

# ChatGPT Tweets Sentiment Analysis Using Machine Learning and Data Classification

Aliea Sabir<sup>1</sup>, Huda A. Ali<sup>2</sup>, Maalim A. Aljabery<sup>2</sup>

<sup>1</sup>Faculty of Computer Science and Information Technology, Computer Information System Dept., University of Basrah, Basrah, Iraq

<sup>2</sup>Faculty of Computer Science and Information Technology, Computer Science Dept., University of Basrah, Basrah, Iraq

E-mail:

**Keywords:** sentiment analysis, natural language processing, ChatGPT, machine learning algorithms, feature extraction, classification algorithms

**Received:** December 11, 2023

*Many things, such as goods, products, and websites are evaluated based on user's notes and comments. One popular research project is sentiment analysis, which aims to extract information from notes and comments as a natural language processing (NLP) to understand and express emotions. In this study we analyzed the sentiment of ChatGPT labeled tweet datasets sourced from the Kaggle community using five Machine Learning (ML) algorithms; decision tree, KNN, Naïve Bayes, Logistic Regression, and SVM. We applied three feature extraction techniques: positive/negative frequency, a bag of words (count vector), and TF IDF. For each classification algorithm. The results were assessed using accuracy measures. Our experiments achieved an accuracy of 96.41% with the SVM classifier when using TF- IDF as a feature extraction technique.*

*Povzetek: Raziskava obravnavajo analizo sentimenta tvitov ChatGPT z uporabo petih algoritmov strojnega učenja in treh tehnik ekstrakcije značilk, pri čemer SVM z metodo TF-IDF doseže 96,41% točnost.*

## 1 Introduction

The application of text sentiment analysis is one of the applications of artificial intelligence to investigate the implicit sentiments in texts, whether they are bad, good, or neutral. It is very popular in many fields, such as analyzing sentiment towards new services or products, as well as following up on audience opinions and activating the brand to develop digital shopping and digital products [1][2][3]. Through several methods, emotions in texts are analyzed, including language modeling techniques, artificial intelligence, and machine learning. These methods are used to examine many emotions, including contentment, fear, sadness, anger, and others. These techniques are based on linguistic text analysis, statistical conclusions, and text context analysis to distinguish the forms of emotions contained in the text [4][5][6].

Moreover, deep learning methods are used to enhance the analysis outcomes [7]. Machine learning models that are trained on sentiment analysis are used by These techniques [8].

Comments and responses to services and products, analyzing text conversations and emails, monitoring social media sites, as well as analyzing political, news, and financial reports are among the most popular types of uses of text sentiment analysis. It contributes to understanding customers' opinions and trends to enhance and develop marketing and commercial relationships, etc. [2][4][10].

This study aims to utilize different feature extraction techniques as well as machine-learning algorithms to classify chatGPT tweet reviews. it contributes to

introducing a comparative analysis of different feature extraction techniques and machine learning algorithms. The paper is organized as section two highlights the recent related works, then discusses the methodology in section three. Section four explains the experimental results. Finally, the last section discussed the conclusion as well as the future work.

## 2 Related work

Sentiment analysis is an explored field that encompasses techniques including machine learning, rule-based approaches, and bag of words (BOW) methods. Researchers focus on two areas within sentiment analysis: analyzing sentiments at the document level and at the sentence level. These approaches rely on identifying terms that convey emotions or feelings.

A derived technique is employed to extract emotional terms from sentences using the Wordnet POS (Part-of-Speech) characteristic. The words are then determined sentiment polarity using either a lexicon-based approach or the "Sentiwordnet" dictionary. The machine learning algorithms Naïve Bayes (NB) and SVM (Support Vector Machine) are engaged to further evaluate the polarity given as a result [4]. This work goals to provide the most sufficient method for extracting polarity from social media texts by suggesting a novel machine learning-based sentiment analysis system. The policy framework is based on the Bayesian Rough Decision Tree (BR DT) algorithm, and the system performed an accuracy of 95% on social media data [5].

Three deep learning networks utilized for sentiment analysis of IMDB movie reviews are displayed in this research. 50% of the dataset contains positive feedback and 50% of negative feedback. They used Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) neural networks as two essential combinations that are repeatedly employed in NLP tasks. The outcomes demonstrate that the CNN network model, when used for sentiment analysis of movie reviews, may produce good [5]. Table 1 compares our outcomes with the related works from many aspects. dataset data splitting, Evaluation measures, and used techniques.

classification effects. While RNN and LSTM recorded an accuracy of 68.64% and 85.32%, respectively, CNN reported an accuracy of 88.22% [4].

