure of a sentence
- Semantics: determining the meaning of words and phrases in the context.

These tasks can be challenging due to the complexity of natural language and the many variations and exceptions that exist. Despite these challenges, NLP has made significant progress in recent years, thanks in part to the availability of large amounts of annotated text data and the development of more powerful machine learning techniques such as deep learning. NLP algorithms are now used in various applications, including language translation, information retrieval, and sentiment analysis.

## 3.6. SENTIMENT ANALYSIS METHODS

Sentiment analysis is one of the NLP tasks that aims to identify the opinion or polarity based on the words in a text data. Multiple deep learning models have been build by training complex neural network architectures on data from social media platforms, blogs, and review forums. Due to the wide scopes of their training data, some of these models are generalizable to conduct sentiment analysis in several contexts such as business products, movies, retail popularity, etc.

The next section provides an overview of the main models that have been built for sentiment

analysis in English language.

### 3.6.1. Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers (BERT), is a deep learning model developed by Google for natural language processing tasks, mainly sentiment analysis. BERT is based on the transformer architecture introduced in the paper "Attention is All You Need" by Vaswani et al. in 2017. One of the key features of BERT is its ability to process input text in both directions (bi-directional). This contrasts traditional language models, which typically only process text in a left-to-right or right-to-left direction. By considering the context of words in both directions, BERT can capture more subtle and nuanced meanings in the input text. BERT is trained on a large dataset of unlabeled text and is then fine-tuned on a specific task, such as sentiment analysis. To perform sentiment analysis, BERT takes text as input, and outputs a prediction of whether the sentiment is positive, negative, or neutral. Figure 3.5 shows the architecture of BERT.

One of the main challenges in using BERT for sentiment analysis is the need for large amounts of labeled training data. It can be time-consuming and expensive to label large datasets manually, and the quality of the model will depend on the quality of the labeled data. Additionally, BERT and other deep learning models can be computationally intensive, requiring specialized hardware and infrastructure to train and deploy. Despite these challenges, BERT has several benefits for sentiment analysis. It can capture complex and nuanced meanings in text, which can be difficult for traditional rule-based approaches. BERT is also relatively easy to fine-tune for specific tasks and can achieve state-of-the-art performance on a wide range of natural language processing tasks. There are also some ethical considerations to keep in mind when using BERT for sentiment analysis. As with any machine learning model, it is important to ensure that the model is fair and unbiased and to consider the potential impacts of the model on individuals and society. It is also important to be transparent about the model's limitations and to ensure that it is used appropriately and responsibly.

BERT is based on the transformer architecture, which uses self-attention mechanisms to

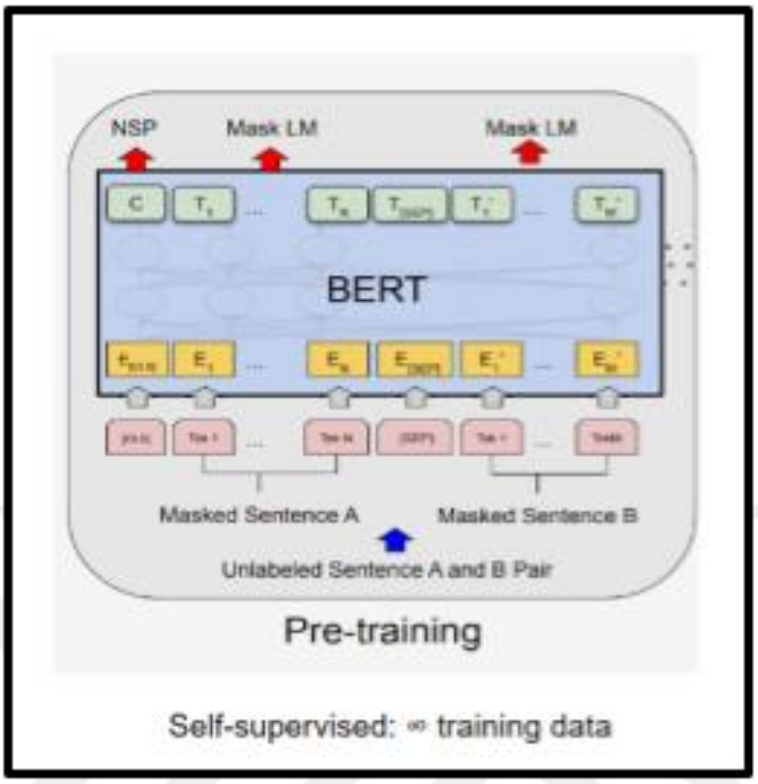


Figure 3.5. Architecture of BERT

process input text. In self-attention, the model considers the relationships between all pairs of input tokens (e.g., words and subwords) at each position in the input sequence. This is performed using the following equation:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3.2)$$

Where $Q$, $K$, and $V$ are the query, key, and value matrices, respectively, and $d_k$ is the dimensionality of the keys. The attention function returns a weighted sum of the values, with the weights determined by the dot product of the query and key matrices normalized by the square root of the key dimensionality. The Architecture of the transformer is provided in figure 3.6. BERT uses multiple transformer layers in its architecture, and each layer processes the input text using self-attention and feedforward neural network layers. The output of each layer is passed to the next layer, and the final output of the model is obtained by applying a linear transformation and a $softmax$ function to the output of the last layer. BERT uses a variant of the transformer architecture called the Transformer-$XL$, which can process input

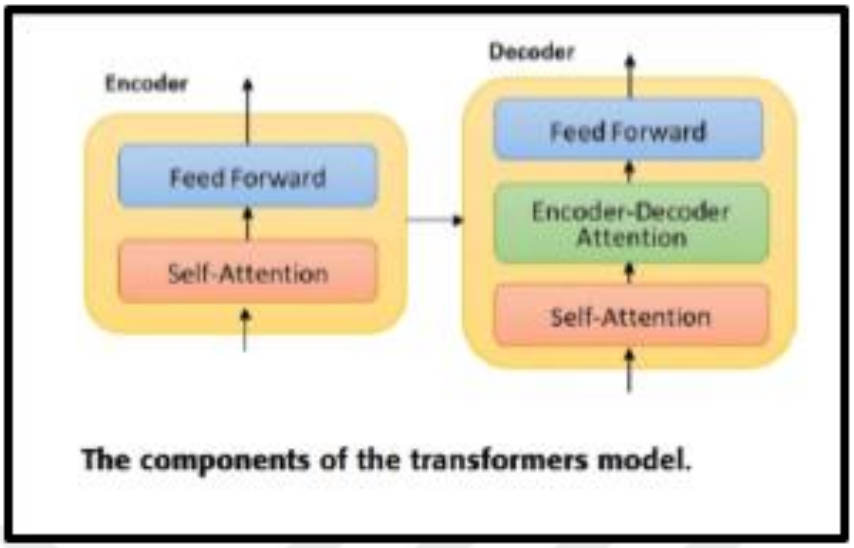


Figure 3.6. Architecture of the transformer

sequences longer than the standard transformer model. This is done using a "segment-level recurrence technique,", which allows the model to remember information from previous segments of the input sequence and use it to process the current segment. BERT includes several different pre-trained models that can be fine-tuned for specific tasks, such as sentiment analysis. The most commonly used models are BERT-Base and BERT-Large, which have 12 and 24 transformer layers, respectively.

### 3.6.2. Robustly Optimized BERT Approach (RoBERTa)

Robustly Optimized BERT Approach (RoBERTa), is a variant of the BERT model developed by Facebook AI that is designed to improve the performance of BERT on a wide range of natural language processing tasks, including sentiment analysis. Similar to BERT, RoBERTa is based on a transformer architecture and uses self-attention mechanisms to process input text. However, RoBERTa makes several improvements to the original BERT model to achieve better performance. These improvements include:

- **Training on a larger dataset**: RoBERTa is trained on a larger dataset. This allows RoBERTa to learn more robust and generalizable representations of language.
- **Removing the next sentence prediction task**: BERT includes a next sentence

prediction task as part of its training procedure, which is intended to encourage the model to capture contextual relationships between sentences. However, this task is less effective at improving performance on downstream tasks. RoBERTa removes the next sentence prediction task and instead focuses solely on the masked language modeling task, which involves predicting the masked words in a sentence given the context of the other words.

- **Modifying the training schedule**: RoBERTa uses a longer training schedule than BERT, with more training steps and larger batch size.
- **Using dynamic masking**: In the masked language modeling task, RoBERTa uses dynamic masking, which involves randomly sampling different masking patterns for each training example. This is in contrast to BERT, which uses a fixed masking pattern. Dynamic masking has been found to improve the robustness and generalization of the model.

RoBERTa has performed strongly on various natural language processing tasks, including sentiment analysis. It is particularly useful for tasks that require a high level of understanding of the underlying context and meaning of the input text. Regarding the specific work procedure for using RoBERTa for sentiment analysis, the process is similar to that of using BERT or any other machine learning model. First, the model would need to be trained on a large dataset of labeled text, to predict the text's sentiment as positive, negative, or neutral. This can be done using the pre-trained RoBERTa model or training a custom model from scratch. However, since training a custom model requires huge computational powers it is very common to use the RoBERTa model to predict sentiments after processing the input data. Also, in cases of re-training the model for a specific dataset, it would need to be fine-tuned for the sentiment analysis task. This involves adjusting the model's parameters to optimize its performance on the task. This can be done using techniques such as hyperparameter optimization, which involves adjusting the model's hyperparameters (e.g., learning rate, batch size, etc.) to find the best combination for the task. Once the model has been trained and fine-tuned, it can predict new, unseen input text. The model will take in a piece of text and output a prediction of the sentiment, which can then be used for various purposes, such as sentiment analysis of social media posts or customer reviews. Figure 3.7 shows the architecture of RoBERTa. In

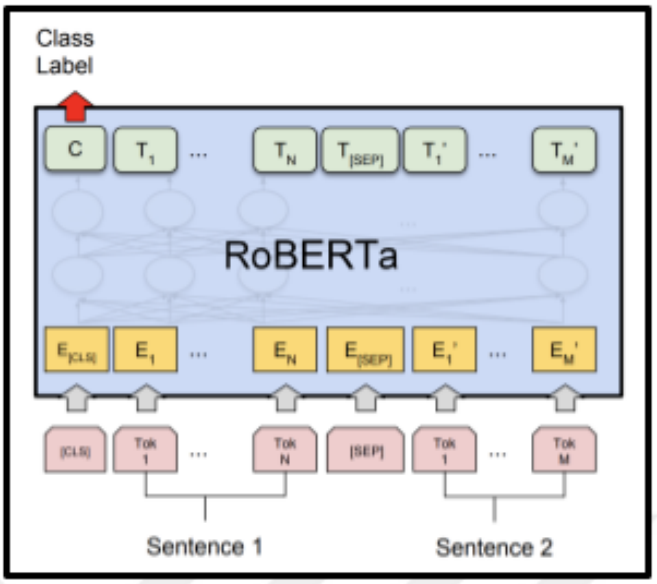


Figure 3.7. Architecture of RoBERTa

summary, RoBERTa is a powerful sentiment analysis tool based on transformer architecture. It has been specifically designed to improve the performance of BERT on a wide range of natural language processing tasks. While there are challenges and ethical considerations to keep in mind when using RoBERTa, it has the potential to provide accurate and robust predictions of sentiment in a variety of contexts.

## 3.7. EVALUATION METRICS

The aim of this task is to classify text into positive, negative, or neutral categories. Sentiment analysis has applications in various fields such as marketing, customer service, and social media analysis. In this project, we are using Logistic Regression, XGBoost, Naive Byes, Neural Networks, BERT, and RoBERTa for sentiment analysis of the IKEA Furniture store Reviews Dataset. Evaluation metrics are an essential aspect of any machine learning project. These metrics help us to evaluate the performance of the models and provide insights into their strengths and weaknesses. In this section, we will discuss the evaluation metrics that will be used for this project. We will focus on four common evaluation metrics - accuracy, F1-score, precision, and recall.

### 3.7.1. Accuracy

Accuracy popular evaluation metrics in machine learning. It measures the percentage of correctly classified instances in the test set. It is defined as the ratio of the number of correctly classified instances to the total number of instances in the test set.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.3}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives. Accuracy is a useful metric when the classes are balanced. However, when the classes are imbalanced, accuracy can be misleading. For example, in a binary classification problem with 90% of the instances belonging to the positive class, a model that always predicts the positive class will have an accuracy of 90%. Therefore, accuracy alone is not sufficient to evaluate the performance of the models.

