ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 14 (2024) No. 10, pp.170-179
DOI:10.33168/JSMS.2024.1010

Handling Class Imbalance for Indonesian Twitter Sentiment Analysis A Comparative Study of Algorithms

Muhammad Zhikri, Wirawan Istiono

Universitas Multimedia Nusantara

wirawan.istiono@umn.ac.id

Abstract. This research investigates the Multinomial Naive Bayes (MNB) and Logistic Regression (LR) algorithms for sentiment analysis on Indonesian language tweets related to ChatGPT. Before classification, TF-IDF and SMOTE will be implemented. A total of 16500 tweets written in the Indonesian language were collected. These tweets were subsequently classified using an Indonesian dictionary of positive and negative phrases. Subsequently, the process of case folding, data purification, tokenization, removal of stopwords, and stemming is executed. The SMOTE oversampling approach is employed to address the issue of class imbalance in the dataset. Comparative evaluation on a trial split of 80:20 shows that LR has a higher accuracy of 86% compared to MNB (74%). LR also shows superior precision, recall, and F1 scores. The results show better LR for Twitter sentiment analysis without significant improvement of the sampling technique.

Keywords: Chat GPT, Logistic Regression, Multinomial Naïve Bayes, Sentiment Analysis, SMOTE

1. Introduction

In November 2022, OpenAI, an artificial intelligence laboratory established by Elon Musk, made a public announcement regarding the introduction of their artificial intelligence robot, ChatGPT. The introduction of this intelligent chatbot garnered global interest and reportedly amassed a user base of 57 million within the initial month following its debut (Cao *et al.*, 2023; Lund and Wang, 2023). One notable benefit of the chatbot under consideration is its ability to engage in interactive written exchanges with individuals and generate responses that closely resemble those of human beings (Ray, 2023; Sallam, 2023).

Even after a span of three months after its initial release, conversations pertaining to ChatGPT continue to persist across various platforms such as online discussion forums, news portals, and social media channels. The intelligent robot created by OpenAI has elicited emotions and opinions from Indonesian netizens on various social media platforms. Twitter is a widely utilized social media platform for individuals to express their responses and opinions. According to scholarly sources, Twitter has been identified as the most prominent microblogging platform globally (McGee, 2023b, 2023a). This platform enables users to generate concise messages or material, including many formats such as text, images, and videos. Twitter is a highly prevalent social media platform in Indonesia, with a substantial user base of 18.45 million individuals as of January 2022 (Hill-Yardin *et al.*, 2023; Wu *et al.*, 2023). Indonesia's inclusion as the fifth country with the highest number of Twitter users globally has been documented. Given this numerical value, it is extremely probable that individuals in Indonesia can effectively articulate their reactions and viewpoints on ChatGPT through the medium of Twitter, thereby enabling their comments and thoughts to serve as a collective representation for gauging public mood pertaining to this subject (King, 2023; Ollivier *et al.*, 2023).

Sentiment analysis refers to a computer approach that seeks to discern and categorize individuals' viewpoints, feelings, assessments, attitudes, and appraisals pertaining to specific subjects, goods, services, entities, individuals, or undertakings. The objective of this study is to analyze and classify individuals' viewpoints expressed on social media platforms, with the aim of categorizing them into good, negative, or neutral sentiments (Giachanou and Crestani, 2016; Antonakaki, Fragopoulou and Ioannidis, 2021). When conducting sentiment analysis using a machine learning methodology, there exist multiple algorithms that can be employed for this purpose (Ardelia and Istiono, 2021; Philips and Istiono, 2021). In this study, sentiment analysis was conducted using two classification algorithms: Multinomial Naive Bayes and Logistic Regression. The purpose was to determine the most effective algorithm for doing sentiment analysis in the context of ChatGPT. The Naive Bayes algorithm is widely acknowledged as a major method for classifying text in the field of text categorization (Wongkar and Angdresey, 2019; Wickramasinghe and Kalutarage, 2021).

The popularity of the subject in question can be attributed to its noteworthy characteristics, such as its efficiency, speed, and simplicity. In a recent study conducted by Kristiyanti et al, a notable discrepancy of 18.50% was identified in the accuracy comparison between the Naive Bayes algorithm and Support Vector Machine (SVM). Based on the aforementioned study conducted by Kristiyanti et al (Kristiyanti *et al.*, 2019), it can be deduced that the Naive Bayes algorithm exhibited a higher level of effectiveness compared to the SVM algorithm, as indicated by the findings. A separate study conducted by Muhammad Yusril et al. demonstrated that the Logistic Regression method displays a significant level of performance in terms of accuracy and precision (Setyawan, Awangga and Efendi, 2018; Kristiyanti *et al.*, 2019). As a result, it can be observed that the Logistic Regression approach exhibits a higher level of effectiveness in comparison to the Naive Bayes algorithm. The distinction between prior research and this study lies in the variation of the analysis conducted. Specifically, this research centers on the sentiment analysis of ChatGPT in the Indonesian language. Additionally, this study aims to compare two algorithms, namely the multinomial naive Bayes algorithm and logistic regression, in relation to sentiment analysis of ChatGPT.

