International Journal of Advanced and Applied Sciences, 10(10) 2023, Pages: 94-102

Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

Efficient social media sentiment analysis using confidence interval-based classification of online product brands

Tawfik Guesmi [1,*], Fawaz Al-Janfawi [1], Ramzi Guesmi [2], Mansoor Alturki [1]

[1]*Department of Electrical Engineering, College of Engineering, University of Ha'il, Ha'il, Saudi Arabia*
[2]*LETI-Laboratory, National Engineering School of Sfax, University of Sfax, Soukra Sfax, Tunisia*

ARTICLE INFO

Article history:
Received 15 April 2023
Received in revised form
16 September 2023
Accepted 21 September 2023

Keywords:
Machine learning
SVM
NLP
Sentiment analysis
Confidence intervals

ABSTRACT

This paper presents an efficient method for categorizing the sentiments of Internet users, with a focus on social media users, using a confidence interval to estimate the reliability of sentiment predictions. The classification is based on the sentiments expressed in their posts, which are divided into positive, negative, and neutral categories. The paper presents an analysis table that analyzes sentiments and opinions about online product brands. The process includes two steps: 1) analyzing sentiments from text data using machine learning techniques, and 2) describing a five-step sentiment and opinion classification process that includes data collection, preprocessing, algorithm application, validation, and visualization. The proposed solution is implemented using Python, along with the scikit-learn, NumPy, pandas, and Dash libraries, and leverages the use of confidence intervals to assess the accuracy and reliability of the sentiment analysis model.

1. Introduction

In recent years, the amount of data circulating on the web has increased significantly. What has caught the attention of researchers is that users are sharing their feelings, opinions, or views almost everywhere on the internet. They come in different forms of expression including comments and questions. Researchers are faced with huge amounts of information which takes various forms Alone textual form accounts for 80% of this mass of information. However, the sheer quantity of data will be of no use if we do not explore it using methods aimed at extracting meaningful information (Zhao et al., 2015). Artificial intelligence is increasingly involved in all areas of human life. AI plays a crucial role in the customization of technology to meet the needs of humans. With the use of emotion recognition and sentiment analysis algorithms, AI is able to analyze human emotions and behavior patterns' leading to more effective adaptation of technology. Examples include personalized content, speech recognition, and self-driving cars. Nevertheless, it's crucial to approach the development and use of AI with caution, as it may also introduce biases and ethical issues (Liu, 2012; Sánchez et al., 2020; Wong and Farooq, 2020; Alsaif et al., 2022). The primary goal of AI techniques is to empower machines with the ability to recognize emotions. In this study, we will employ popular machine learning models including Support Vector Machine (SVM), Decision Tree (DT), Multilayer Perceptron (MLP), and Convolutional Neural Networks (CNN). Machine Learning (ML) has demonstrated indisputable effectiveness in detecting emotions. In recent years, deep learning has become a very attractive field offering a wide range of possibilities for innovation. It has contributed effectively to the field of detecting the emotions of users of social networks such as Facebook and Twitter. As a result, research works related to this field have increased in number and the quality of their results (Pradhan et al., 2016; Kharde and Sonawane, 2016; Sandoval-Almazan and Valle-Cruz, 2018). A convolutional neural network (CNN) is one of the most well-known deep learning techniques. CNNs are designed with the assumption that they will be 36 processing images and, as a result, they will allow the networks to operate more efficiently and handle larger images. Sentiment analysis, a subset of natural language processing, endeavors to classify 38 emotions conveyed through text and is commonly referred to as opinion mining. In recent years, several research works in the field have been published. For instance, Hatzivassiloglou and McKeown (1997) were among the first to study

* Corresponding Author.
Email Address: tawfik.guesmi@istmt.rnu.tn (T. Guesmi)
https://doi.org/10.21833/ijaas.2023.10.011
Corresponding author's ORCID profile:
https://orcid.org/0000-0001-8221-2610

sentiment and opinion classification. The task of categorizing documents based on sentiment was first thoroughly discussed by Pang et al. (2002) to distinguish between positive and negative reviews. The study utilized movie reviews as its dataset. Three machine learning techniques were applied in the research: Naive Bayes, Maximum Entropy Classification, and Support Vector Machines. Recently, Kaur et al. (2016) developed an effective solution to determine the influence of a user's tweets on the fluctuation of the stock prices of Samsung Electronics Ltd. The paper's objective was to study positive, negative, and neutral tweets. AI techniques were used to perform sentiment analysis in Cambria et al. (2018). In this study, the authors employed a long short-term memory (LSTM) network (Hochreiter and Schmidhuber, 1997) to construct a novel representation of knowledge for sentiment analysis, referred to as SenticNet 5. Basiri et al. (2021), developed a solution for sentiment analysis based on Deep Neural Network (DNN). This solution is considered to be a new deep architecture for sentiment analysis and it was evaluated on two types of social media texts: long reviews and short tweets. Some weaknesses and problems of the supervised methods such as the need for a large amount of training data are listed in Šperková (2018). Taking these problems into consideration, solutions based on unsupervised methods were proposed by Liu (2017), Basiri et al. (2017), and Basiri and Kabiri (2017). The authors have shown that their methods are simple, fast, and scalable. In other research papers, the authors proposed to take advantage of both supervised and unsupervised methods. Zhang et al. (2011) proposed a new entity-level sentiment analysis method for Twitter. High precision is guaranteed by this method but recall should be improved. Mudinas et al. (2012) outlined a hybrid methodology in their paper, pSenti, a concept-level sentiment analysis system that merges lexicon-based and learning-based techniques. The goal is to benefit from the stability and readability of the lexicon and the high accuracy of supervised learning methods. Several contributions to Twitter sentiment analysis were detailed and presented by Ghiassi and Lee (2018). To achieve a more precise evaluation of tweet sentiments, the authors created a Twitter-specific lexicon set for sentiment analysis. Chikersal et al. (2015) proposed to improve supervised learning methods by benefiting from linguistic rules and computing resources. The proposed solution is tested on two Twitter corpora in order to illustrate its effectiveness.