In this study, four text classification algorithms (Naïve Bayes, Support Vector Machine, Decision Tree, and Random Forest) are compared using several feature extraction methods, including TF-IDF and Bag of Words. Out of all the experiments, Support Vector Machine and Random Forest yielded the best results of all four algorithms, with respective accuracy of 92.7% and 86.1%

Table 1: Related work

Year	Author Name	DataSet	Splitting	Evaluation measures	Technique used
2017	Ravikumar Patel	WEKAPlatform	26.7% positive, 58.44%neutral, 14.8% negative	Naïve Bayes (Naïve Bayes (accuracy of 90.44 %, Precision 90.8%, Recall 90.4%,F-Measure 90.4%) SVM (accuracy of 99.99%, 100% for Precision & Recall & F-Measure)	Naïve Bayes (NB) and SVM (Support Vector Machine)
2020	Hayder A. Alatabi* , Ayad R. Abbas	Facebook,	50% for positive 50% for Negative	BRDT (Accuracy of 99.625%, Precision 99.9%, Recall 99.4%, F-Measure 99.6%) DT (Accuracy 98.95%, Precision 99.3%, Recall 98.8%, F-Measure 99%)	Bayesian Rough Decision Tree (BRDT). Decision Tree (DT)
		Movie reviews	50% for positive 50% for Negative	BRDT (Accuracy 96.15%, Precision 96.8%, Recall 95.6%, F-Measure 96.2%) DT (Accuracy 95.45%, Precision 95.8%, Recall 95.5%, F-Measure 95.4%)	
2022	Anjali Goswami, Muddada Murali Krishna, Jayavani Vankara	IMDB movie reviews	50% positive 50%negative	Accuracy of 88.22% for CNN Accuracy of 68.64% for RNN Accuracy of 85.32% for LSTM	Convolutional Neural Networks (CNN), RecurrentNeural Networks (RNN) and Long Short-Term Memory (LSTM)
2024	Our Work	Kaggle Platform	50% Positive 50%Negative	LR (Accuracy 95.75%) Naïve Bayes (Accuracy 88.63%) KNN(Accuracy of 80.41%) DT(Accuracy of 80.04%) SVM( Accuracy 96.05%)	Naïve Bayes, Logistic Regression, Decision Tree, Random Forest, K-nearest neighbors(KNN), SVM
			65% Positive 35%Negative	LR (Accuracy 96.20%) Naïve Bayes (Accuracy 92.57%) KNN(Accuracy of 82.77%) DT(Accuracy of 85%) SVM( Accuracy of 96.41%)	

### 3 Methodology

The approach used to analyze ChatGPT tweets is presented in this section. The steps of this study are depicted in Figure 1, which begins with data collection and continues through the classification of each tweet as either

positive or negative using several machine learning algorithms until the evaluation step.

#### 3.1 Data collection

In this study, we have used a variable ChatGPT sentiment analysis available in the Kaggle community<sup>1</sup>. This dataset contains 219294 chatGPT tweets categorized

<sup>1</sup> <https://www.kaggle.com/datasets/charunisa/chatgpt-sentiment-analysis>

as 56012 positive, 107796 negative, and 554870 neutral depicted in Figure 2. After dropping neutral tweets, the dataset content 112022 just positive and negative tweets. In this study we use two versions of the dataset unbalance 65:35 (65% bad (negative) tweets, 35 % good (positive) tweets) depicted in Figure 3, and the balances 50:50 (50 bad. 50% good) depicted in Figure 4.

### 3.2 Preprocessing

Clean data is playing important role in the classification process [6], therefore we applied many preprocessing steps: text normalization, word features, tokenization, stemming, and text Representation because the tweets include a lot of non-useful noise like tags, URLs, and RT (retweet), the preprocessing began with text normalization which involves many subs functions that mentioned in figure 1, which delete occurrences of Re-Twitt, handle the User Tag, Replace emojis with a meaningful text, delete the

occurrences of http:// or https:// and #. In the Word features step, its subs function deals with words in the text, replacing upper capitalization with lower, replacing word repetition with a single occurrence, replacing punctuation repetition with a single occurrence, replacing contractions with their extended forms by using the contractions package., like ("I'd": 'I would' ).In the tokenization step each tweet text is converted into tokens or words, we used custom to remove punctuation and remove stop-words (exclude "not").

Stemming is a natural language processing (NLP) approach that breaks down words into their "stem," or root. To get at a word's fundamental meaning or root, prefixes, suffixes, and other affixes must be removed. There are numerous stemming algorithms; we employ the Snowball stemming algorithm. Finally, in the text representation step convert the sentiment label in the dataset to numerical values 1 for good and 0 for bad. Table 2 depicted applying preprocessing steps