### 3.7.2. F1-score

The F1-score is a harmonic mean of precision and recall. It provides a balance between precision and recall, which is useful when the classes are imbalanced. F1-score is defined as:

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} \tag{3.4}$$

where precision is the ratio of true positives to the total number of instances predicted as positive, and recall is the ratio of true positives to the total number of instances that are actually positive. The F1-score ranges from 0 to 1, with 1 being the best score. The F1-score is a useful metric when the classes are imbalanced, as it takes into account both precision and recall.

### 3.7.3. Precision

Precision measures the proportion of true positives to the total number of instances that are predicted as positive. Precision is defined as:

$$Precision = \frac{TP}{TP + FP} \tag{3.5}$$

Precision is a useful metric when we want to minimize false positives. For example, in a spam detection problem, we want to minimize the number of legitimate emails that are classified as spam. Therefore, we want to maximize precision.

### 3.7.4. Recall

Recall measures the proportion of true positives to the total number of instances that are actually positive. Recall is defined as:

$$Recall = \frac{TP}{TP + FN} \tag{3.6}$$

The recall is a useful metric when we want to minimize false negatives. For example, in a disease diagnosis problem, we want to minimize the number of cases where the disease is present but not detected. Therefore, we want to maximize recall.

### 3.7.5. Conclusion

section have discussed the evaluation metrics that will be used in our sentiment analysis project. We will use accuracy, F1-score, precision, and recall to evaluate the performance of our models in the section by discussing how these evaluation metrics will be used to compare the performance of the different models. To compare the performance of the different models, we will compute the evaluation metrics on the test set for each model. We will use the same test set for all models to ensure a fair comparison. Accuracy will provide an overall measure of how well each model performs in terms of correctly classifying positive and negative reviews. However, as previously mentioned, accuracy can be misleading when the classes

are imbalanced. Therefore, we will also compute the F1-score, which provides a balanced measure of precision and recall. Precision and recall will be useful in identifying the strengths and weaknesses of each model. For example, a model with high precision but low recall may be better suited for tasks where minimizing false positives is important. On the other hand, a model with high recall but low precision may be better suited for tasks where minimizing false negatives is important.

In addition to these metrics, we may also consider other metrics such as the area under the ROC curve (AUC-ROC) or the area under the precision-recall curve (AUC-PR). These metrics provide a measure of the trade-off between true positive rate and false positive rate or precision and recall, respectively. In summary, we will use accuracy, F1-score, precision, and recall to evaluate the performance of our sentiment analysis models. These metrics will provide a comprehensive measure of how well each model performs in terms of correctly classifying positive and negative reviews, and will help us identify the strengths and weaknesses of each model.

# 4. ANALYSIS AND DESIGN

The system has many requirements. System requirements are classified into two types: functional and non-functional requirements. Each of the functional requirements in the programming stage has been mapped into different diagrams to figure out the most comprehensive picture. A flowchart is provided in the design section.

## 4.1. FUNCTIONAL REQUIREMENTS

Multiple Python packages are required to perform this analysis. First, the Snscrape package is needed to extract Tweets from the Twitter API platform for sentiment analysis. Next, data pre-processing or data cleaning require additional Python libraries including nltk, stopwords, and CountVectorizer. Tokenizer converts tweet texts into numbers to pass to the model. (Attention-mask)-Encoder converts to PyTorch tensors, then it can pass that to the model. The system must classify tweets to polarize them into positive, negative, or neutral sentiments using machine learning classifiers from the deep learning models therefore models such as Roberta Model BERTA Model are required for this purpose.

Another functional requirement is using supervised machine learning to find sentiment polarity, including Gradient boosting (GB), Stochastic Gradient Descent (SGD), K-nearest neighbors (KNN), and Naive Bayes classifier for IKEA tweet sentiment analysis purposes.

The system must be able to compare the identified features with other sentiment analysis methods and provide a visualization or tabular representation of the results. After sentiment analysis identifies positive, negative, and neutral polarity, it extracts and analyzes the bi-gram and tri-gram features of the dataset by identifying combinations of words thatoften occur together benefiting from BigramAssocMeasures, BigramCollocationFinder, TrigramCollocationFinder,TrigramAssocMeasures.

Most associated with different sentiments by implementing a statistical analysis or feature selection technique. Then, evaluates the result accurately and comprehensively and measures the differences between utilized algorithms. Finally, the system must offer a comprehensive and detailed analysis of consumer sentiment toward IKEA through multiple sentiment analysis

methods and the identification of specific features of the dataset mostly associated with different sentiments. It requires the system to have the capability to generate a report or presentation of the findings.

## 4.2. NON-FUNCTIONAL REQUIREMENTS

Accuracy, security, availability, and usability are primary non-functional requirements in all software projects. The system must have a high level of accuracy in classifying tweets' sentiment polarity and to handle many potential variations in language, slang, noise, or irrelevant data in the tweets without errors. The system must be secure and protect data privacy. The data of tweets requires the system to have security measures such as encryption and access controls. Efficient algorithms and processing capabilities are required to process dataset analysis (60,000 tweets) within a limited time. In addition, the system must be available without any difficulties and be user-friendly and easy for people from different academic backgrounds. A simple user interface is required to run all functional sections correctly. The flexible system must handle large amounts of data and perform well under stress or heavy loads. For that purpose, sufficient memory and processing power are required. The system must also handle potential noise and irrelevant data in the tweets, such as links or hashtags. It requires the system to have the ability to export data or integrate with other systems as necessary. Finally, some of the soft parts in some sections of this thesis can be used by different researchers to save time and avoid repeated work.

## 4.3. RESEARCH SCOPE

The research evaluates the performance of various sentiment analysis methods on a dataset of 60,000 tweets toward IKEA and identifies the bi-gram and tri-gram features of the dataset that are most associated with different sentiments. The tested methods include Google Berta, Roberta, Logistic Regression, Naive Bayes, XGboost, and KNN. The results will be compared based on the tweet's classification polarity. The research covers the following aspects:

1. Compare the performance of multiple sentiment analysis methods on a dataset of 60,000 tweets toward IKEA and providing an understanding of the strengths and limitations

of each approach in this context.

2. Extract and analyze IKEA dataset's bi-gram and tri-gram features and identifying those most associated with different sentiments.

However, the research does not cover Sentiment analysis in languages other than English. Analysis of images or videos, and identifying individual entities mentioned in the tweets. In terms of Data Preprocessing, specific steps will be taken to clean and pre-process the dataset, which includes removing duplicate tweets, and handling missing data, emoticons, and emojis. It also includes removing irrelevant tweets, non-English tweets, links, and hashtags.
The evaluation metric used in the research will be the accuracy of classification as positive, negative, or neutral sentiment.

## 4.4. RESEARCH QUESTIONS

1. To do different sentiment analysis methods on Ikea's dataset of 60,000 tweets, and what are the strengths and limitations of each approach in this context?

2. What are the bi-gram and tri-gram features toward IKEA's dataset most associated with? Is it associated with the positive, negative, or neutral sentiment, and how do these features differ from different sentiment analysis methods?

## 4.5. SYSTEM MODEL

This thesis focuses on a graphical language and uses flowcharts to represent a workflow toward IKEA's sentiment analysis. In this step, the Snscrap package is required with Python as an input of the process. During the computational process, the tweets convert to a data frame to combine all tweets into one data structure for easy processing and it will be saved as a CSV file which contains stored Data type: object (tweet-Date, tweet-User, tweet-Contain, tweet-URL, tweet-Lang, and tweet-Location) as explained in figure (4.1). 4.1. Roberta Model is used for analysis purposes and IKEA's tweet dataset is used that was extracted previously. Implement AutoTokenizer, and ENcoder. Roberta's result in the dataset. During the computational

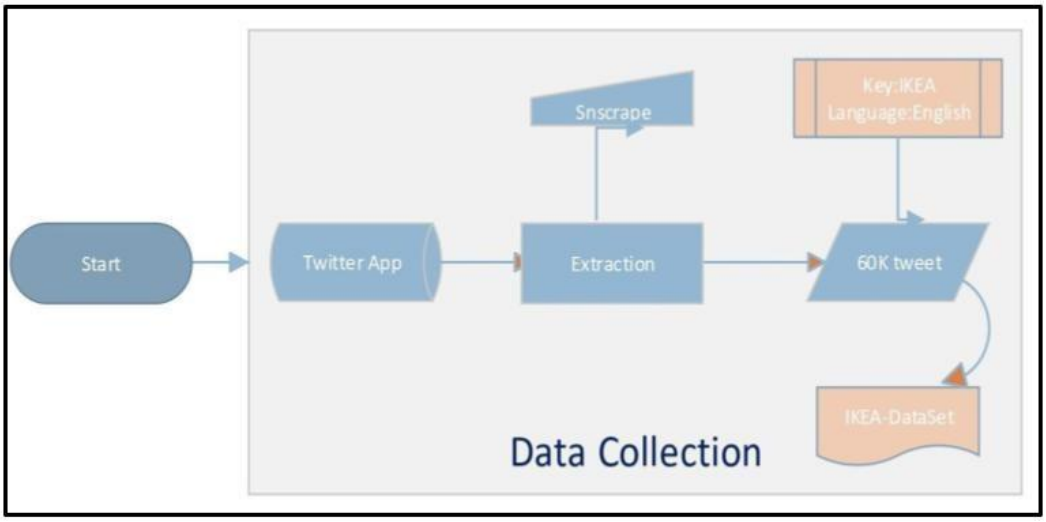


Figure 4.1. Data collection through extracting online tweet

process, all tweets are analyzed to (positive, neutral, and negative polarity and score). It reads IKEA's tweet dataset in a data frame shape. Tokenizer converts tweet texts into numbers to pass the model. The attention-mask-Encoder converts to PyTorch tensors, then it can pass that to the model which is explained in figure 4.2 BERTA-Model is used for analysis purposes

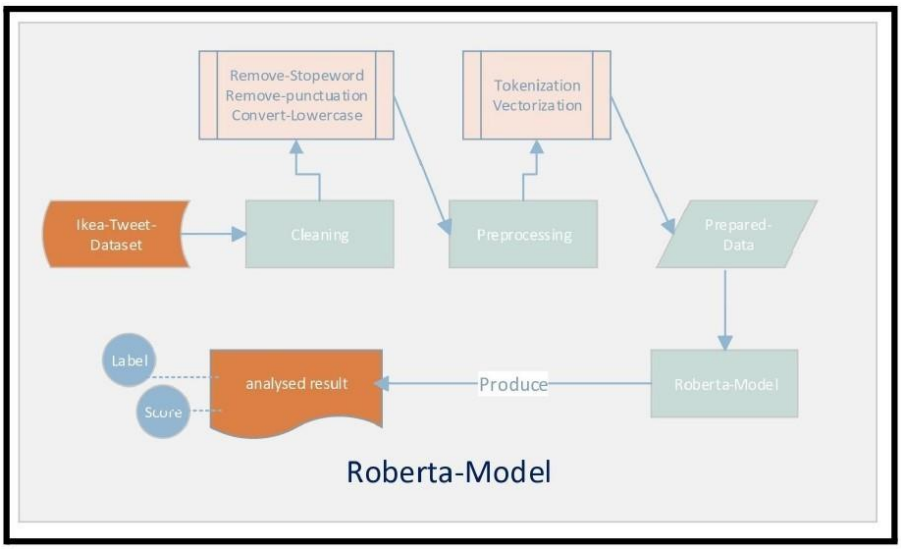


Figure 4.2. Extract sentiment analysis polarity (positive, neutral, and negative) for each tweet via using RoBERTa model

and IKEA's tweet dataset is used that was extracted previously. Implement AutoTokenizer

and ENcoder. Roberta's result in the dataset. All tweets are analyzed to (positive, neutral, and negative polarity and score) during the Computational process. Read IKEA's tweet dataset in data frame shape; Tokenizer converts tweet text to numbers to pass the model. (Attention-mask)-Encoder converts to PyTorch tensors, then it can pass that to the model, as explained in figure 4.3 The system finds the accuracy of both Roberta and Berta models. It