The use of confidence intervals in sentiment analysis can help to provide a quantitative measure of the reliability and uncertainty of sentiment predictions made by a machine learning model. The confidence interval represents the range of values within which the true sentiment score is likely to fall with a certain level of confidence (Lenc and Hercig, 2016). By examining the confidence intervals of sentiment predictions, we can identify areas where the model may be less reliable or certain in its predictions. For example, a wide confidence interval may suggest that the model is uncertain or lacks confidence in predicting a particular sentiment, indicating that more data or model improvements may be needed. On the other hand, a narrow confidence interval may indicate that the model is highly confident in its prediction, suggesting that the model is well-trained and the prediction is likely to be accurate. Thus, confidence intervals can help in interpreting the results of sentiment analysis and identifying areas for improvement (Tackstrom and McDonald, 2011). Additionally, confidence intervals can also help in decision-making processes, where it is important to have a level of confidence in the sentiment prediction. For example, in a marketing campaign, knowing the confidence interval of sentiment predictions can help in identifying which products or messages are likely to be well-received by the target audience, thus improving the success of the campaign (Mahyoob et al., 2022). The main objectives of this paper are:

- Detecting emotions by using Machine Learning.
- Identifying the models that perform best according to the experimental results.
- By quantifying the uncertainty of sentiment analysis results using confidence intervals, we can assess the reliability and accuracy of the sentiment analysis model. The confidence interval indicates the range of sentiment scores within which the true sentiment score is likely to fall with a certain level of confidence.

The rest of this paper is divided into 3 sections: In the next section, we introduce the basic theory of the proposed algorithm. Section 3 introduces the methodology of the proposed solution. Experimental results are presented in Section 4. The conclusions and future work directions are mentioned in the last section.

2. Preliminaries

Machine learning algorithms are essential tools for data analysis and prediction tasks. Naive Bayes is a popular probabilistic algorithm used for classification tasks that works on the basis of Bayes' theorem. Logistic regression is a linear algorithm that models the probability of a binary response variable given one or more explanatory variables. Random Forest, an ensemble algorithm, is a combination of decision trees that creates multiple models and combines their predictions to improve accuracy and robustness. Support Vector Machine (SVM) is a binary classification algorithm that finds a hyperplane that separates the data points into different classes. Decision Tree (DT) is a non-parametric algorithm that creates a tree-like model of decisions and their possible consequences. All of these algorithms have different strengths and weaknesses, making them suitable for different types of data and applications. These different models are grouped together in Fig. 1.