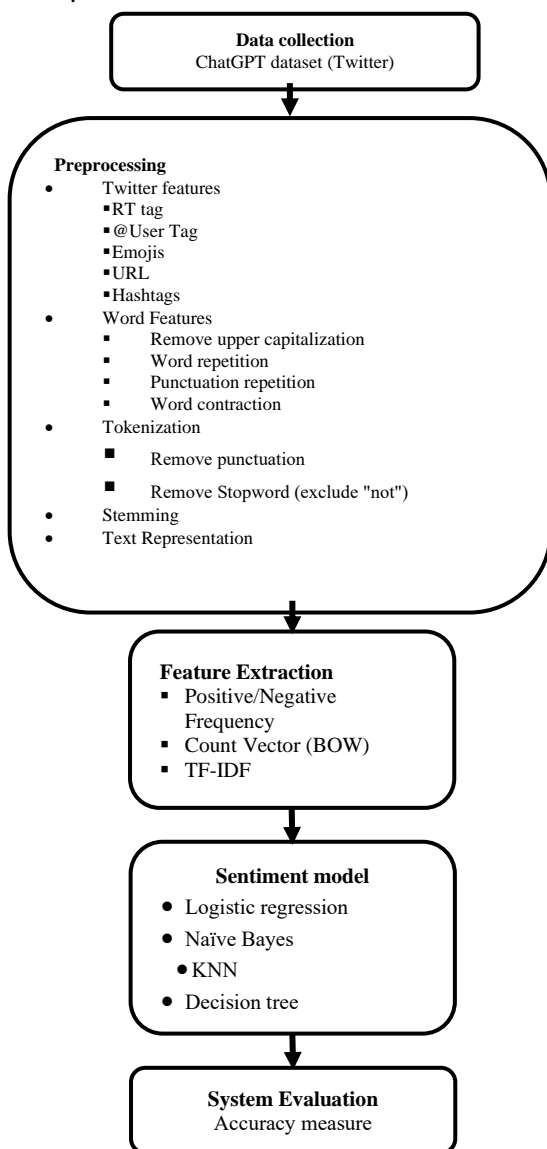


Figure1:The Sentiment analysis system steps

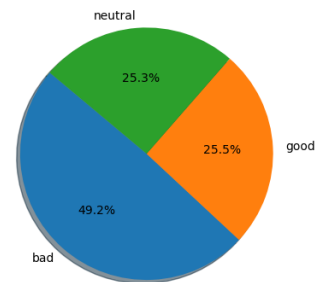


Figure 2: chatGPT dataset

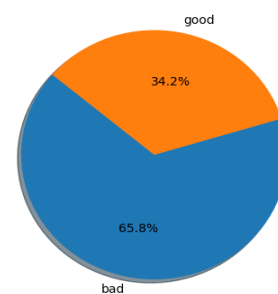


Figure 3: 65:35 dataset

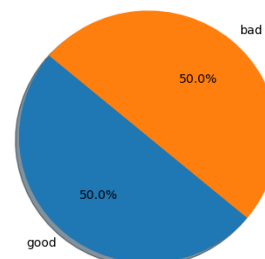


Figure 4: 50:50 dataset

Table 2: Applying preprocessing step

tweets	labels	tokens	Tweet_ sentiment
Try talking with ChatGPT, our new AI system wh...	good	[tri, talk, chatgpt, new, ai, system, optim, d...]	1
THRILLED to share that ChatGPT, our new model...	good	[thrill, share, chatgpt, new, model, optim, di...]	1
Just launched ChatGPT, our new AI system which...	good	[launch, chatgpt, new, ai, system, optim, dial...]	1
ChatGPT coming out strong refusing to help me...	good	[chatgpt, come, strong, refus, help, stalk, so...]	1
#OpenAI just deployed a thing I've been helpin...	good	[deploy, thing, help, build, last, couple, mont...]	1

### 3.3 Feature extraction

In machine learning and data analysis, feature extraction is a key concept. It describes the method of choosing and modifying pertinent data (features) from raw data to be utilized as inputs for machine learning algorithms. To help the machine learning model discover patterns and provide more accurate predictions, feature extraction is designed to present the data in a more condensed and informative manner. we use three types of feature extraction techniques.

First, the positive-negative Frequency is a lexicon-based approach to sentiment analysis. It consists of a list of words with their associated sentiment polarity, such as positive and negative [7][8].

The second is the Count vector or the bag-of-words method is a common technique used in natural language processing (NLP) for feature extraction from text. In this method, a text document is represented as a collection of words, ignoring the order of words and the grammatical structure of sentences [9][10][11].

Finally, TF-IDF (Term Frequency-Inverse Document Frequency) is widely used as a feature extraction technique from text by weighting and ranking the importance of words in a document corpus [9][12].

In TF-IDF, each word in a document corpus is assigned a score that reflects its relevance to a particular document. So, we can calculate the score by multiplying the Inverse Document Frequency (IDF) of any word by the Term Frequency (TF) of this word in the document across the entire text, the resulting TF-IDF score for any word reflects how important this word in the specific document, relative to how common or rare the word is across the entire corpus. Words with higher TF-IDF scores are considered more important or relevant to the document [9][12].

### 3.4 Sentiment model

Our study suggests implementing five popular machine learning classifiers on the numerical representation of tweets after applying feature extraction steps to see which one gives higher accuracy. we select Logistic regression (LR) [13][14], Naïve Bayes [15][16],

NB is particularly useful when dealing with high-dimensional datasets with many features [19][20][24].

K-nearest neighbors (KNN) [16][17], Decision trees [13][18], and Support Vector Machine (SVM) [15][19].

### 3.5 Experimental results

This section will demonstrate the results of five experiments results, LR-Exp, NB-Exp, KNN-Exp, DT-Exp, and finally SVM-Exp, for all these experiments we have used the three feature extraction techniques mentioned in section 3.3. applying in 65:35 and 35:50 datasets. the details of the five experiments will be discussed in an expanded way in the next sections. We use the Grid parameters function in Python, which is known as hyperparameters, to find the best set of values that optimize the model's performance by using evaluation metrics accuracy, precision, recall, and F1.