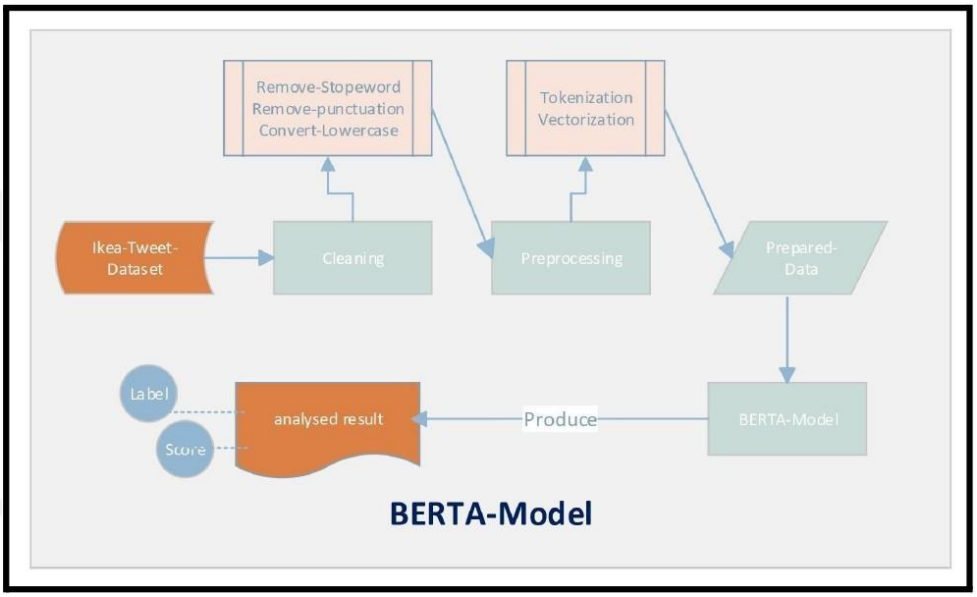


Figure 4.3. Extract sentiment analysis polarity (positive, neutral, and negative) for each tweet via using BERTA model

has IKEA's tweet dataset-manual, ROberta-output-dataset, and BERTA-output-dataset to find Roberta and Berta models accuracy, F1, Precision, Recall, Mean Squared Error, and Mean Absolute Error. Through the Computational process: compare the Manual-tweet dataset with the Predicted dataset, and calculate MSE and MAE as described in figure 4.4. The system will implement different supervised machine learning classifiers including (K-nearest neighbors (KNN), Gradient boosting (GB), Stochastic Gradient Descent (SGD), and Naive Bayes classifier for IKEA tweet sentiment analysis purpose. Utilize labeled-IKEAtweetdataset as input. Implement MultinomialNB to split the train-test tweets. Batch the tweet- while the algorithm cannot read all at once due to memory issues. and cleaning data including(removing Punctuation, removing stop words converting to lower case, vectorization with fit transformer, Implementing AutoTokenizer, calculating MSE and MAE, and calculating confusion matrix.

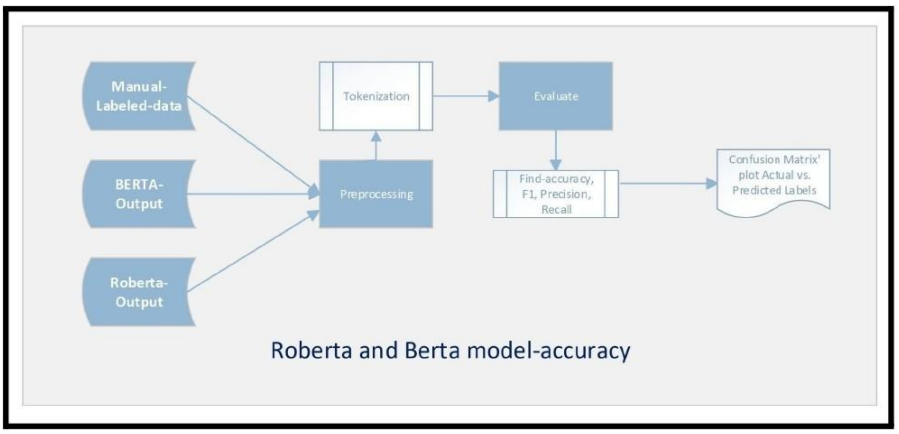


Figure 4.4. Evaluate both RoBERTa and BERTA accuracy

Under The condition: dataset should be in CSV file type. Implement MultinomialNB to split the train-test tweets. Batch the tweet- while the algorithm cannot read all at once due to memory issues. As the result will Produce output: analyze the tweet to (positive, neutral, and negative polarity), find model accuracy, F1, Precision, and Recall. Print the classification report, and visualize the confusion matrix, which is explained in figure 4.5. The system

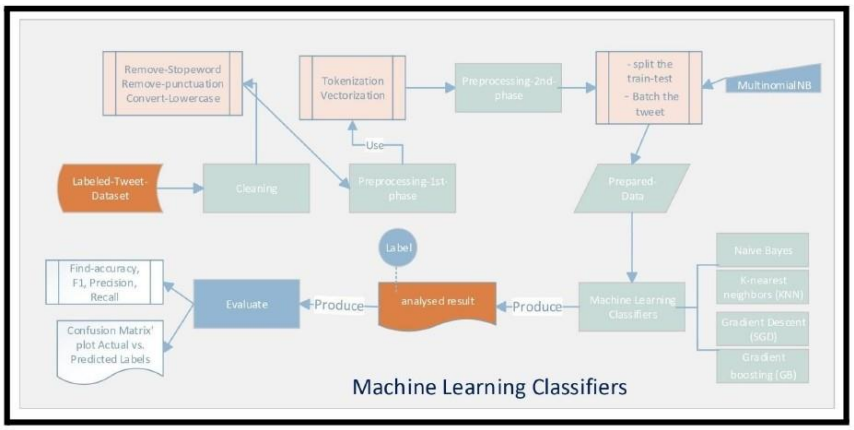


Figure 4.5. Extract sentiment analysis polarity (positive, neutral, and negative) for each tweet using supervised machine learning

will extract features for analyzing tweets via using analyzing the IKEAtweetdataset sample. That Dataset polarity has been analyzed. Then, Implement AutoTokenizer, implement vectorization. Apply Bi-gram assocMeasuer, and cleaning process to produce the top 20 Bi-gram for each polarity( positive, neutral, and negative), graph the top 20 Tri-gram for each

polarity( positive, neutral, and negative), print word cloud for each ( positive, neutral, and negative), as explained in figure 4.6. Lastly, the summary of the previous overall steps starts

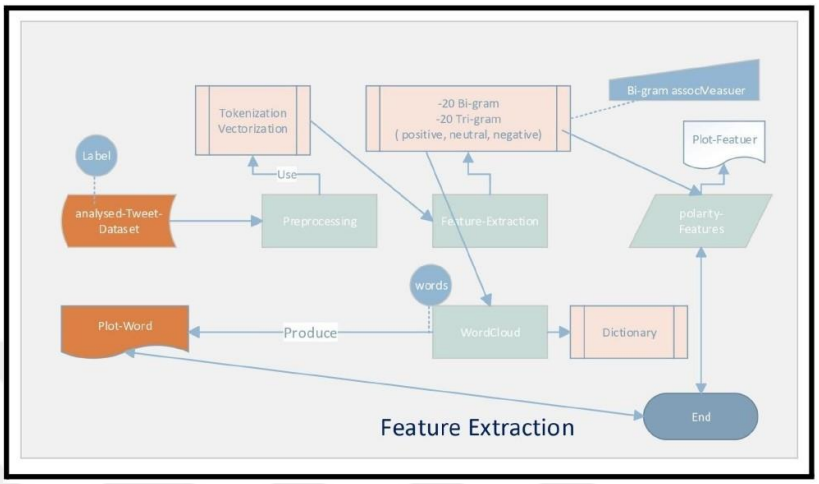


Figure 4.6. Extracting sentiment feature for each polarity (positive, neutral, and negative)

with extracting data from Twitter and cleaning it from outliers. After that, data preprocessing is a repetitive step in all sections, which includes tokenization and vectorization. Roberta and Berta's model uses extracted data after cleaning and preprocessing; furthermore, evaluating both algorithms through manually labeled data to find accuracy. At the same time, machine learning classifiers use their output which is labeled data, and find sentiment polarity (positive, neutral, and negative). Finally, through the feature extraction step, Apply Bi-gram, and Tri-gram for each polarity (positive, neutral, and negative), then find a word cloud for each (positive, neutral, and negative), this is elaborated in figure 4.7.

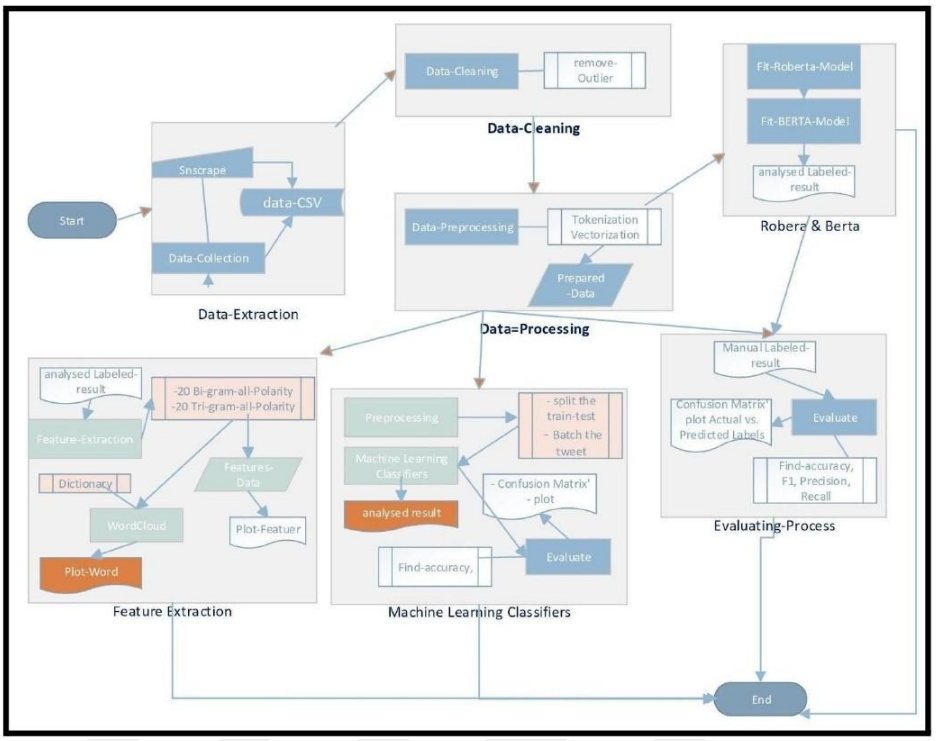


Figure 4.7. Overall sentiment analysis system

# 5. IMPLEMENTATION

This chapter provides a detailed description of the implementation of the study. Includes data collection, data cleaning, preprocessing, training machine learning (ML) models, plotting n-gram features of IKEA tweets, and evaluating the performance of BERT and RoBERTa models. Data collection involves collecting tweets related to IKEA. After collecting the tweets, cleaning and preprocessing data start to remove irrelevant or duplicate information. The preprocessed data is then used to train several ML models, including n-gram features, to plot the most frequently occurring word combinations in the tweets. Finally, the chapter evaluates BERT and RoBERTa models, which are state-of-the-art natural language processing models, in terms of their performance on the IKEA tweet dataset. Overall, this chapter provides a comprehensive understanding of the implementation of the study and the techniques used to analyze the collected data.

## 5.1. DATA COLLECTION

Python implements the data collection and preprocessing stage(s). In addition, the following modules: Pandas, CSV, and OS are imported. Panda and CSV handle and manipulate the data while the OS interacts with the operating system. In order to scrape tweets from Twitter, utilize the snscrape module. This module is a package without API scraping from the social network. For the first time, the pip package installer is used to install the latest version of snscrape. The specific module **snscrape.modules.twitter** imports as sntwitter implements for Twitter scraping. It extracts 60,000 tweets from Twitter that are related to mentions of "IKEA." This has been achieved by searching for tweets that includes the keyword "IKEA" in the text.

The code is designed only to include tweets written in English. It discards tweets that have been written in other languages to ensure that the dataset is homogeneous and accurate for sentiment analysis. This includes tweets from the first quarter of 2020 to the last quarter of 2022. After data collection, a preprocessing stage applies to clean data and prepare it for analysis. This stage includes the removal of duplicate tweets and the discarding of empty values, emojis, URLs, and personal information in the tweets. It is also important to ensure

the data is accurate and does not expose any sensitive information. The tweets that are irrelevant to the research topic, such as IKEA stores' locations or a specific IKEA product, are also discarded. The preprocessing steps are implemented to ensure that the data only contains meaningful, valuable, and useful information for sentiment analysis. Furthermore, the data collection and preprocessing stage lays the foundation for the subsequent analysis of sentiments towards IKEA on social media and is the first step toward the study's goal. Finally, the sentiment analysis process cleans the data and is carried out using suitable techniques and models.