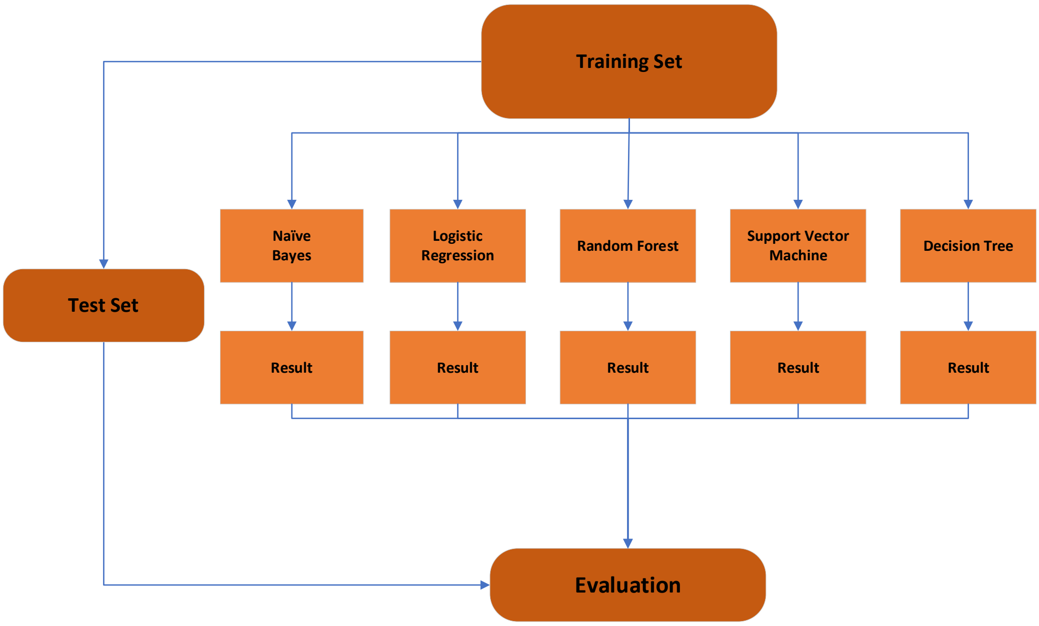


Fig. 1: Different models

Naive Bayes: Naive Bayes is a probabilistic algorithm that uses the Bayes theorem to calculate the probability of a given class given the features. It is often used in text classification, spam filtering, and sentiment analysis. The "naive" assumption refers to the fact that it assumes independence between features, which helps in handling high-dimensional data (John and Langley, 2013; Friedman et al., 1997). It makes the assumption that the probabilities of the various events are 100% independent. Given a set of features x1, x2, . . . , xn and a set of classes C, we draw a model of the value of P (Ck|x1, x2, . . . , xn). In Naive Bayes, the class with the highest probability is selected as the prediction. The algorithm calculates the probability of each class given the features, and the class with the highest probability is considered the most likely class. This is known as the Maximum A Posteriori (MAP) estimate.

$$C_{predicted} = argmax\ P(C_k|x_1, x_2, \cdots, x_n) \quad (1)$$

Logistic regression: Logistic regression is a binary classification algorithm that predicts the probability of an event occurring. The predicted probability is then transformed into a binary outcome using a threshold, usually set at 0.5. Logistic regression models the relationship between the dependent variable and one or more independent variables using the logistic function (sigmoid curve), which outputs a probability value between 0 and 1. The model of logistic regression is given by:

$$\begin{cases} Output: 0\ or\ 1 \\ Hypothesis: Z = WX + B \\ h(x) = sigmoid(Z) \end{cases} \quad (2)$$

where, $h(x)$ is the predicted probability of the positive class, W and B are the model parameters (weights and bias), and X is the input feature vector. The function $sigmoid(Z)$ maps the weighted sum of features, Z, to a value between 0 and 1, which represents the predicted probability of the positive class.

Random forest: The Random Forest algorithm is a type of ensemble machine learning technique that employs several decision trees for making predictions. It works by training multiple decision trees on different random subsets of the training data, and then aggregating their predictions to obtain a final prediction for new data. This results in a "forest" of decision trees, where the final prediction is the average or majority vote of the individual tree predictions. Random Forest is known to handle large datasets efficiently and also effectively reduces overfitting compared to a single decision tree (Witten et al., 2005).

Support vector machine (SVM): The purpose of SVM is to classify data into distinct categories by locating the optimal hyperplane that maximizes the margin between the categories. The margin refers to the distance between the hyperplane and the closest data points from each class, which are referred to as support vectors. SVM has been widely used in many applications due to its ability to handle non-linearly separable data by transforming the input features into a higher dimensional space through the use of kernels. Thus, it aims to identify the optimal boundary between two classes of data by constructing a hyperplane that maximally separates them. The method employs a technique called structural risk minimization to create this boundary, effectively separating positive examples from negative ones (Pisner and Schnyer, 2020). SVM maximizes the margin between support vectors because separating all classes is necessary. SVMs are widely used in various fields including computer vision, natural language processing, bioinformatics, and finance. Some common applications of SVMs include text classification, image classification, intrusion detection, and gene expression analysis. Overall, SVMs are popular due to their ability to handle high-dimensional and complex data, their robustness to overfitting, and their ability to perform well in multi-class classification problems

Decision tree (DT): A Decision Tree is a type of predictive model that categorizes instances by evaluating the values of certain features. The tree consists of nodes, each representing a feature in the instance to be classified, with branches indicating the possible values that the feature can take. The process of classification begins at the root node and continues based on the values of the features. In data mining and machine learning, Decision Tree Learning is utilized to create such trees, which are also known as classification trees or regression trees, and they aim to make predictions about the target value of an item based on its observable characteristics (Hastie et al., 2009). Decision tree classifiers often utilize post-pruning methods that assess the efficiency of the trees as they are trimmed by evaluating them with a validation set. The pruning process involves removing nodes and assigning the most frequently occurring class of the training instances that belong to it (Priyam et al., 2013).

3. Methodology

This paper outlines the proposed approach for collecting public opinions on a phone product. The process begins by gathering data from Twitter and includes several essential stages, such as preparing the data for machine analysis as outlined (Zhao et al., 2015). We then calculate and classify the sentiment of the tweets, followed by visualizing performance metrics and predicting tweet sentiments using machine learning algorithms. In the final stage, a dashboard can be used to explore the sentiment of various factors. Fig. 2 illustrates the flowchart of our methodology.