#### 3.5.1 LR-Exp

Logistic regression (LR) is a statistical method applied for binary classification problems, the goal of LR is to predict binary outcomes for dependent variables depending on one or more independent variables. It is a Supervised Learning Algorithm and it is widely utilized in many fields such as finance, marketing, healthcare, etc. [13][14].

Table 3 displays the results obtained from applying the logistic regression approach to the data sets 50-50 and 35-65 using various feature extraction techniques and evaluation metrics. Where the count vector extraction approach produces the best results.

#### 3.5.2 NB-Exp

Naïve Bayes (NB) is an ML algorithm employed with classification task based on Bayes' theorem which is the more essential Naïve Bayes (NB) is an ML algorithm utilized with classification tasks based on Bayes' theorem which is the more fundamental Naïve Bayes (NB) is an ML algorithm utilized with classification tasks based on Bayes' theorem which is the more fundamental principles in the probability theory [19][20]. To classify a new instance, NB calculates the probability of each class with given values of its features. Then it selects the highest probability class as predicted class.

The outcomes of using the Naive Bayes technique to the dataset 50-50 and 35-65 for positive and negative features are shown in Table 4. Where the count vector extraction approach delivers the most promising outcomes.

### 3.5.3 KNN-Exp

K-nearest neighbors (KNN) is a simple and broadly used machine learning algorithm for classification and regression tasks [20][21].

In KNN, the algorithm classifies a new instance based on the "k" closest labeled instances in the training data, where "k" is a user-defined parameter. This algorithm specifies the length between new instances and training instances based on the nearness or the similarities of their features[17]. For classification tasks, KNN assigns the new instance to the class that appears frequently in the group of its "k" nearest neighbors. Furthermore, KNN in the regression tasks returns the average of target values for its "k" nearest neighbors [17][20].

Table 5 shows the results of using the KNN method by using different methods and parameters for feature extraction on the data set 50-50, and 65-35 for positive and negative. Where the best results are when using the Positive/Negative extraction method.

### 3.5.4 DT-Exp

Decision trees are a popular machine-learning algorithm for both classification and regression tasks [18]. They are based on a tree-like model where the internal nodes represent tests on the input features, the branches represent the outcomes of these tests, and the leaves represent the predicted target values [13][18].

During training, the algorithm recursively partitions the feature space into subsets depending on the value of a single feature at each node. The partitioning criterion aims to maximize the information gain of the resulting subsets, creating a classification that stops when all subgroups conform to the same item or if they are not eligible to continue with an additional section for other subgroups [21][22][23]. Once the tree is constructed, a new instance can be classified or predicted via the traversing of the tree from its root to the leaf node depending on its feature values. So, the output of the algorithm is the class label or the target value associated with the leaf node [13][21][22]. Table 6 shows the results of using the Decision tree method by using different methods and parameters for

feature extraction on the data set 50-50, and 35-65 for positive and negative. Where the best results are when using the Positive/Negative extraction method on a 50-50 data set to split, and the Count Vector extraction method on a 65-35 data set to split.

### 3.5.5 SVM-Exp

It is a machine learning model used for classification and optimization, as it can be used for classification problems. SVM is one of the most popular approaches in machine learning [6][15].

The first step to using SVM is to collect training data. This data must be class known, i.e. the categories of items in the data must be pre-marked. Hence the selection of properties, you may have several properties for each item in the training data. You must choose the appropriate characteristics that contribute to distinguishing the different categories [12]. for sentiment analysis, it would be positive or negative. The training data is divided into two sets; the Training set and the Testing set. The training set is applied for training the model, while the test set is applied for evaluating the model performance [12][24].

In the training phase, the SVM algorithm defines a dividing line between the two classes so that this line is closer to the training samples from both classes and achieves the largest margin. This margin is considered the distance between the dividing line and the nearest training point of each category. where the choice of the kernel function: can be utilized for converting the data to higher dimensional space, allowing for a more complex dividing line to separate classes in the case of nonlinear data. Some examples of kernel functions are linear, radial, basic, and logical [12][25].

There are parameters in SVM like C and Gamma that affect model performance. Adjusting these parameters requires repeated testing and evaluation of the model. After training, the model is tested using the test suite to measure its performance and its ability to correctly classify the new data [12].

SVM is a powerful model in the field of machine learning and is effective in dealing with binary classification and multiclass classification problems [6][15].

Table 7 shows the results of using the Decision tree method by using different methods and parameters for feature extraction on the data set 50-50, and 35-65 for positive and negative. Where the best results are when using the TF-IDF extraction method.

Table 3: Lr-exp results

Model	data set split	Feature extraction method	Model Accuracy	Precision	Recall	F1
Logistic Regression	50_50	50-50 pos / neg	73.41%	0.73	0.73	0.73
	65-35	65-35 pos / neg	73.32%	0.70	0.68	0.69
	50_50	50-50 Count Vector	95.75%	0.96	0.96	0.96
	65-35	65-35 Count Vector	96.20%	0.96	0.95	0.96
	50_50	50-50 TF-IDF	95.70%	0.96	0.96	0.96
	65-35	65-35 TF-IDF	96.12%	0.96	0.95	0.96