## 5.2. DATA PREPROCESSING

The preprocessing stage is implemented by iterating over each row of the test dataset "**test-tweets-df**" using the "**iterrows()**" method. It is provided by the Pandas library. For each row, the tweet text and its actual label extract and assign to the corresponding column values of the current row. The tweet text is then tokenized into individual words and processed through various conditions to filter out irrelevant information.

The first condition that applies is the removal of mentions of other Twitter users from the tweets. The words that start with the "@" symbol and are longer than one character will be replaced with a blank space. This step is essential as it eliminates information that is not relevant for sentiment analysis.

The second condition that applies is removing any links or URLs from the tweets. The words that start with "HTTP" will be replaced with a blank space. This is an essential step as it eliminates information that could bring bias to the analysis. It also improves the accuracy of the results.

Other steps such as removing emojis, hashtags, and personal information are subject to data cleaning. To ensure the data is accurate and is not exposing any sensitive information.

The code also implements the removal of stop words and lemmatization as an additional step. Stop words are common words that do not add significant meaning to the text, like "the," "a," etc. the removal of stop words will be implemented by using the "string" and "stopwords" libraries of nltk. The "string. punctuation" is a predefined string in the string library in python; it contains all punctuation characters.

The function tweet_clean(text) is defined and takes the tweet text as input. First, it removes the punctuation from the text by iterating over each character of the text and checking if it is a punctuation mark or not. The characters that are not punctuation marks will be added to a new list, then joins to form a string free of punctuation marks. The next step is removing the stopwords from this string, which is done by iterating over each word in the string and checking if the word is present in the list of English stopwords. If the word is present, it is not added to the final list of words instead it will be added to the final list. After all these steps, the cleaned and preprocessed data is ready for the following steps in sentiment analysis. Text preprocessing is essential in preparing the data for sentiment analysis and improves the results' accuracy.

## 5.3. EXTRACT FEATURES

Extracting the bi-grams and trigrams of the preprocessed tweets is a vital step in sentiment analysis. This step identifies the words and phrases in the tweets that occur frequently. The words and phrases that are indicative of a particular sentiment. The extracted n-grams were then used to create visualizations that provide insights into the sentiment of the tweets.

The first step in this process is to group the tweets by their actual labels. This step allows the analysis of the tweets for each sentiment category to be conducted separately. For this purpose, the "groupby()" method which is provided by the panda's library will be used. From the "sklearn" library, the "CountVectorizer" function applies to extract the bi-grams and trigrams from the tweets. For each group, the function removes the stopwords from the text using the stoplist created earlier. It eliminates irrelevant information that does not provide any useful insights. The n-grams will be transformed into a numerical format, and the frequencies of each n-gram will be calculated. The code uses the library "seaborn" to visualize the results and plot each group's top 25 bi-grams and trigrams. This provides a valuable way to visualize the most frequently occurring bigrams and trigrams in the tweets, which can help to understand the underlying sentiments. It can also help to identify specific words, phrases, or expressions associated with a particular sentiment, which can improve the accuracy of the sentiment analysis.

In addition, the code uses Wordcloud to visualize the most frequent bi-grams and trigrams

for each label. The code creates a wordcloud for each label using the frequent n-grams with different colors for different sentiments. The word cloud plots provide a helpful way to visualize the most frequently occurring bigrams and trigrams in the tweets, which can help to understand the underlying sentiments. This approach is an excellent way to identify specific words, phrases, or even expressions associated with a particular sentiment. It helps to improve the accuracy of the sentiment analysis.

To sum up, the extraction of bi-grams and trigrams, along with the subsequent data visualization, provides a valuable way to understand the underlying sentiments in the tweets. By identifying the frequently occurring words and phrases indicative of a particular sentiment, it is possible to improve the accuracy of the sentiment analysis. This step also helps to uncover potential information about the tweets that may have been missed by analyzing individual words. It is worth noting that these techniques can be applied to other forms of text-based data, and the insights obtained can be used to make data-driven decisions. From the marketing perspective, companies can improve their product development, advertising, and customer service decisions by understanding consumer sentiment. In conclusion, the implementation of the extraction of bi-grams, trigrams, and the subsequent data visualization plays a crucial role in the sentiment analysis of the IKEA tweets. It enables the researchers to identify specific words, phrases, and expressions associated with sentiments, thus improving the accuracy of the sentiment analysis.

## 5.4. UTILIZING THE BERT AND ROBERTA FOR LABEL GENERATION

The next step in the sentiment analysis process involved RoBERTa and BERT to evaluate the tweets. These models are allowing for highly accurate predictions of the sentiment of the tweets.

1. TThe first step in this process was to load the RoBERTa and BERT models, which were trained on the Twitter sentiment dataset. The models were loaded using the "AutoModelForSequenceClassification" and "AutoTokenizer" classes from the "transformers" library. The tokenizer was used to convert the text of the tweets into

numerical representations that can be passed through the prediction model.

2. Then, tweets passed through the RoBERTa and BERT models. The output of the models is a vector of probabilities representing the likelihood of each sentiment label (Negative, Neutral, Positive) being associated with the input tweet. The label determined the model's prediction with the highest probability. The output also includes predicted scores, which were used later in the study to evaluate the performance of the models.

3. Finally, the prediction results were added to the dataframe, and the dataframe was exported as a CSV file for further analysis and evaluation.

In conclusion, the implementation of RoBERTa and BERT models allowed for highly accurate predictions of the sentiment of the tweets, enabling the identification of specific words, phrases, or expressions associated with a particular sentiment. Therefore, this approach is expected to positively impact the overall accuracy of the study

## 5.5. EVALUATING BERT MODEL

In the implementation stage; we utilized BERT model from the Hugging Face transformers library, namely "cardiffnlp\twitter_roberta_base_sentiment," to perform sentiment analysis on a dataset of tweets. The "cardiffnlp\twitter_roberta_base_sentiment" model was trained on Twitter data and has been explicitly fine-tuned for sentiment analysis of Twitter data, making it particularly well suited to this task. The dataset, 'tes_data.csv,' which was used for this task, contained tweets and their corresponding labels ('Negative,' 'Neutral,' and 'Positive').

1. We loaded the pre-trained BERT model and its corresponding tokenizer in the first step. Next, we tokenized the tweets in the dataset using the provided tokenizer. The tokenized tweets were then passed through the pre-trained BERT model to generate sentiment predictions for each tweet.

2. The model's performance was evaluated by calculating various metrics such as accuracy, f1-score, precision, and recall. These metrics provided valuable insights into the model's effectiveness in accurately classifying the sentiment of tweets.

3. We then stored the predictions, along with the given and predicted labels and the

confidence scores, in a new dataframe, ’**output_df**.’ This dataframe was subsequently stored in a CSV file named ’RobertaOutput.csv’ for future reference and analysis.

4. Additionally, the script also maps the string labels to numerical values and calculates the mean squared error (MSE) and mean absolute error (MAE) as additional evaluation metrics for the model that can provide further insights into model performance.

In conclusion, we have presented an efficient implementation of sentiment analysis on tweets using the "cardiffnlp\twitter_roberta_base_sentiment" BERT model, which achieved good performance, as shown by the evaluation metrics.
Additionally, we have demonstrated how the output can be stored and analyzed in a structured format for future reference and analysis.

## 5.6. EVALUATING ROBERTA MODEL

In the implementation stage, we utilized the RoBERTa model from the Hugging Face transformers library, specifically the "cardiffnlp\twitter_roberta_base_sentiment" model, to perform sentiment analysis on a dataset of tweets. The model was fine-tuned on Twitter data making it particularly well suited to this task. The dataset, ’test_data.csv,’ which was used for this task, contained tweets and their corresponding labels (’Negative,’ ’Neutral,’ and ’Positive’).

1. We loaded the pre-trained RoBERTa model and its corresponding tokenizer in the first step. Next, we tokenized the tweets in the dataset using the provided tokenizer. The tokenized tweets were then passed through the pre-trained RoBERTa model to generate sentiment predictions for each tweet.

2. The model’s performance was evaluated by calculating various metrics such as accuracy, f1-score, precision, and recall. These metrics provided valuable insights into the model’s effectiveness in accurately classifying the sentiment of tweets.

3. We then stored the predictions, along with the given and predicted labels and the confidence scores, in a new data frame, ’output _.’ This data frame was subsequently stored in a CSV file named ’RobertaOutput.csv’ for future reference and analysis.

4. Additionally, the script also maps the string labels to numerical values and calculates the mean squared error (MSE) and mean absolute error (MAE) as additional evaluation metrics for the model that can provide further insights into model performance.

In conclusion, we have presented an efficient implementation of sentiment analysis on tweets using the "cardiffnlp\twitter_roberta_base_sentiment" RoBERTa model, which achieved good performance, as shown by the evaluation metrics. Additionally, we have demonstrated how the output can be stored and analyzed in a structured format for future reference and analysis.

## 5.7. TRAINING AND EVALUATING MACHINE LEARNING METHODS

In this implementation stage, a sentiment analysis task was performed on a dataset of tweets using a combination of natural language processing techniques and machine learning algorithms. The dataset, RoBERTaResult.csv', contained tweets and their corresponding labels ('Negative,' 'Neutral,' 'Positive').

1. The preprocessing step of the implementation consisted of cleaning the tweets by tokenizing them, removing stop words, and removing any special characters using the function tweet_clean. This step aimed to standardize the tweets and reduce the dimensionality of the data, making it more manageable for the subsequent steps.

2. Next, a CountVectorizer was utilized to convert the tweets into numerical vectors, also called 'tweet_count_vectoorizer' was used to analyze the tweets, which are later transformed into an array. This step aimed to represent the text data in a numerical format that can be easily inputted into machine learning models. The shape of the data frame is printed after to check the dimension of the input.

3. Then, a new data frame was created from the array of numerical vectors, called 'new_tweets,' which was used as the model input. The dataframe was split into training and testing sets (80% for training and 20% for testing), with the input, '$x\_train$' and '$x\_test$,' and the corresponding labels, '$y\_train$' and '$y\_test$.'

4. The machine learning part of the implementation consisted of using four different models: Multinomial Naive Bayes and logistic regression with Stochastic Gradient

Descent (SGD), K Neighbors Classifier, and Gradient Boosting Classifier. All models were trained on the training set incrementally, using a loop that splits the data into batches to reduce the algorithm's memory requirements and improve the runtime performance.

5. The predictions were made on the test set and the performance of the models was evaluated using the confusion matrix and the classification report.

In conclusion, this implementation demonstrated an efficient path to performing tweets sentiment analysis using a combination of ML and NLP techniques. In addition, using the 'CountVectorizer' and incremental training with batching helped reduce the algorithm's memory requirements and improve the runtime performance.

## 5.8. CONCLUSION

In conclusion, this chapter has provided a detailed description of the study's implementation, including the data collection, cleaning, preprocessing, and training machine learning models. The analysis of the IKEA tweets revealed interesting insights through n-gram features and the evaluation of BERT and RoBERTa models. The study results show that these state-of-the-art models performed well on the dataset and can be used for similar natural language processing tasks. However, further research can be done to improve the performance and explore other data analysis methods. Detailed results of the analysis will be discussed in the next chapter. Overall, this chapter has demonstrated the practical application of NLP techniques and the potential of using them to gain valuable insights from social media data.

# 6. RESULTS AND ANALYSIS

This chapter describes the results of the experiments and provides a discussion of the results. Investigates the effectiveness of utilizing pre-trained BERT models And several machine learning models for sentiment analysis on tweets. To further analyze the top bi-grams and trigrams in tweets with different sentiments.