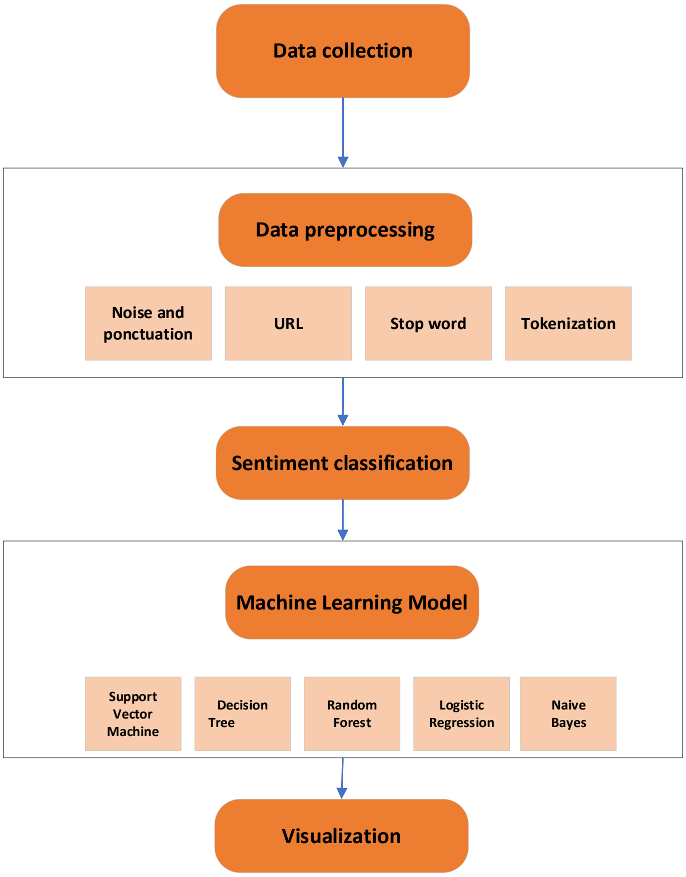


Fig. 2: Flowchart of the new approach

Data collection: We utilized the Tweepy API to gather public opinion on phone products by collecting hashtags on Twitter and its top trends. Our Twitter account was linked to a Tweepy API account, which we set up. The Tweepy API accepts parameters and returns data from the Twitter account. The retrieved tweets were stored in the database with the following information: twitter_id, hashtag, and tweet_text. We recall that Tweepy is a Python library for accessing the Twitter API. It provides a convenient way for Python developers to access and work with tweets and other data from the Twitter platform. It abstracts the complexities of making requests to the Twitter API, handling authentication, and parsing responses, allowing developers to focus on building their application logic. With Tweepy, developers can perform tasks such as retrieving tweets, posting tweets, retrieving user information, and more. Additionally, Tweepy provides support for streaming tweets in real-time and offers a convenient way to work with Twitter's rate limits and pagination

Data pre-processing: The raw corpus that we compiled from Twitter includes a number of flaws, like insufficient space, invalid characters, improper structure, etc. Therefore, before applying a classification algorithm to the corpus, we first pre-process data to clean it up and put it in the proper format. Some pre-processing steps include:

- Noise and Punctuation Removal: a step in the data pre-processing phase of natural language processing and text analysis. It involves removing non-informative characters and symbols such as punctuation marks, special characters, and digits that do not carry any meaning in the analysis. The aim is to remove any irrelevant information and make the data more suitable for analysis by machine learning algorithms. The process of noise and punctuation removal helps to increase the accuracy of the results by reducing the dimensionality of the data and allowing the algorithms to focus on the meaningful content of the text.
- URL Removal: is a pre-processing step to remove uniform resource locators (URLs), also known as web links, from the text corpus. URLs can have little to no impact on the sentiment expressed in the text and can also affect the performance of machine learning algorithms, so it is common to remove them. The removal process can involve using regular expressions or string manipulation techniques to identify and remove the URLs from the text. This helps to clean up the text corpus and improve the quality of the data that is used as input to the sentiment analysis model.
- Apply stop word filtering: a pre-processing step in NLP that involves removing common, unimportant words (known as "stop words") from a text corpus. Examples of stop words include "a," "an," "the," "and," etc. Stop words are removed because they add noise to the data and do not contribute much to the sentiment or meaning of a text. By removing stop words, the focus is on the meaningful words, which can help improve the accuracy and performance of NLP models
- Tokenization: the process of splitting text into smaller units, called tokens. In NLP and text mining, tokens are often words or phrases. The goal of tokenization is to convert a sentence or a document into a sequence of meaningful tokens that can be used for further analysis and processing. Tokenization helps to break down text into its constituent parts and remove irrelevant information, making it easier to analyze the text and extract relevant information. For example, after tokenization, a sentence such as "The cat chased the mouse." becomes a sequence of tokens: ["The," "cat," "chased," "the," "mouse"].

Polarity calculation and sentiment analysis: Sentiment analysis can uncover valuable insights from social media platforms by identifying emotions or views from vast amounts of unstructured data. The analysis categorizes sentiments into three polarity classes: negative, neutral, and positive. Each tweet is assigned a score ranging from -1 to 1, where negative scores indicate a negative sentiment, positive scores denote a positive emotion, and a score of zero signifies a neutral feeling. Tweets are assigned a subjectivity score based on the level of subjectivity or objectivity they convey. The score ranges from 0 to 1, where values close to 0 indicate objectivity and values close to 1 signify subjectivity. To accurately determine tweet polarity and subjectivity, we utilized TextBlob as the polarity and subjectivity calculator. Table 1 shows the top positive and negative tweets from TextBlob.

Table 1: Top positive and negative tweets from TextBlob

Tweet	Sentiment score
trethansw xe2 x80 x99t bloat ware company add android hate Samsung	Negative
fromkorea5 samsung galaxy z fold 2 whitest one dome screen protector http co 0gnb82hjhr the digidigest nthe world first tempere xe2 x80 xa6	Positive
perkurowski l_lucullus janineyve lars9596 helgy2 stewartbutton jbhearn great lakes forex bulk biker xe2 x80 xa6 http co p3wgyihzbp	Neutral

Modeling: This step presents the sentiment classification and prediction of tweets using five popular supervised machine learning algorithms: SVM, Decision Tree, Random Forest, Naive Bayes, and Logistic Regression. To train the models, the text data must first be prepared for the computer to understand by converting it into numerical format. This manual process involves transforming the raw text into numerical form (Liu, 2012). In fact, the manual process of converting raw text into numerical form involves techniques such as tokenization, stemming, and vectorization, where the text is broken down into individual words or phrases, normalized, and represented as numerical data, such as 215-word counts or word embeddings, that can be processed by machine learning algorithms.