Table 4: Naïve Bayes results

Model	data set split	Feature extraction method	Model Accuracy	Precision	Recall	F1
Naïve Bayes MultinomialNB()	50_50	50-50 pos /neg	72.26%	0.73	0.72	0.72
	65-35	65-35 pos /neg	73.83%	0.71	0.72	0.71
	50_50	50-50 Count Vector	87.00%	0.88	0.87	0.87
	65-35	65-35 Count Vector	92.57%	0.92	0.91	0.92
	50_50	50-50 TF-IDF	84.66%	0.86	0.85	0.85
	65-35	65-35 TF-IDF	88.63%	0.89	0.85	0.87

Table 5: KNN results

Model	data set split	extraction method	Model Accuracy	Precision	Recall	F1
KNN	50_50	50-50 pos /neg	80.41%	0.80	0.80	0.80
	65-35	65-35 pos /neg	82.77%	0.81	0.80	0.81
	50_50	50-50 Count Vector	68.74%	0.78	0.68	0.66
	65-35	65-35 Count Vector	75.92%	0.84	0.65	0.66
	50_50	50-50 TF-IDF	75.22%	0.77	0.75	0.75
	65-35	65-35 TF-IDF	72.34%	0.82	0.60	0.58

Table 6: Decision trees results

Model	data set split	extraction method	Model Accuracy	Precision	Recall	F1
DecisionTree	50_50	50-50 pos /neg	80.04%	0.80	0.80	0.80
	65-35	65-35 pos /neg	81.70%	0.80	0.79	0.79
	50_50	50-50 Count Vector	79.74%	0.83	0.80	0.79
	65-35	65-35 Count Vector	84.79%	0.86	0.80	0.82
	50_50	50-50 TF-IDF	79.19%	0.83	0.79	0.79
	65-35	65-35 TF-IDF	84.41%	0.86	0.79	0.81

Table 7: SVM results

Model	dataset split	Feature extraction method	Model Accuracy	Precision	Recall	F1
SVM	50_50	50-50 pos/neg	74.86%	0.73	0.76	0.74
	65-35	65-35 pos/neg	75.01%	0.75	0.74	0.74
	50_50	50-50 Count vector	95.34%	0.96	0.94	0.95
	65-35	65-35 Count vector	95.99%	0.97	0.97	0.97
	50_50	50-50 TF+IDF	96.05%	0.96	0.96	0.96
	65-35	65-35 TF+IDF	96.41%	0.97	0.97	0.97

### 3.6 Evaluation and discussion

In this study, we evaluate the created machine learning classifier using the most used metric accuracy, precision, recall, and F1. The outcomes of various research methodologies are examined.

The graphs in Figure (5,6), summarize the results of our five experiments and show that the SVM algorithm based on the T F-IDF feature extraction method achieves the highest results than others, as the accuracy was 96.05%

and 96.61% for the data model 65:35 and 50-50, respectively, this is due to the SVM algorithm's several advantages. For example, it excels in high-dimensional spaces, which makes it appropriate for issues involving a lot of features. Additionally, by maximizing the margin between classes, the SVM helps to effectively generalize to new or unknown data. SVM is less susceptible to outliers than some other algorithms like KNN, which use the training data for classification directly. It is also less prone to overfitting, especially in high-dimensional spaces than certain other algorithms like Decision Trees. When compared to algorithms like kNN or Decision Trees, Support Vector Machines (SVM) require less tuning of their hyperparameters since they concentrate on the support vectors or the data points that are closer to the decision border and less impacted by outliers. This can improve the model selection procedure and minimize the possibility of hyperparameter tuning-induced overfitting.

Figure 7 shows that the positive-negative frequency achieves the highest accuracy at 82.77% in KNN-Exp and 65-35 datasets and the lowest accuracy at 72.26% with NB-EXP and 50-50 dataset, In binary classification tasks on textual datasets, the KNN and NB algorithms may have inferior accuracy for some reasons, high-dimensional feature spaces, in which each feature represents a distinct word or term, have become prevalent in textual datasets. The affliction of dimensionality can make KNN less

successful in high-dimensional settings, and many features (words or concepts) in textual datasets are limited, which means that their frequencies are either none or extremely low for each instance. This limited data may make it harder to use Naive Bayes to estimate probability effectively. Additionally, the feature representation used for textual data affects both KNN and Naive Bayes. The preprocessing steps (which include tokenization, stemming, and stop-word removal) and feature representation (such as bag-of-words, and TF-IDF) can have a significant impact on how these algorithms operate. Lower accuracy may arise if the feature representation is not well-suited to the properties of the textual data.

Figure 8 shows that when using LR-Exp and the 36-35 dataset, the count vector obtains the highest accuracy of 92.20%, while when using KNN-Exp and the 50-50 dataset, it achieves the lowest accuracy of 68.74%. Figure 9 shows that TF-IDF yields the highest results (96.41%) when SVM-exp is used with the 65-35 dataset because SVM with TF-IDF can effectively handle imbalanced datasets commonly encountered in sentiment analysis tasks. By maximizing the margin between classes, SVM focuses on correctly classifying instances from both classes, even when one class is more prevalent than the other. While KNN-Exp produces the least amount of results (72.34%).