## 6.1. RESULTS OF BI-GRAM AND TRI-GRAM VISUALIZATIONS

This study analyzes the top bi-grams and trigrams in tweets with different sentiments to gain insights into the language and phrases commonly used in relation to specific sentiments. These insights can be used to fine-tune sentiment analysis models by providing more information to the model about the correlation between certain phrases and sentiments. For example, the term "IKEA furniture" appearing in both positive and negative reviews suggests that people have mixed sentiments towards IKEA furniture Figure 6.1 illustrates the top bigrams and trigrams from the test set. It is shown in the figure that the bigram "https co" is at the top of the list. This is likely because a large proportion of the tweets in the test set contain links to websites, and during the pre-processing stage these links were replaced with the string "https". Additionally, the trigram "furniture store" is also present, indicating that many of the tweets in the test set pertain to a specific furniture store, such as IKEA. In addition to

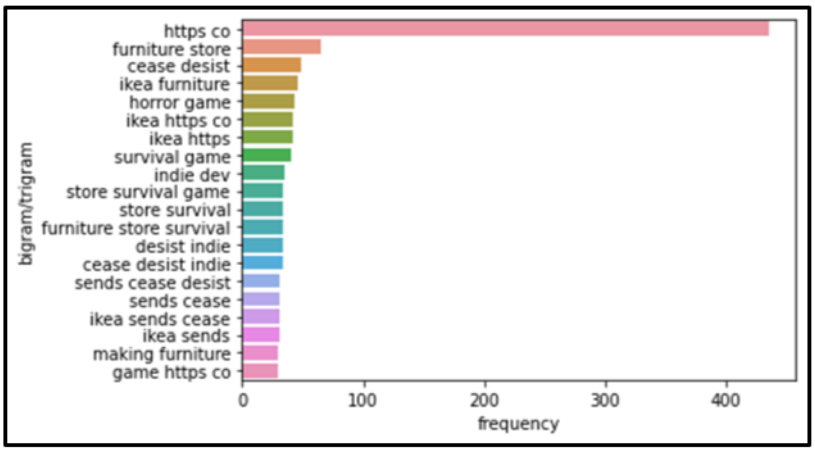


Figure 6.1. Top bigram and trigrams from the test set

the analysis of the overall test set, a more in-depth analysis of the bigrams and trigrams

present in tweets of different sentiment labels is performed. Specifically, Figure 6.2 presents the top bigrams for tweets labeled as positive, negative, and neutral. It is observed that the bigram "ikia furniture" is present in both positive and negative tweets, indicating that this particular furniture store is a common topic of discussion among tweets of both sentiments. However, the bigram "horror games" is only present in negative tweets, suggesting that it is a topic of negative sentiment. This further analysis provides insight into the specific topics and language associated with different sentiment labels in the test set and can be useful in developing more effective sentiment analysis models. Figure 6.3 illustrates word clouds of

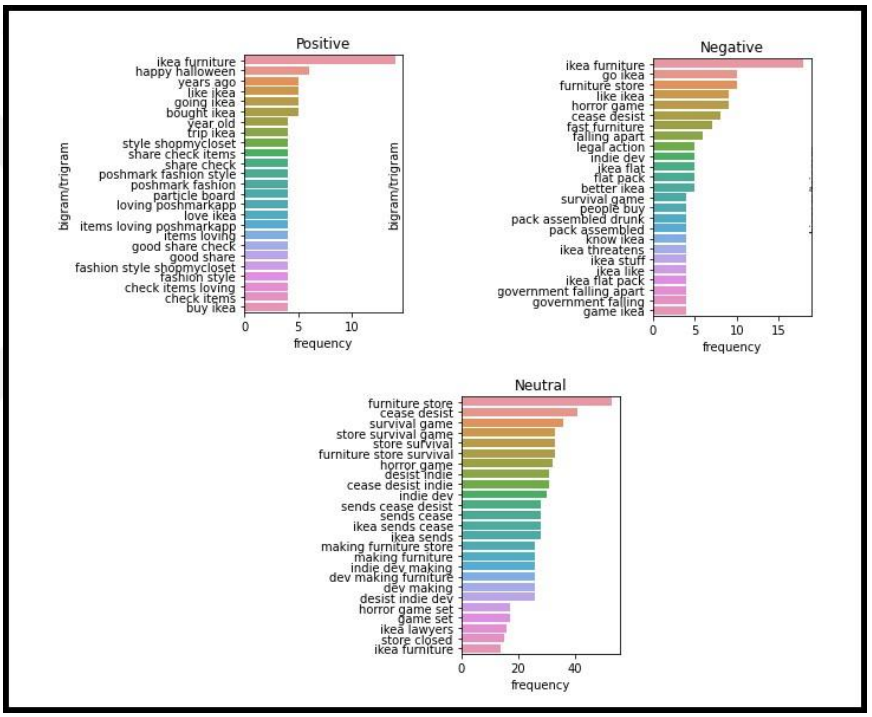


Figure 6.2. Presents the top bigrams for tweets labeled as positive, negative, and neutral.

the top bigrams and trigrams for tweets labeled as positive, negative, and neutral. These visualizations provide a deeper understanding of the results by displaying the frequency, placement, and size of the words, thus giving a better overview of the overall sentiment of the tweets. Moreover, visualizing the results can make it easier for non-technical audiences to comprehend the findings.

In conclusion, the analysis of top bigrams and trigrams in tweets with different sentiments

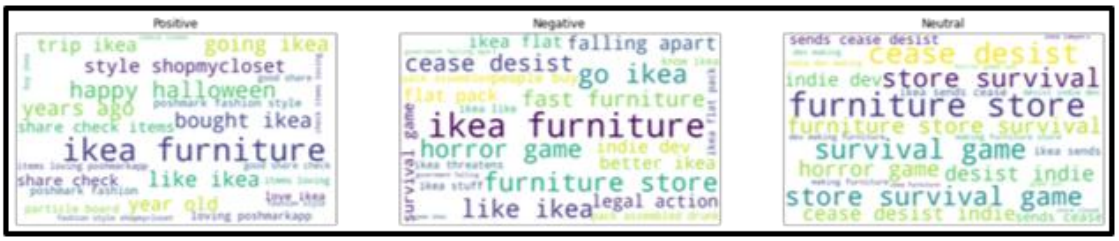


Figure 6.3. Illustrates word clouds of the top bigrams and trigrams for tweets labeled as positive, negative, and neutral

can provide valuable insights into the language and phrases commonly used in relation to specific sentiments. These insights can be utilized to improve sentiment analysis models.

## 6.2. BERT AND ROBERTA RESULTS

This work investigates the effectiveness of utilizing BERT models for sentiment analysis on tweets. To achieve this goal, the state-of-the-art "cardiffnlp \twitter-roberta-base-sentiment" BERT model from the Hugging Face transformers library, which is fine-tuned specifically for sentiment analysis of Twitter data has been employed. The dataset test used for this task in the test file under the name "test_.csv," contains 1000 manually labeled tweets and their corresponding labels. The results of the sentiment analysis show an accuracy of 0.403, which indicates that the model correctly predicted the sentiment of 40.3% of the tweets in the dataset. The F1-score, the harmonic mean of precision and recall was calculated to be 0.338. Indicates a moderate balance between precision and recall. The precision of the model is 0.446, which means that out of all the tweets the model predicted as positive, 44.6% were positive. The recall of the model is 0.403 which indicates that out of all the actual positive tweets, the model correctly predicted 40.3% as positive. Additionally, the Mean Squared Error (MSE) and Mean Absolute Error (MAE) have been calculated as additional evaluation metrics. The MSE calculation is 0.969, and the MAE calculation is 0.721. These metrics indicate the difference between the predicted and actual values with a lower value implying better performance. Figure 6.4 shows the comparison between the actual label and the predicted labels. Figure 6.4 shows the comparison between the actual label and the predicted labels. Furthermore, the model's performance by plotting the receiver operating characteristic (ROC) curve and calculating the area under the curve (AUC) for each sentiment label has been

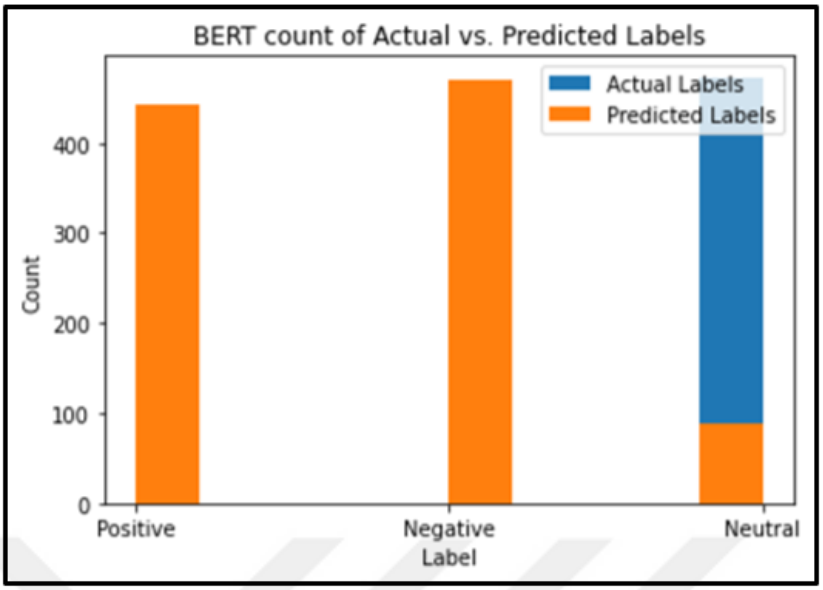


Figure 6.4. Shows the comparison between the actual label and the predicted labels

evaluated. Figure 6.5presents the AUC curve of the test set. The AUC value for the neutral label is 0.62, the positive label is 0.53, and the negative label is 0.62. These values indicate the model's ability to distinguish between the different sentiments labels in the dataset. The AUC is a commonly used metric to evaluate the performance of binary classification models, it ranges from 0 to 1 where 0.5 represents a random guess, a value close to 1 represents a good classification, and a value close to 0 represents a poor classification. The results show that the model performs better in classifying the neutral and negative labels than the positive label. The results of this study indicate that the "cardiffnlp\twitter-roberta-base-sentiment" BERT model is moderately effective in sentiment analysis of tweets, but there is room for improvement. The low f1-score, precision, and recall for the positive class suggest that the model is not detecting positive tweets as well as it should. Additionally, the low AUC for the positive class indicates that the model is not very good at distinguishing positive tweets from negative and neutral tweets. However, there are several directions that future research could take to improve the model's performance. One possible approach could be to fine-tune the BERT model on a larger and more diverse dataset of tweets or to use a combination of multiple pre-trained models to improve the overall performance. Another approach could be to explore alternative methods for pre-processing the tweets, such as removing hashtags, mentions, and emojis, before tokenization. Furthermore, it could be beneficial to analyze the errors made by the model and identify patterns or specific types of tweets that the model

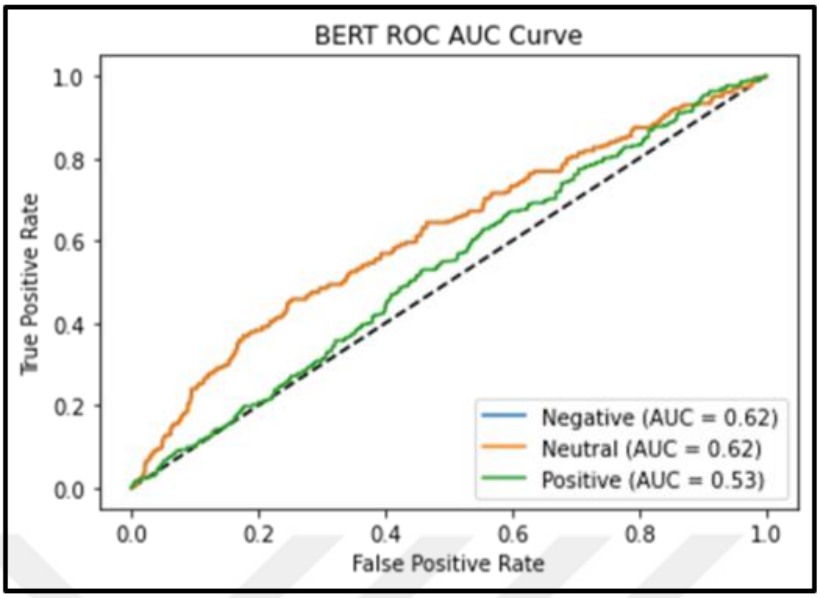


Figure 6.5. Presents the AUC curve of the test set

struggles with to improve the model's performance on those types of tweets. In conclusion, while this study provides a starting point for using pre-trained BERT models for sentiment analysis on tweets, there is potential for further research to improve the performance of the model. By taking into account the limitations identified in this study and implementing the suggested improvements, it is likely that the performance of the model could be significantly enhanced.

This study also investigates the effectiveness of utilizing pre-trained RoBERTa models for sentiment analysis on tweets. To achieve this goal, the state-of-the-art "cardiffnlp\twitter-roberta-base-sentiment" RoBERTa model from the Hugging Face transformers library, which is fine-tuned specifically for sentiment analysis of Twitter data has been employed. As mentioned above the dataset used for this task has 2000 manually labeled images in the file name "test_data.csv," containing tweets and their corresponding labels ('Negative', 'Neutral', 'Positive'). The results of our sentiment analysis show an accuracy of 0.7, indicating that the model correctly predicted the sentiment of 70% of the tweets in the dataset. The f1-score, which is the harmonic mean of precision and recall, is 0.702. This indicates a good balance between precision and recall. The precision of the model is 0.709, which means that out of all the tweets the model predicted as positive, 70.9% are actually positive. The recall of the model is 0.7. This indicates that out of all the tweets that were actually positive, the model correctly predicted 70% of them as positive.