4. Experiment results

For analyzing tweet sentiments, we are employing scikit-learn, a Python-based machine learning library, to fit five different algorithms (SVM, Decision Tree, Random Forest, Naive Bayes, and Logistic Regression) to our tweet datasets. By fitting these algorithms to the tweet datasets, we are training them to recognize patterns in the text data and predict the sentiment of a given tweet. The process of fitting is essential for developing a reliable sentiment analysis model that can accurately classify tweets as positive, negative, or neutral. By accurately predicting tweet sentiments, businesses, and organizations can gain valuable insights into how their brand is perceived on social media and make informed decisions to improve their online presence.

4.1. Machine learning models evaluation

The training accuracy of a model is a measure of how well the model is able to predict the correct labels for the training data. The validation accuracy is a measure of how well the model is able to predict the correct labels for the validation data, which is a subset of the data that is held out from the training process and used to evaluate the model's performance. In general, the training accuracy of a model will be higher than the validation accuracy, because the model has seen the training data during the training process and has learned to predict the labels for those data points accurately. The validation accuracy is a more realistic measure of the model's performance because it reflects the model's ability to generalize to unseen data. In order to determine the performance of the proposed method, we examined the accuracy, precision, recall, and F1 score. Accuracy is the proportion of correctly classified instances (True Positives and True Negatives) out of all instances. It is computed as follows:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad (3)$$

where, true positives (TP) are the number of instances where the model correctly predicted the positive class, true negatives (TN) are the number of instances where the model correctly predicted the negative class, False Positives (FP) are the number of instances where the model incorrectly predicted the positive class and false negatives (FN) are the number of instances where the model incorrectly predicted the negative class. Precision is the proportion of correctly classified positive instances out of all positive instances predicted by the model. It is calculated as follows.

$$Precision = TP/(TP + FP) \quad (4)$$

Recall is the proportion of correctly classified positive instances out of all actual positive instances. It is calculated as follows.

$$Recall = TP/(TP + FN) \quad (5)$$

F_{1Score} is the harmonic mean of precision and recall. It is calculated as follows.

$$F_{1score} = (2TP)/(2TP + (FP + FN)) \quad (6)$$

Table 2 represents the precision, recall, and F_{1Score} for each class of different models. It shows that the SVM model performs better than other ones. Fig. 3 shows sentiments per month in 2021. The developed dashboard (Fig. 4) displayed a visualization of Huawei tweet sentiments. The results shown in Figs. 3 and 4 provide a foundational understanding of the sentiment landscape that underlies the social discourse captured by these tweets. The prevalence of neutral sentiment (42.50%) suggests that a substantial portion of the content can be characterized as informative or factual in nature, possibly encompassing news updates, announcements, or general information sharing. The nearly equal distribution of positive sentiment (41.70%) highlights the presence of favorable expressions, which may encompass expressions of joy, enthusiasm, support, or approval. Meanwhile, the relatively smaller percentage of negative sentiment (15.80%) indicates that expressions of discontent, dissatisfaction, or criticism are less common. These findings serve as a crucial starting point for further exploration into the factors that shape sentiment patterns within this dataset.

Table 2: Comparison between different models

Model	Precision	Recall	F1 score
SVM	0.952	0.951	0.951
Naive Bayes	0.912	0.911	0.910
Decision tree	0.932	0.932	0.931
Random forest	0.935	0.929	0.928
Logistic regression	0.948	0.947	0.947