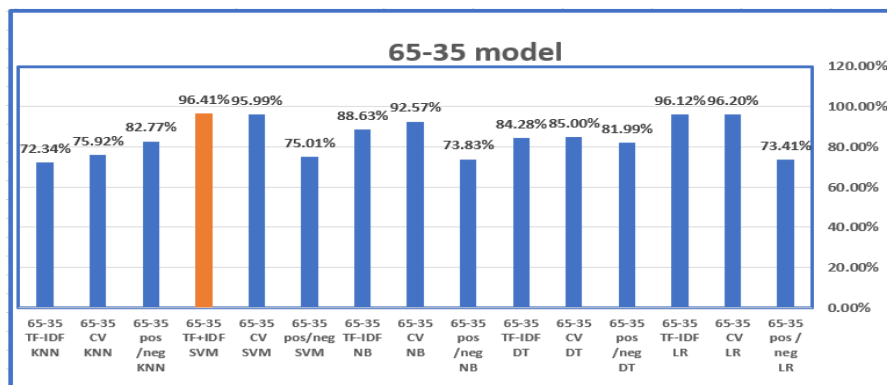


Figure 5: The system evaluation when using a 65-35 data set model

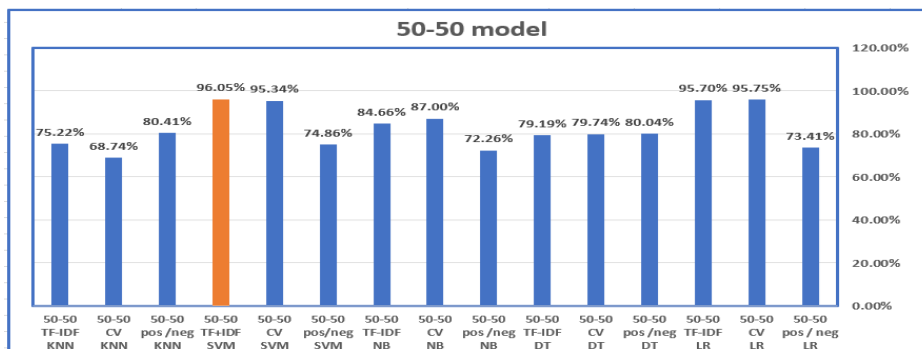


Figure 6: The system evaluation when using a 50-50 data set model

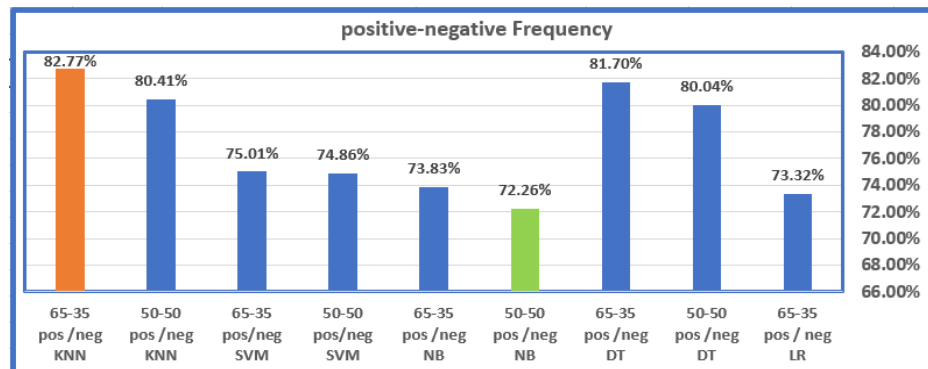


Figure 7: The positive-negative frequency

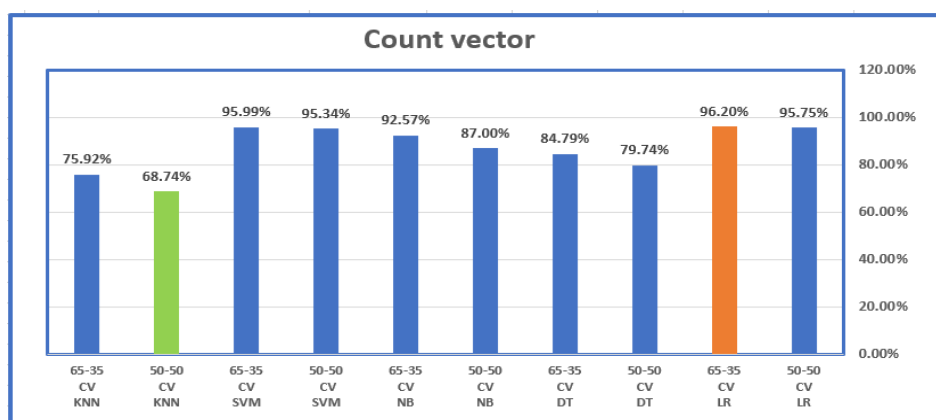


Figure 8: The count vector evaluation

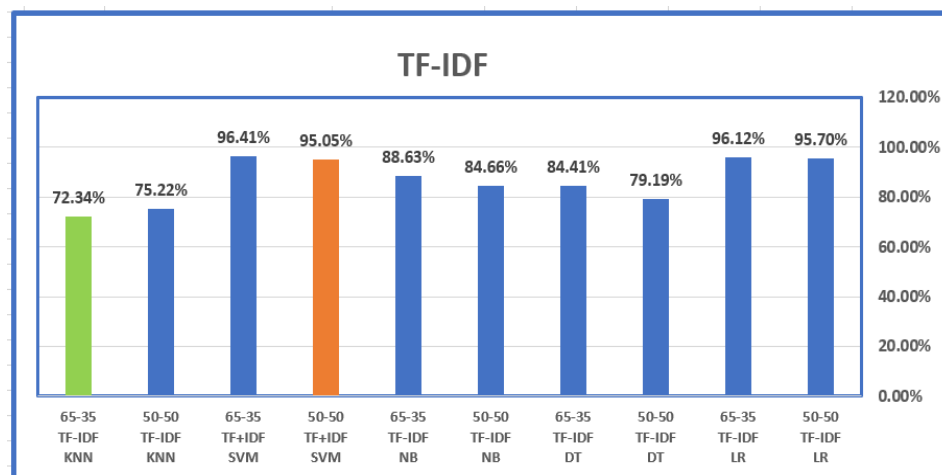


Figure 9: TF-IDF evaluation



When comparing Table 8's results with the highest values of our study's results using the five algorithms with Table 9 results Researchers who worked on the same database using machine learning and deep learning algorithms give the highest accuracy.

Table 7: Related research results

code no	kaggle dataset link	Feature extraction method	Model	Model Accuracy
1	<a href="https://www.kaggle.com/code/svedali110/chatgpt-sentiment-analysis-using-tf-idf">https://www.kaggle.com/code/svedali110/chatgpt-sentiment-analysis-using-tf-idf</a>	CountVector	Naïve Bayes MultinomialNB()	91%
		TF-IDF	Naïve Bayes MultinomialNB()	84%
2	<a href="https://www.kaggle.com/code/sakethkmal/milestone-project-1">https://www.kaggle.com/code/sakethkmal/milestone-project-1</a>	TF-IDF	Logistic Regression	69%
			Decision Tree	73%
			Random Forest	56%
3	<a href="https://www.kaggle.com/code/mafathulhidayah/chatgpt-sentiment-">https://www.kaggle.com/code/mafathulhidayah/chatgpt-sentiment-</a>	CountVector	Naïve Bayes MultinomialNB()	72%
4	<a href="https://www.kaggle.com/code/shubhamf73/chatgpt-sentiment-analysis">https://www.kaggle.com/code/shubhamf73/chatgpt-sentiment-analysis</a>	CountVector	LogisticRegression	85%
			Decision Tree	79%
			Naïve Bayes MultinomialNB()	72%
			LSTM	89%
5	<a href="https://www.kaggle.com/code/hassansanaullah/chatgpt-sentiment-analysis-deep-machine-learning">https://www.kaggle.com/code/hassansanaullah/chatgpt-sentiment-analysis-deep-machine-learning</a>	Converting text to numerical form	LogisticRegression	43%