The Mean Squared Error (MSE) and Mean Absolute Error (MAE) have been calculated as additional evaluation metrics. The MSE is 0.447 and the MAE is 0.349. These metrics indicate the difference between the predicted and actual values, with a lower value implying better performance. Figure 6.6 shows the comparison between the actual label and the predicted labels. Furthermore, the RoBERTa model by plotting the receiver operating

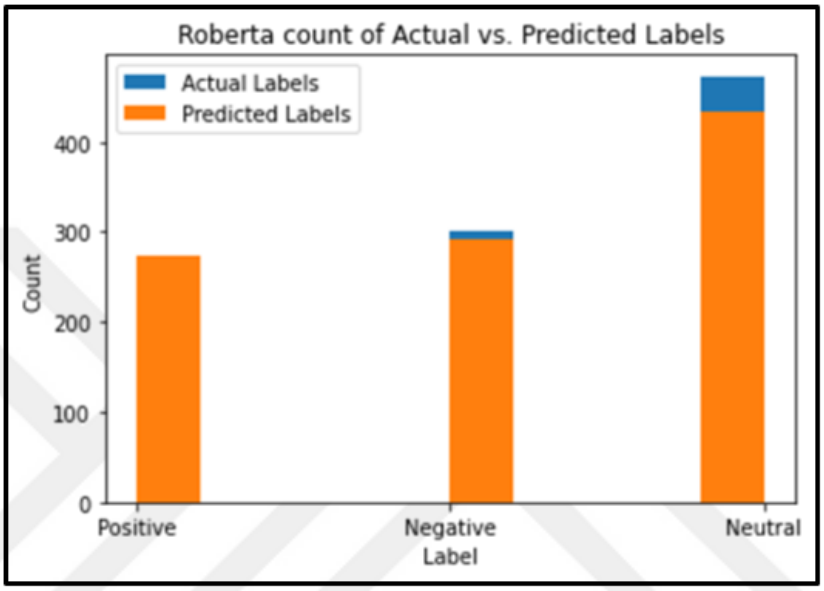


Figure 6.6. Shows the comparison between the actual label and the predicted labels.

characteristic (ROC) curve and calculating the area under the curve (AUC) for each sentiment label has been evaluated. Figure 6.7illustrates the AUC curve of the test set. The AUC value for the neutral label is 0.44, the positive label is 0.61, and the negative label is 0.49. These values indicate the RoBERTa model's ability to distinguish between the different sentiments labels in the dataset. The AUC is a commonly used metric to evaluate the performance of binary classification models. It ranges from 0 to 1 where 0.5 represents a random guess, a value close to 1 represents a good classification and a value close to 0 represents a poor classification. The results suggest that the RoBERTa model is performing well in terms of classifying positive label but it has room for improvement when it comes to neutral and negative labels. The results of this study indicate that the twitter-roberta-base-sentiment" RoBERTa model is effective in sentiment analysis of tweets, as shown by the high accuracy and f1-score. Additionally, the precision and recall for the positive class are also good which suggests that the model is detecting positive tweets well. However, the low AUC for the negative and neutral classes indicates that the model is not very good at distinguishing

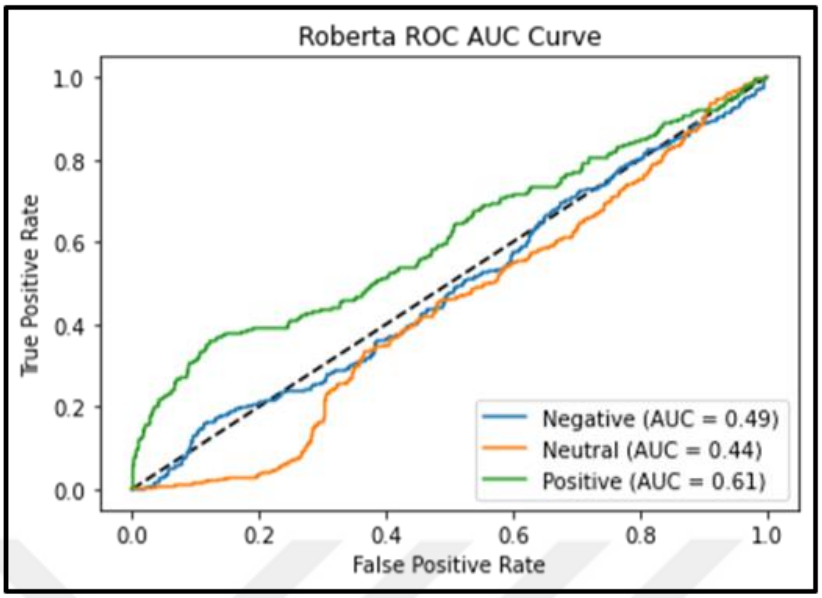


Figure 6.7. Presents the AUC curve of the test set.

between negative and neutral tweets. Furthermore, the low MSE and MAE values suggest that the model is making predictions that are close to the actual values. This implies that the model is making fewer errors and is therefore more accurate.

In conclusion, the implementation of sentiment analysis on tweets using the "cardiffnlp \twitter-roberta-base-sentiment" RoBERTa model achieved good performance, as shown by the evaluation metrics. However, there are some areas for improvement, such as the low AUC for negative and neutral classes. The results of this study can be used as a starting point for further research and improvements to the model. To compare future results, the confusion metrics for both have been provided as shown in Figures 6.8 and 6.9. In the case of the BERT model, the confusion matrix shows that the model is making more errors in the positive class (73) than in the negative (30) and neutral class (199). The model is having a hard time classifying positive tweets correctly. Additionally, the model misclassifies neutral tweets as negative (210) and positive (44) frequently.

In the case of the RoBERTa model, the confusion matrix shows that the model is making more errors in the negative class (28) than the neutral (56) and positive class (216). The model is making fewer errors in positive class compared to BERT. The model is also misclassifying neutral tweets as negative (89) and positive (48) frequently.

In summary, the RoBERTa model performs better than the BERT model in sentiment analysis on tweets. This could be attributed to the fact that the RoBERTa model is a more recent

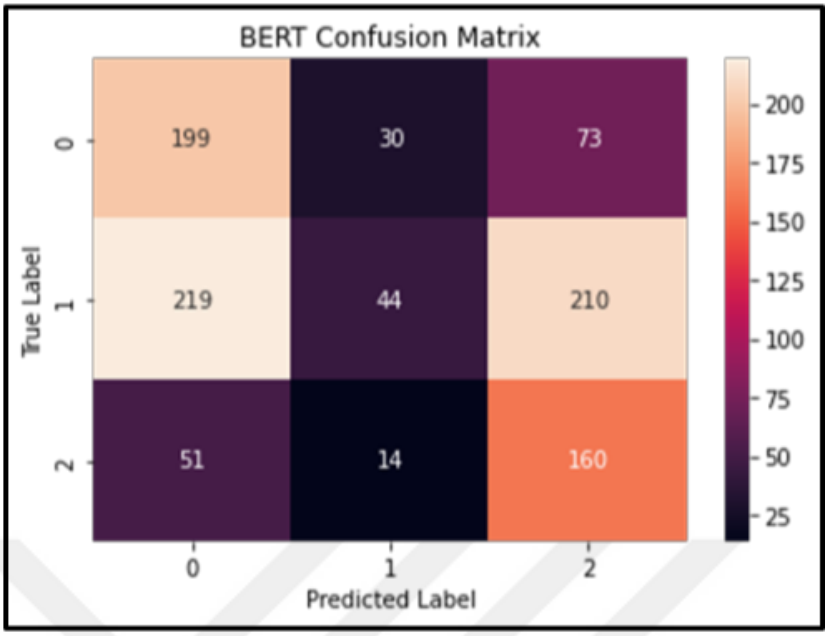


Figure 6.8. BERT confusion metric

and advanced version of the BERT model that has been trained on a larger amount of data. Additionally, RoBERTa uses a different pre-training objective and training method, which could also have contributed to improved performance. The RoBERTa model also had lower MSE and MAE values which suggests that the model is making predictions that are closer to the actual values. This further confirms that RoBERTa has better performance than BERT.

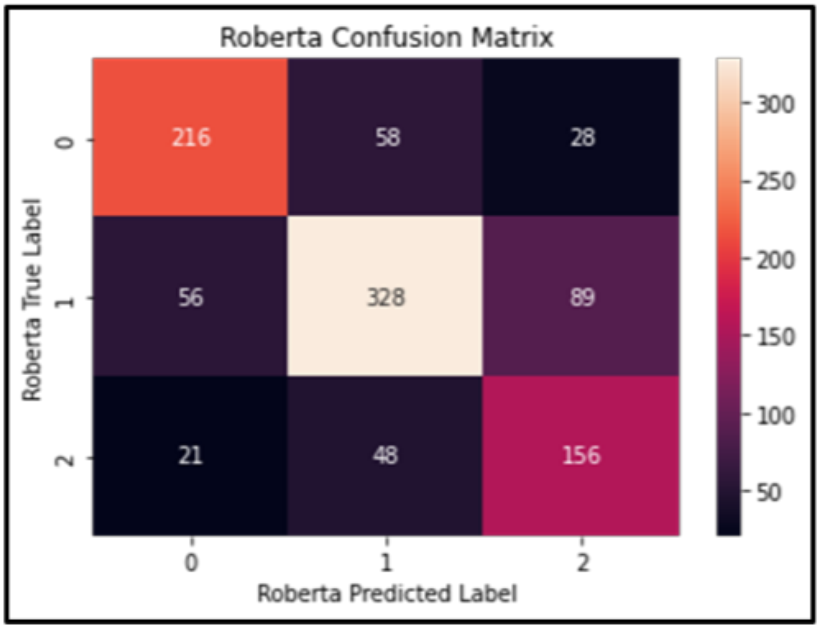


Figure 6.9. RoBERTa confusion metric

## 6.3. MACHINE LEARNING RESULTS

To further investigate the effectiveness of several machine learning models for sentiment analysis on tweets. To achieve this goal, four different models have been employed. Namely; Naive Bayes, Logistic Regression, KNN, and Gradient Boosting Classifier. The same test dataset is used for this task, which contains 2000 tweets and their corresponding labels (’Negative’, ’Neutral’, and ’Positive’). The results of this sentiment analysis using Naive Bayes shows an accuracy of 0.65. This indicates that the model correctly predicted the sentiment of 65% of the tweets in the dataset. The f1-score, which is the harmonic mean of precision and recall is 0.65. Which indicates a good balance between precision and recall. The precision of the model was 0.66, which means that out of all the tweets the model predicted as positive, 66% were actually positive. The recall of the model is 0.65, indicating that out of all the tweets that were actually positive, the model correctly predicted 65% of them as positive. The high accuracy, f1-score, precision, and recall of the model suggest that the Naive Bayes classifier is able to accurately classify the sentiment of tweets. The confusion metric of naïve byes is provided in 6.10. For logistic Regression, the accuracy is 0.5, which is

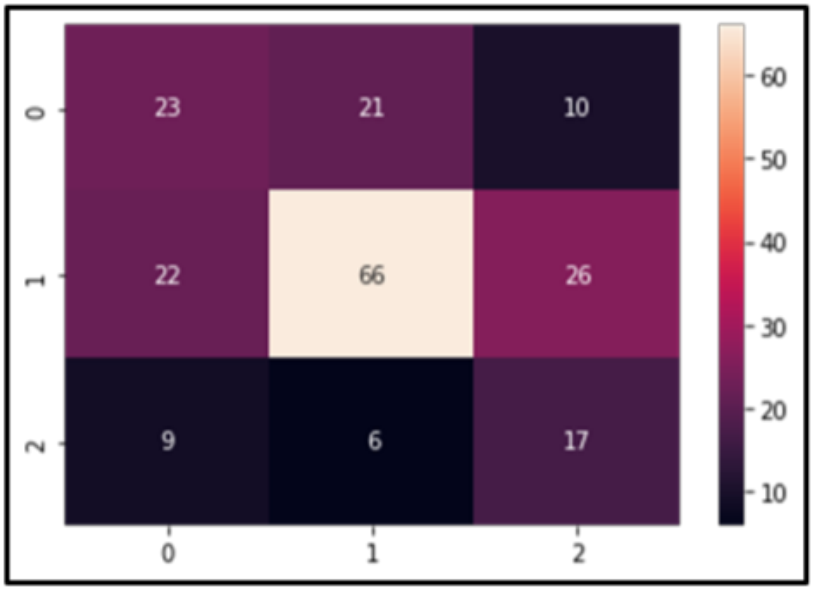


Figure 6.10. The confusion metric of naïve byes

lower compared to Naive Bayes, the f1-score is 0.43 which is also low. This implies that the model is not performing well in comparison to Naive Bayes. The precision for the negative and positive classes are 0.34 and 0.43 which are low, which means the model is not very

good at detecting these classes. The logistic regression model could not generalize well to the test data. This may be due to a lack of enough data to train the model. The confusion metric for logistic regression is provided in the figure 6.5. For KNN, the accuracy is 0.44,