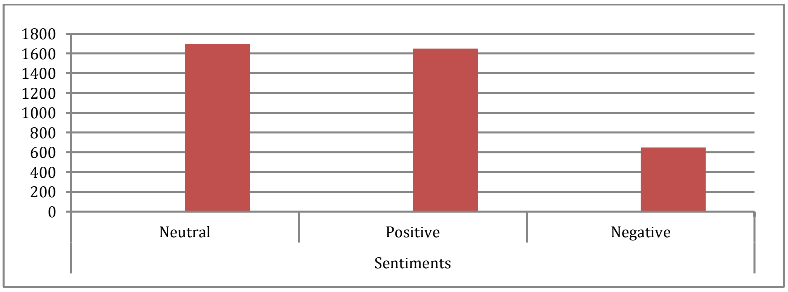


Fig. 3: Sentiments in April 2021

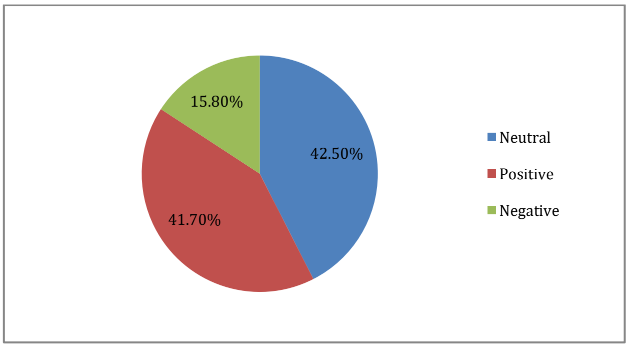


Fig. 4: Huawei tweet sentiments

4.2. Confidence intervals

Confidence intervals are statistical measures that provide a range of values within which the true value of a population parameter is believed to lie with a certain level of confidence. In the context of machine learning, confidence intervals can be used to estimate the uncertainty of a prediction made by a model. For example, let's say we have trained a sentiment analysis model to predict the sentiment of customer reviews. We can use the model to predict the sentiment of new reviews, and then calculate a confidence interval for each prediction. This confidence interval will give us a range of values within which we can be confident that the true sentiment score of the review lies, based on the performance of the model on the training data. The level of confidence associated with the confidence interval is typically expressed as a percentage, such as 95% or 99%. A 95% confidence interval means that if we repeated the same experiment (predicting the sentiment of reviews) many times, 95% of the resulting confidence intervals would contain the true sentiment score. By using confidence intervals to measure the uncertainty of predictions, we can get a better sense of the reliability of the model's output. This information can be used to make more informed decisions about the accuracy of the predictions and to identify areas where the model may need to be improved (Boukes et al., 2020; Chu and Roy, 2017). Algorithm 1 is used to calculate the confidence interval for a set of sentiment scores, given a confidence level. Take, for example, the following sentences. "What a great airline, the trip was a pleasure!," "My issue was quickly resolved after calling customer support. Thanks!," "What the hell! My flight was canceled again. This sucks!," "Service was awful. I'll never fly with you again," "You stupid lost my luggage. Never again!," "I have mixed feelings about airlines. I don't know what I think."

Algorithm 1. Confidence interval

Require: *sentiment_scores, confidence_level*
1: ***function*** *Calculate_confidence_interval (sentiment_scores, confidence_level)*
2: $n \leftarrow$ *number of samples in sentiment_scores*
3: $\bar{x} \leftarrow$ *mean of sentiment_scores*
4: $s \leftarrow$ *standard deviation of sentiment_scores*
5: $z \leftarrow$ *z-score for confidence_level*
6: $m \leftarrow z \times \frac{s}{\sqrt{n}}$
7: $l \leftarrow \bar{x} - m$
8: $u \leftarrow \bar{x} + m$
9: ***return*** *[l, u]*
10: ***end function***

where, sentiment_scores: The input array or list of sentiment scores for which the confidence interval is to be calculated. confidence_level: The desired level of confidence for the calculated interval, expressed as a percentage. n: The number of samples in the input sentiment scores. $\bar{x}$: The mean of the input sentiment scores. s: The standard deviation of the input sentiment scores. z: The z-score for the desired level of confidence, which can be obtained from a standard normal distribution table or calculated using statistical software. m: The margin of error for the calculated confidence interval. l: The lower bound of the calculated confidence interval. u: The upper bound of the calculated confidence interval.

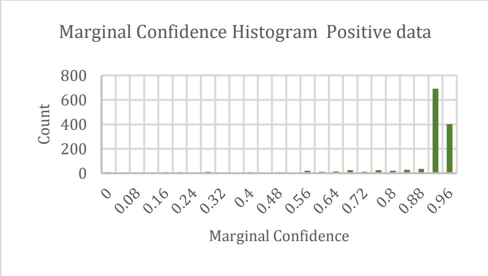
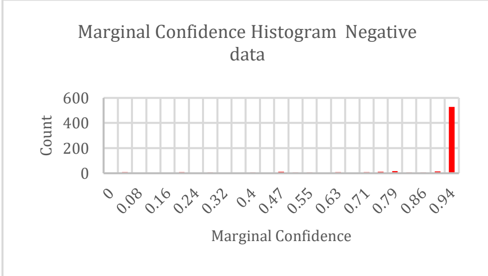
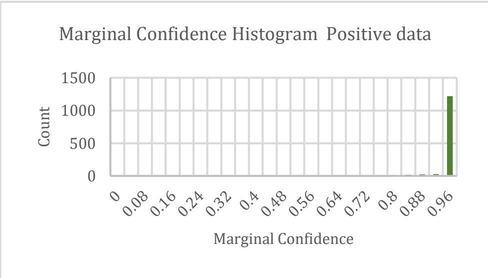
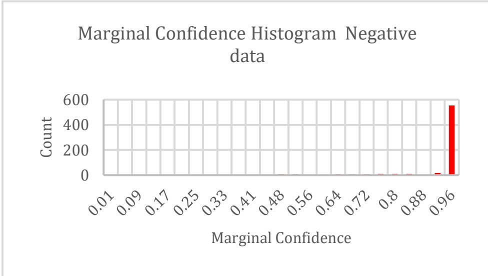
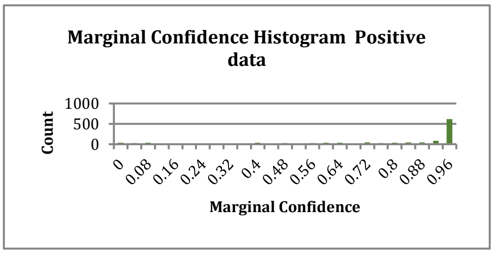
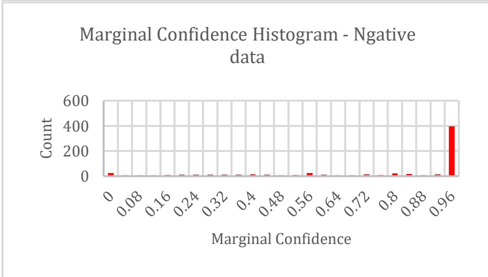
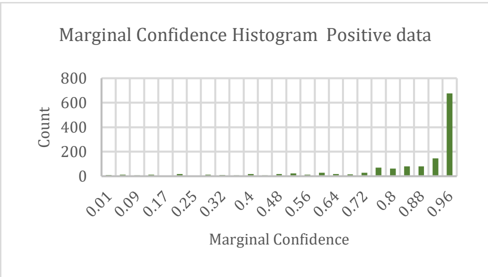
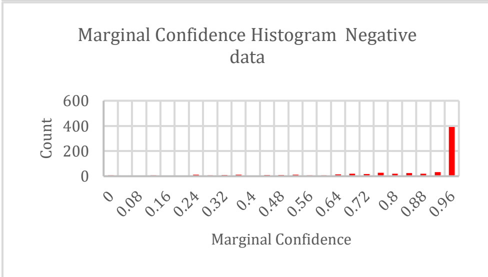
4.3. Marginal confidence histogram