			Naïve Bayes MultinomialNB()	44%
			Random Forest	55%
			Cnn	88%
6	<a href="https://www.kaggle.com/code/ahmed3sam/chatgpt-sentiment-analysis-using-istm-and-cnn/notebook">https://www.kaggle.com/code/ahmed3sam/chatgpt-sentiment-analysis-using-istm-and-cnn/notebook</a>	Converting text to numerical form	Lstm	89%

Table 8: Our study highest results

Model	Feature extraction method	Model Accuracy
Logistic Regression	CountVector	96.20
Naïve Bayes	CountVector	92.57
KNN	positive/negative	82.77
Decision Tree	CountVector	85.00
SVM	TF-IDF	96.41

### 4 Conclusion

The chatGPT tweets dataset was categorized into positive and negative polarity using machine learning approaches in this study. To accomplish polarity classification, this study analyzed five machine-learning algorithms and three feature extraction strategies. Five trials yielded findings that demonstrated that the TF-IDF outperforms other feature extraction methods while the SVM algorithm outperforms all others in terms of accuracy. Because they will be used as a guide to evaluate subsequent research across various machine learning classifiers, the findings of our study are practically significant.

One of the most significant factors that could lower the accuracy of our study is the use of texts with conflicting or misleading ideas since this could confuse the system or prevent the necessary preprocessing steps from being performed properly. There are many ways to improve the suggested system's accuracy, like using

SMOTE or Tomek links as useful techniques for handling data Imbalance, applying deep learning techniques like RNN or CNN, using to adjust a pre-trained BERT model, testing our approach on datasets from different domains or platforms could help in establishing the generalizability.