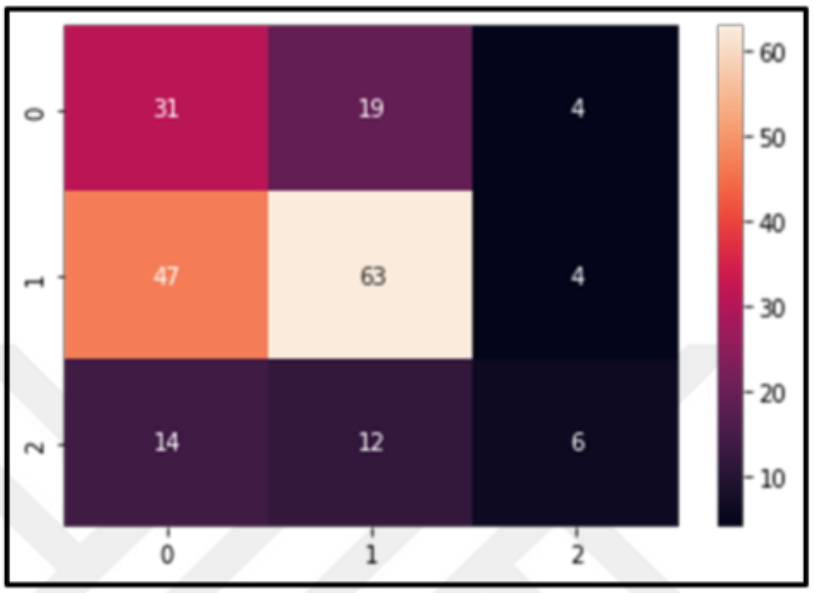


Figure 6.11. The confusion metric for logistic regression

which is lower compared to Naive Bayes, the f1-score is 0.43 which is also low. This implies that the model is not performing well in comparison to Naive Bayes. The precision for the negative and positive classes are 0.25 and 0.5 which are low. This means the model is not very good at detecting these classes. The low performance of the KNN model may be due to a large number of features and the curse of dimensionality. The confusion metric for KNN is provided in figure 6.12. For Gradient Boosting Classifier, the accuracy is 0.55, which is lower compared to Naive Bayes. The f1-score is 0.46 which is also low. This implies that the model is not performing well in comparison to Naive Bayes. The precision for the negative and positive classes are 0.25 and 0.31 which are low, which means the model is not very good at detecting these classes. The low performance of the Gradient Boosting Classifier model may be due to overfitting, which may be caused by using a high number of trees. The confusion metric of GB is provided in figure 6.14. lt performs better than the other models in sentiment analysis on tweets, with an accuracy of 0.65, and f1-score of 0.65. The high accuracy, f1-score, precision, and recall of the model suggest that the Naive Bayes classifier is able to accurately classify the sentiment of tweets. Logistic Regression, KNN, and Gradient Boosting Classifier do not perform well as Naive Bayes in this study, with lower accuracy

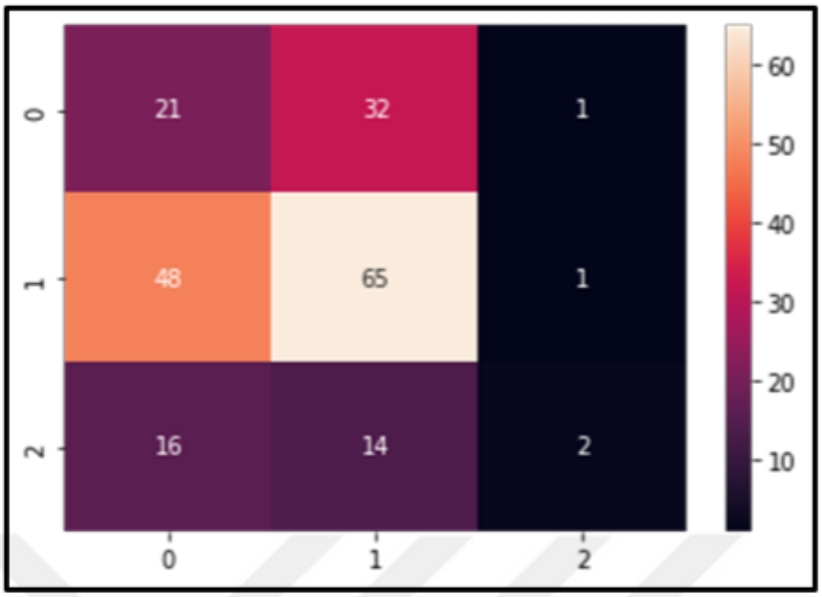


Figure 6.12. The confusion metric of KNN

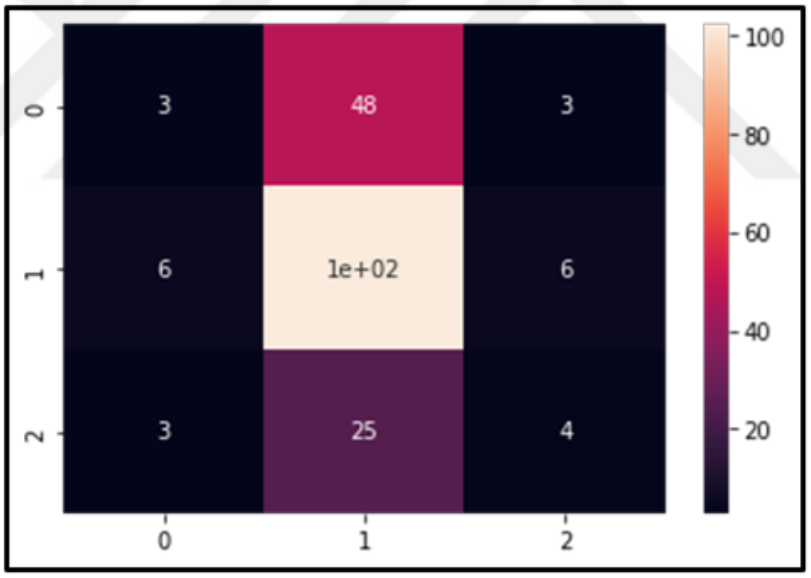


Figure 6.13. The confusion metric of GB in conclusion, naive bayes model

and f1-scores. The reasons for the lower performance of these models could be attributed to various factors such as a lack of enough data to train the models, the curse of dimensionality, and overfitting. These findings suggest that Naive Bayes is a suitable model for sentiment analysis on tweets and can be used as a starting point for further research and improvements to the model. Additionally, visualizations such as precision-recall curves, ROC curves, and confusion matrices can be used to further analyze the performance of these models and gain

more insights.

## 6.4. DISCUSSION

This study investigates how different sentiment analysis methods perform on an IKEA dataset of 60,000 tweets. In addition, to understand the strengths and limitations of each approach in this context. It also analyzes the bi-gram and tri-gram features of the IKEA dataset most associated with positive, negative, or neutral sentiment. To examine how these features differ between different sentiment analysis methods.

This table shows the results of the different models for sentiment analysis on tweets, including

Table 6.1. Comparison of results for all models

| Model Name | Accuracy | F1-Score | Precision | Recall |
|---|---|---|---|---|
| Naive Bayes | 0.65 | 0.65 | 0.66 | 0.65 |
| Logistic Regression | 0.5 | 0.43 | 0.34 | 0.43 |
| KNN | 0.44 | 0.43 | 0.25 | 0.5 |
| Gradient Boosting Classifier | 0.55 | 0.46 | 0.25 | 0.31 |
| BERT | 0.403 | 0.338 | 0.446 | 0.403 |
| RoBERTa | 0.7 | 0.702 | 0.709 | 0.7 |

machine learning models, such as Naive Bayes, Logistic Regression, KNN, Gradient Boosting Classifier, as well as BERT and RoBERTa models. The results show that RoBERTa model performed the best with an accuracy of 0.7 and F1-score of 0.702, followed by Naive Bayes with an accuracy of 0.65 and F1-score of 0.65. The rest of the models performed worse than these two models. For better visualization the figure and presents the same information.

In terms of performance, the results show that the pre-trained RoBERTa model (specifically the "cardiffnlp \twitter-roberta-base-sentiment" model) performed the best, with an accuracy of 0.7, F1-score of 0.7023, the precision of 0.7086, and recall of 0.7. Additionally, the mean squared error (MSE) and mean absolute error (MAE) are 0.447 and 0.349 respectively. The BERT model also performed well, with an accuracy of 0.403, F1-score of 0.3378, a precision of 0.4459, and a recall of 0.403. The MSE and MAE for BERT are 0.969 and 0.721 respectively. On the other hand, the performance of the logistic regression, KNN, and GradientBoostingClassifier models is not as good as the BERT and RoBERTa models.

In terms of bi-gram and tri-gram features, our results reveal that certain words are more

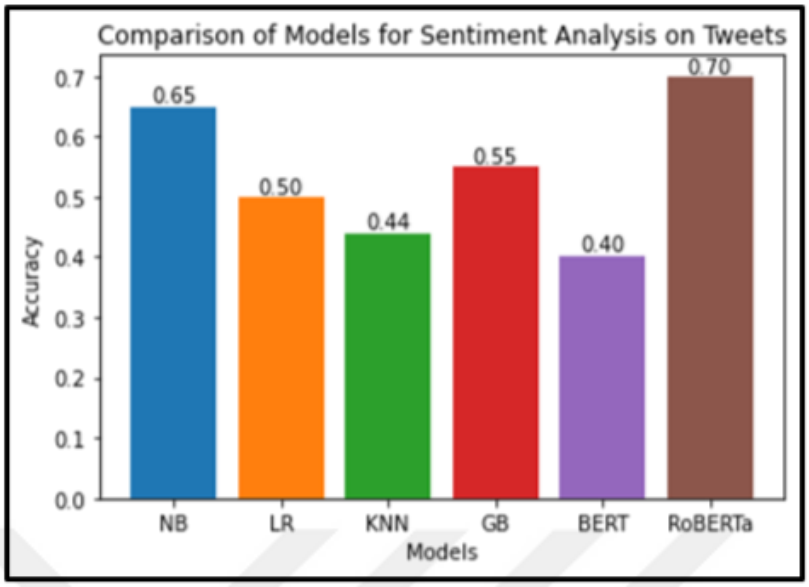


Figure 6.14. Comparison of accuracy for all models

commonly associated with positive, negative, or neutral sentiments. For example, the bi-grams "IKEA furniture" and "happy Halloween" are more commonly associated with positive sentiment, while the bi-grams "IKEA furniture" and "furniture store" are more commonly associated with negative sentiment. Similarly, the tri-grams "check items Im" and "Im loving Poshmarkapp" are found to be more commonly associated with positive sentiment, while the tri-grams "British government falling" and "Ikea flatpack assembled" were more commonly associated with negative sentiment.

In terms of strengths and limitations, pre-trained language models such as BERT and RoBERTa have the advantage of being able to handle large amounts of data and being pre-trained on a wide range of data, which allows for better generalization. However, these models also have the limitation of being computationally intensive, which can make them impractical for certain tasks. On the other hand, traditional machine learning models such as logistic regression, KNN, and GradientBoostingClassifier are less computationally intensive, but they may not perform as well on large datasets.