A histogram can be used to display the distribution of confidence intervals. It shows the frequency of the intervals that occur within a certain range. This can be useful in understanding the degree of uncertainty in a prediction and assessing the reliability of the results. The x-axis of the histogram would represent the range of possible sentiment scores, while the y-axis would represent the frequency or count of the calculated confidence intervals. The shape of the histogram would provide insight into the variability and distribution of the sentiment scores, as well as the reliability and precision of the sentiment analysis model. For example, a histogram showing a narrow distribution of confidence intervals clustered around a mean sentiment score would indicate that the sentiment analysis model was accurate and reliable. On the other hand, a histogram showing a wide distribution of confidence intervals with a large range of sentiment scores would indicate that the sentiment analysis model was less accurate and more variable. Histograms, in Table 3, show the marginal confidence using different models of machine learning. A large spread of confidence levels for a particular sentiment prediction may indicate that the model is uncertain or unreliable in making predictions for that sentiment. On the other hand, a narrow spread of confidence levels for a particular sentiment prediction may indicate that the model is confident in its predictions for that sentiment. Thus, the interpretation of confidence intervals in sentiment analysis can help to assess the reliability and robustness of the sentiment analysis model.

5. Conclusion

Now, we have the chance to express our views, opinions, and thoughts through digital media. Social networks are now widely used for this as well as for spreading ideas and creating personal perspectives. In this paper, we have analyzed methods for comprehending people's sentimentality by developing a sentiment analysis model. We have found that the majority of tweets are neutral, followed closely by positive feelings and then negative sentiments. Five popular machine learning models were applied for prediction. The experiment and evaluation's findings indicate that the Support Vector Machine model outperforms other classifiers. We have achieved an accuracy rate of 95% on sentiment detection from tweets. In the future, we plan to improve our results by utilizing deep neural networks such as LSTM and CNN. Additionally, we will investigate alternative word embedding methods such as Bert and Elmo.

Table 3: Marginal confidence using different models

Model	Positive	Negative
SVM		
Naive Bayes		
Random forest		
Logistic regression		

Acknowledgment

This research has been funded by the Scientific Research Deanship at the University of Ha'il–Saudi Arabia through project number GR-22 089.

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Alsaif H, Guesmi R, Alshammari BM, Hamrouni T, Guesmi T, Alzamil A, and Belguesmi L (2022). A novel data augmentation-based brain tumor detection using convolutional neural network. Applied Sciences, 12(8): 3773. https://doi.org/10.3390/app12083773

Basiri ME and Kabiri A (2017). Translation is not enough: Comparing lexicon-based methods for sentiment analysis in Persian. In the 2017 International Symposium on Computer Science and Software Engineering Conference (CSSE), IEEE, Shiraz, Iran: 36-41. https://doi.org/10.1109/CSICSSE.2017.8320114

Basiri ME, Ghasem-Aghaee N, and Reza A (2017). Lexicon-based sentiment analysis in Persian. Current and Future Developments in Artificial Intelligence, 30: 154-183. https://doi.org/10.2174/9781681085029117010009

Basiri ME, Nemati S, Abdar M, Cambria E, and Acharya UR (2021). ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis. Future Generation Computer Systems, 115: 279-294. https://doi.org/10.1016/j.future.2020.08.005

Boukes M, Van de Velde B, Araujo T, and Vliegenthart R (2020). What's the tone? Easy doesn't do it: Analyzing performance and agreement between off-the-shelf sentiment analysis tools. Communication Methods and Measures, 14(2): 83-104. https://doi.org/10.1080/19312458.2019.1671966

Cambria E, Poria S, Hazarika D, and Kwok K (2018). SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. Proceedings of the AAAI

Conference on Artificial Intelligence, New Orleans, USA, 32(1). https://doi.org/10.1609/aaai.v32i1.11559

Chikersal P, Poria S, Cambria E, Gelbukh A, and Siong CE (2015). Modelling public sentiment in Twitter: Using linguistic patterns to enhance supervised learning. Computational Linguistics and Intelligent Text Processing: 16th International Conference, Springer International Publishing, Cairo, Egypt: 49-65. https://doi.org/10.1007/978-3-319-18117-2_4