### References

- [1] S. Al-Otaibi and A. Al-Rasheed, "A Review and Comparative Analysis of Sentiment Analysis Techniques," *Informatica (Slovenia)*, vol. 46, no. 6. Slovene Society Informatika, pp. 33–44, 2022. doi: 10.31449/inf.v46i6.3991.
- [2] H. O. Ahmad and S. U. Umar, "Sentiment Analysis of Financial Textual data Using Machine Learning and Deep Learning Models," *Informatica (Slovenia)*, vol. 47, no. 5, pp. 153–158, Jan. 2023, doi: 10.31449/inf.v47i5.4673.
- [3] A. Goswami *et al.*, "Sentiment Analysis of Statements on Social Media and Electronic Media Using Machine and Deep Learning Classifiers," *Comput Intell Neurosci*, vol. 2022, 2022, doi: 10.1155/2022/9194031.
- [4] A. Goswami *et al.*, "Sentiment Analysis of Statements on Social Media and Electronic Media Using Machine and Deep Learning Classifiers," *Computational Intelligence and Neuroscience*, vol. 2022, 2022, doi: 10.1155/2022/9194031.
- [5] P. Cen, K. Zhang, and D. Zheng, "Sentiment Analysis Using Deep Learning Approach," *Journal on Artificial Intelligence*, vol. 2, no. 1, pp. 17–27, 2020, doi: 10.32604/jai.2020.010132.
- [6] B. Ondara, S. Waithaka, J. Kandiri, and L. Muchemi, "Machine Learning Techniques, Features, Datasets, and Algorithm Performance Parameters for Sentiment Analysis: A Systematic Review," *Open Journal for Information Technology*, vol. 5, no. 1, pp. 1–16, 2022, doi: 10.32591/coas.ojit.0501.01001o.
- [7] M. Hu and B. Liu, "Mining and summarizing customer reviews," *KDD-2004 - Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 168–177, 2004, doi: 10.1145/1014052.1014073.
- [8] S. Mansour, "Social media analysis of user's responses to terrorism using sentiment analysis and text mining," *Procedia Computer Science*, vol. 140, pp. 95–103, 2018, doi: 10.1016/j.procs.2018.10.297.
- [9] M. Eklund, "Comparing Feature Extraction Methods and Effects of Pre-Processing Methods for Multi-Label Classification of Textual Data," *Degree Project InComputer Science and Engineering Computer Science and Engineering*, pp. 1–50, 2018.

- [10] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of Sentimental Reviews Using Machine Learning Techniques," *Procedia Computer Science*, vol. 57, pp. 821–829, 2015, doi: 10.1016/j.procs.2015.07.523.
- [11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, pp. 1–12, 2013.
- [12] D. Aggarwal, V. Bali, A. Agarwal, K. Poswal, M. Gupta, and A. Gupta, "Sentiment analysis of tweets using supervised machine learning techniques based on term frequency," *Journal of Information Technology Management*, vol. 13, no. 1, pp. 119–141, 2021, doi: 10.22059/jitm.2021.80028.
- [13] M. Hamza and S. Gupta, "a Comparison of Sentimental Analysis Algorithms on Twitter Data Using Machine Learning," 2022.
- [14] A. Bashir and A. B. Musa, "Logistic Regression Classification for Uncertain Data," *Research Journal of Mathematical and Statistical Sciences ISSN*, vol. 2, no. 2, pp. 1–6, 2014.
- [15] S. B. Fatma Jemai, Mohamed Hayouni, "Sentiment Analysis Using Machine Learning Algorithms," in *2021 International Wireless Communications and Mobile Computing (IWCMC)*, Harbin City, China: IEEE. doi: 10.1109/IWCMC51323.2021.9498965.
- [16] S. D. Anjali Gupta, Amita Dhankar, "SENTIMENT ANALYSIS USING MACHINE LEARNING: A REVIEW," *Journal of Emerging Technologies and Innovative Research (JETIR) www.jetir.org (ISSN-2349-5162)*, vol. 5, no. 2, pp. 935–937.
- [17] S. Hota and S. Pathak, "KNN classifier based approach for multi-class sentiment analysis of twitter data," *International Journal of Engineering and Technology(UAE)*, vol. 7, no. 3, pp. 1372–1375, 2018, doi: 10.14419/ijet.v7i3.12656.
- [18] Meenu and Sunila Godara, "Sentiment Analysis using Decision Tree," <http://www.csjournals.com/IJEE/PDF11-1/154.%20Sunila.pdf>, vol. 11, no. 1, pp. 965–970, 2019.
- [19] S. Kethavath, "Classification of Sentiment Analysis on Tweets using Machine Learning Techniques," National Institute of Technology Rourkela Rourkela-769 008, Odisha, India, 2015.
- [20] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: A methodology review," *Journal of Biomedical Informatics*, vol. 35, no. 5–6, pp. 352–359, 2002, doi: 10.1016/S1532-0464(03)00034-0.
- [21] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," *Informatica 31*, vol. 31, pp. 249–268.
- [22] S. Kasthuri and D. A. N. Jebaseeli, "An efficient Decision Tree Algorithm for analyzing the Twitter Sentiment Analysis," *Journal of Critical Reviews*, vol. 7, no. 4, pp. 1010–1018, 2020.
- [23] A. S. and C. R. Bharathi, "Sentiment Classification using Decision Tree Based Feature Selection Sentiment Classification using Decision Tree Based Feature Selection," *Ijcta*, vol. 9, no. January, pp. 419–425, 2017.
- [24] P. A. Grana, "Sentiment Analysis of Text Using Machine Learning Models," *International Research Journal of Modernization in Engineering Technology and Science*, no. 05, pp. 2582–5208, 2022.
- [25] C. A. A. Kaestner, "Support Vector Machines and Kernel Functions for Text Processing," *Revista de Informática Teórica e Aplicada*, vol. 20, no. 3, p. 130, 2013, doi: 10.22456/2175-2745.39702.