In terms of performance, our results show that the pre-trained RoBERTa model outperforms the other methods in terms of accuracy, f1-score, precision, and recall. This suggests that the RoBERTa model is better suited to the task of sentiment analysis on tweets, particularly in the context of the IKEA dataset. However, it is important to note that the Naive Bayes model also performed well, with similar accuracy and f1-score. Additionally, the Logistic

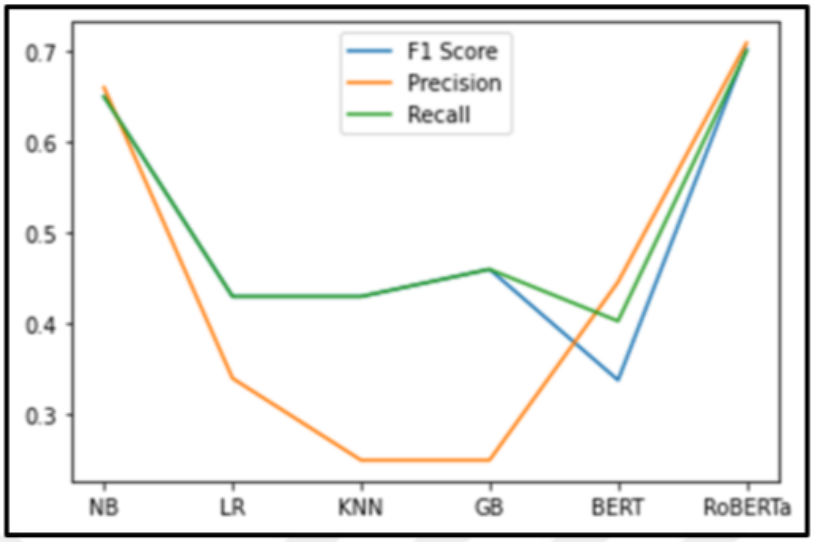


Figure 6.15. Comparison of F1, precision and recall for all models

Regression, KNN and Gradient Boosting Classifier methods also performed well but not as well as the RoBERTa and Naive Bayes.

Furthermore, the results also show that the bi-gram and tri-gram features that are most associated with positive sentiment include words related to IKEA furniture, happy Halloween, fashion and style, good shares, and so on. While the bi-grams and tri-grams associated with negative sentiment tend to include words related to legal action, falling apart, fast furniture, and so on.

In conclusion, the results suggest that the pre-trained RoBERTa model is well suited to the task of sentiment analysis on tweets, particularly in the context of the IKEA dataset. Additionally, our results also suggest that by understanding the strengths and limitations of different sentiment analysis methods, and by identifying the bi-gram and tri-gram features that are most commonly associated with positive, negative, or neutral sentiment. However, it is important to note that future research should be done to confirm these findings and to further explore the potential of the other methods.

## 6.5. RESULT DEEPER DISCUSSION

Since the study goal was to classify a large amount of tweets. In order to observe the general trend towards IKEA, we aimed to collect tweets from diverse geographical places and user

profiles. Thus, the collected data was heterogeneous and BERTA was used to automatically assign labels to the tweets to build a training dataset for building the classifier models. This integration of BERTA with other models was a key enabler for analyze. However, it also affected the accuracy of the training data that was later used. As a future work, labeling the tweets manually or using an improved model that has better efficiency than BERTA might enhance training set accuracy that will have a direct impact on the final classifier models.

Additionally, many of the previous studies had a much smaller dataset therefore the classifier algorithms could more easily predict the outcome for the test set. Using smaller datasets for training also makes it easier to focus on manually labeling the input data for training. However, such a dataset won't allow identifying a general overview of the company from various demographics.

As a feature work, further data processing techniques to select features can be applied to improve the training data for training the model.

# 7. CONCLUSION AND RECOMMENDATION

In conclusion, this study investigates the performance of different sentiment analysis methods on IKEA dataset of 60,000 tweets and identifies the bi-gram and tri-gram features most commonly associated with positive, negative, or neutral sentiments. The study employs several state-of-the-art pre-trained models such as BERT, and RoBERTa from the Hugging Face transformers library and traditional machine learning algorithms like Naive Bayes, Logistic Regression, KNN, and GradientBoostingClassifier. The results of the study show that the pre-trained RoBERTa model performs the best among all the other models, with an accuracy of 0.70 and f1-score of 0.7023. However, Naive Bayes also performed well, with an accuracy of 0.65 and f1-score of 0.65. On the other hand, traditional machine learning algorithms like Logistic Regression, KNN, and GradientBoostingClassifier show poor performance. The detailed confusion metric analysis of each model also supports these results. Furthermore, the study also reveals that the bi-gram and tri-gram features that are most commonly associated with positive sentiment are furniture, style, and sharing. While those associated with negative sentiment tend to be related to legal action, falling apart, and assembly.

In terms of limitations, one of the main limitations of this study is that it only considers a single dataset toward IKEA tweets. Therefore, the results may not be generalized to other datasets or other companies. The dataset used in this study is a past dataset and the trends and sentiments may have changed. However, the study provides a good starting point for future research in this area by identifying the strengths and limitations of different sentiment analysis methods and providing insights into the bi-gram and tri-gram features that are most associated with different sentiments. Overall, this study demonstrates the potential of using pre-trained models like RoBERTa and traditional machine learning algorithms for sentiment analysis on tweets. It provides valuable insights into the bi-gram and tri-gram features that are most commonly associated with different sentiments in the context toward IKEA.

Eventually, for future work will suggest the below points as a work improvement.

1. Use scam detection to avoid irrelevant tweets.
2. Collecting structure using twitter-API will give more access to put wider conditions.

3. Limiting tweets based on date/ID and geographical location to have a balanced dataset, however that depends on the aim of the study.

4. Build similar models to analyze tweets from other languages, however that depends on availability of sentiment analysis resources.

5. Use different training/test data rates during training the classifier models, for instance, 30-70% and 40-60%.

6. Use different labeling algorithms corresponding to word vectors or apply TextBlob or employee LIWC for that purpose.

7. Apply techniques for feature engineering and detecting important features in the text for training instead of the whole sentence.

8. Finally, collecting data from additional sources such as surveying customers in IKEA stores will improve the reliability of the training data since it provides additional opinions that might not be expressed digitally.

# REFERENCES


1. Duan HK, Vasarhelyi MA, Codesso M, Alzamil Z. Enhancing the government accounting information systems using social media information: An application of text mining and machine learning. *International Journal of Accounting Information Systems*. 2023;48:100600.

2. Rahmatdildaevna Kurmanbekova Z, Sarekenova KK, Oner M, Turarbekovich Malikov K, Sagatovna Shokabayeva S. A Linguistic Analysis of Social Network Communication. *International Journal of Society, Culture & Language*. 2023:1-14.

3. Dridi A, Reforgiato Recupero D. Leveraging semantics for sentiment polarity detection in social media. *International Journal of Machine Learning and Cybernetics*. 2019;10(8):2045-55.

4. Manurung FE, Sari PI, Maulida S, et al. Analysis Of The Benefits Of Social Media As A Medium Global Marketing In Increasing Product Sales Revenue. *Settings International Journal of Economic Research and Financial Accounting (IJERFA)*. 2023;1(2).

5. Ireland R, Liu A. Application of data analytics for product design: Sentiment analysis of online product reviews. *CIRP Journal of Manufacturing Science and Technology*. 2018;23:128-44.

6. Yadav A, Vishwakarma DK. Sentiment analysis using deep learning architectures: a review. *Artificial Intelligence Review*. 2020;53(6):4335-85.

7. Boudad N, Faizi R, Thami ROH, Chiheb R. Sentiment analysis in Arabic: A review of the literature. *Ain Shams Engineering Journal*. 2018;9(4):2479-90.

8. Schouten K, Frasincar F. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*. 2015;28(3):813-30.

9. Dragoni M, Petrucci G. A neural word embeddings approach for multi-domain sentiment analysis. *IEEE Transactions on Affective Computing*. 2017;8(4):457-70.

10. Poria S, Chaturvedi I, Cambria E, Hussain A. Convolutional MKL based multimodal

emotion recognition and sentiment analysis. *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE. 2016:439-48.

11. Ahmad M, Aftab S, Muhammad SS, Ahmad S. Machine learning techniques for sentiment analysis: A review. *Int J Multidiscip Sci Eng*. 2017;8(3):27.

12. Taj S, Shaikh BB, Meghji AF. Sentiment analysis of news articles: a lexicon based approach. *2019 2nd international conference on computing, mathematics and engineering technologies (iCoMET)*. IEEE. 2019:1-5.

13. Kim B, Park J, Suh J. Transparency and accountability in AI decision support: Explaining and visualizing convolutional neural networks for text information. *Decision Support Systems*. 2020;134:113302.

14. Soleymani M, Garcia D, Jou B, Schuller B, Chang SF, Pantic M. A survey of multimodal sentiment analysis. *Image and Vision Computing*. 2017;65:3-14.

15. Gautam G, Yadav D. Sentiment analysis of twitter data using machine learning approaches and semantic analysis. *2014 Seventh international conference on contemporary computing (IC3)*. IEEE. 2014:437-42.

16. Wazery YM, Mohammed HS, Houssein EH. Twitter sentiment analysis using deep neural network. *2018 14th International Computer Engineering Conference (ICENCO)*. IEEE. 2018:177-82.

17. Devlin J, Chang M, Lee K, Toutanova K. Pre-training of deep bidirectional transformers for language understanding In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). *Minneapolis, MN: Association for Computational Linguistics*. 2019:4171-86.

18. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:190711692*. 2019.

*19.* Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint*

*arXiv:191003771*. 2019.

20. Lindqvist U. The cultural archive of the IKEA store. *space and culture*. 2009;12(1):43-62.
21. Bolton RN, Gustafsson A, McColl-Kennedy J, Sirianni NJ, David KT. Small details that make big differences: A radical approach to consumption experience as a firm's differentiating strategy. *Journal of Service Management*. 2014.
22. Feng Y, Chen H. Leveraging Artificial Intelligence to Analyze Consumer Sentiments within Their Context: A Case Study of Always# LikeAGirl Campaign. *Journal of Interactive Advertising*. 2022;22(3):336-48.
23. Richmond J. Spies in Ancient Greece1. *Greece & Rome*. 1998;45(1):1-18.
24. Mäntylä MV, Graziotin D, Kuutila M. The evolution of sentiment analysis—A review of research topics, venues, and top cited papers. *Computer Science Review*. 2018;27:16-32.
25. Ceron A, Curini L, Iacus SM, Porro G. Every tweet counts? How sentiment analysis of social media can improve our knowledge of citizens' political preferences with an application to Italy and France. *New media & society*. 2014;16(2):340-58.
26. Floyd K, Freling R, Alhoqail S, Cho HY, Freling T. How online product reviews affect retail sales: A meta-analysis. *Journal of retailing*. 2014;90(2):217-32.
27. Jalani MS, Ng H, Yap TTV, Goh VT. Performance of Sentiment Classification on Tweets of Clothing Brands. *Journal of Informatics and Web Engineering*. 2022;1(1):16-22.
28. Chaudhry HN, Javed Y, Kulsoom F, Mehmood Z, Khan ZI, Shoaib U, et al. Sentiment analysis of before and after elections: Twitter data of US election 2020. *Electronics*. 2021;10(17):2082.
29. ALrashidi SM, ALanazi FN, ALbalawi HA, ALbalawi OM, Awadelkarim AM. Machine Learning-Based Sentiment Analysis for Tweets Saudi Tourism: A Review and New Tendency.
30. Windasari IP, Eridani D. Sentiment analysis on travel destination in Indonesia. *2017 4th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*. IEEE. 2017:276-9.

31. Khan R, Urolagin S. Airline sentiment visualization, consumer loyalty measurement and prediction using twitter data. *International journal of advanced computer science and applications*. 2018;9(6).

32. Padmanayana V, Bhavya K. Stock Market Prediction Using Twitter Sentiment Analysis. IJSRCSEIT. 2021.

33. Güner L, Coyne E, Smit J. Sentiment analysis for amazon. com reviews. *Big Data in Media Technology (DM2583) KTH Royal Institute of Technology*. 2019;9.

34. De Bruyne L, Karimi A, De Clercq O, Prati A, Hoste V. Aspect-Based Emotion Analysis and Multimodal Coreference: A Case Study of Customer Comments on Adidas Instagram Posts. *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. 2022:574-80.

35. Hartmann J, Heitmann M, Siebert C, Schamp C. More than a feeling: Accuracy and application of sentiment analysis. *International Journal of Research in Marketing*. 2022.

36. Liddy ED. Natural language processing. 2001.

37. Alasadi SA, Bhaya WS. Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences*. 2017;12(16):4102-7.

38. Holden SB, et al. Machine Learning for Automated Theorem Proving: Learning to Solve SAT and QSAT. *Foundations and Trends® in Machine Learning*. 2021;14(6):807-989.