Chu E and Roy D (2017). Audio-visual sentiment analysis for learning emotional arcs in movies. In the 2017 IEEE International Conference on Data Mining (ICDM), IEEE, New Orleans, USA: 829-834. https://doi.org/10.1109/ICDM.2017.100 **PMCid:PMC5412413**

Friedman N, Geiger D, and Goldszmidt M (1997). Bayesian network classifiers. Machine Learning, 29: 131-163. https://doi.org/10.1023/A:1007465528199

Ghiassi M and Lee S (2018). A domain transferable lexicon set for Twitter sentiment analysis using a supervised machine learning approach. Expert Systems with Applications, 106: 197-216. https://doi.org/10.1016/j.eswa.2018.04.006

Hastie T, Tibshirani R, and Friedman J (2009). Boosting and additive trees. In: Hastie T, Tibshirani R, and Friedman J (Eds.), The elements of statistical learning: Data mining, inference, and prediction: 337-387. Springer, New York, USA. https://doi.org/10.1007/978-0-387-84858-7_10

Hatzivassiloglou V and McKeown K (1997). Predicting the semantic orientation of adjectives. In the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of The European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Madrid, Spain: 174-181. https://doi.org/10.3115/976909.979640

Hochreiter S and Schmidhuber J (1997). Long short-term memory. Neural computation, 9(8): 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735 **PMid:9377276**

John GH and Langley P (2013). Estimating continuous distributions in Bayesian classifiers. ArXiv Preprint ArXiv: 1302.4964. https://doi.org/10.48550/arXiv.1302.4964

Kaur BR, Navneet K, Ravneet K, and Gurpreet K (2016). Opinion mining and sentiment analysis. In the 3rd International Conference on Computing for Sustainable Global Development (INDIACom), IEEE, New Delhi, India: 452-455.

Kharde V and Sonawane P (2016). Sentiment analysis of Twitter data: A survey of techniques. Arxiv Preprint Arxiv: 1601.06971. https://doi.org/10.5120/ijca2016908625

Lenc L and Hercig T (2016). Neural networks for sentiment analysis in Czech. ITAT 2016 Proceedings, CEUR Workshop Proceedings, 1649: 48-55.

Liu B (2017). Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press, Cambridge, UK.

Liu B (2022). Sentiment analysis and opinion mining. Springer Nature, Berlin, Germany.

Mahyoob M, Al-Garaady J, Alblwi A, and Alrahaili M (2022). Sentiment analysis of public tweets towards the emergence of SARS-CoV-2 Omicron variant: A social media analytics framework. Engineering, Technology and Applied Science Research, 12(3): 8525–8531. https://doi.org/10.48084/etasr.4865

Mudinas A, Zhang D, and Levene M (2012). Combining lexicon and learning based approaches for concept-level sentiment analysis. Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining, Association for Computing Machinery, Beijing, China: 152-165. https://doi.org/10.1145/2346676.2346681

Pang B, Lee L, and Vaithyanathan S (2002). Thumbs up? Sentiment classification using machine learning techniques. Arxiv Preprint cs/0205070. https://doi.org/10.3115/1118693.1118704

Pisner DA and Schnyer DM (2020). Support vector machine. Academic Press, Cambridge, USA. https://doi.org/10.1016/B978-0-12-815739-8.00006-7

Pradhan VM, Vala J, and Balani P (2016). A survey on sentiment analysis algorithms for opinion mining. International Journal of Computer Applications, 133(9): 7-11. https://doi.org/10.5120/ijca2016907977

Priyam A, Abhijeeta GR, Rathee A, and Srivastava S (2013). Comparative analysis of decision tree classification algorithms. International Journal of Current Engineering and Technology, 3(2): 334-337.

Sánchez-Núñez P, Cobo MJ, De las Heras-Pedrosa C, Pelaez JI, and Herrera-Viedma E (2020). Opinion mining, sentiment analysis and emotion understanding in advertising: A bibliometric analysis. IEEE Access, 8: 134563-134576. https://doi.org/10.1109/ACCESS.2020.3009482

Sandoval-Almazan R and Valle-Cruz D (2018). Facebook impact and sentiment analysis on political campaigns. Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age, Association for Computing Machinery, Delft, Netherlands: 1-7. https://doi.org/10.1145/3209281.3209328

Šperková L (2018). Review of latent Dirichlet allocation methods usable in voice of customer analysis. Acta Informatica Pragensia, 7(2): 152-165. https://doi.org/10.18267/j.aip.120

Täckström O and McDonald R (2011). Semi-supervised latent variable models for sentence-level sentiment analysis. In the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Portland, USA: 569-574.

Witten IH, Frank E, and Hall MA (2005). Data minding: Practical machine learning tools and techniques. 3rd Edition, Morgan Kaufmann Publishers, San Francisco, USA.

Wong M and Farooq B (2020). A bi-partite generative model framework for analyzing and simulating large scale multiple discrete-continuous travel behaviour data. Transportation Research Part C: Emerging Technologies, 110: 247-268. https://doi.org/10.1016/j.trc.2019.11.022

Zhang L, Ghosh R, Dekhil M, Hsu M, and Liu B (2011). Combining lexicon-based and learning-based methods for Twitter sentiment analysis. HP Laboratories, Technical Report HPL-2011, 89: 1-8.

Zhao J, Liu K, and Xu L (2015). Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press, Cambridge, UK.