This study centers on the utilization of the Multinomial Naive Bayes algorithm, a variant of Naive Bayes, in comparison with the Logistic Regression algorithm. The objective is to determine the superior algorithm for sentiment analysis in case studies pertaining to ChatGPT, drawing upon the findings of prior research. The utilization of the TF-IDF method was also implemented in the context of sentiment analysis study conducted on ChatGPT. The objective of this study is to perform a sentiment analysis on the Indonesian public's opinion of the ChatGPT chatbot. This analysis will be conducted using the Multinomial Naive Bayes and Logistic Regression algorithms on the social media platform Twitter. The study also aims to evaluate and compare the performance of the Multinomial Naive Bayes and Logistic Regression algorithms in sentiment analysis specifically on Twitter.

2. Materials And Methodology

Contributions In general, the research process can be represented visually as depicted in Figure 1. The process commences with extracting tweets or acquiring data from the social media platform Twitter, followed by categorizing the obtained data based on expressed opinions. Subsequently, the process of data preprocessing is executed, encompassing many essential steps such as data cleaning, case folding, tokenizing, stopword elimination, and stemming. The subsequent phase involves applying the TF-IDF implementation to assign weights to the data. Next, the dataset is divided into two subsets: the training data and the test data. Subsequently, the Multinomial Naive Bayes and Logistic Regression methods are implemented. Subsequently, the data is subjected to a training process, wherein each algorithm generates prediction outcomes that may be compared.

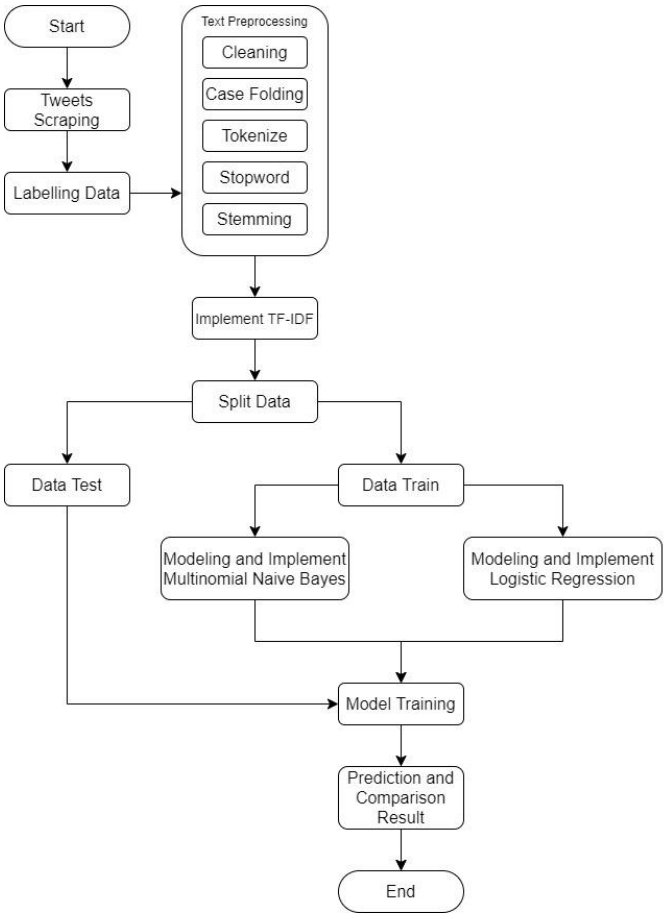

Fig. 1: Research overview diagram

2.1. Data Collection

Tweet scraping is performed to collect tweets or posts from Twitter users in Indonesia. The collection

of tweets is done using the Snscrape library, which is implemented in the Python programming language. This library does not require an official Twitter API (Application Programming Interface) and can gather data directly from Twitter. Tables should be explanatory enough to be understandable without any text reference. Double spacing should be maintained throughout the table, including table headings and footnotes. Table headings should be placed above the table. Footnotes should be placed below the table with superscript lowercase letters. Snscrape also supports the collection of tweets within a specific time range. In this research, it will only gather tweets uploaded between November 30, 2022, and February 28, 2023. This allows for targeted data collection within a defined period.

2.2. Labelling Data

Labeling is a process that assigns each tweet data to a specific sub-class, which includes positive and negative classes. Once the data has been gathered, the subsequent task involves assigning labels to each retrieved tweet. Given the substantial volume of twitter data, totaling 16500 entries, this labeling process is automated through the utilization of an Indonesian dictionary containing positive and negative phrases. Prior to labeling, the tweet data will undergo a cleaning process to remove links, user mentions, symbols, and emoticons. Next, the process of replacing slang phrases is carried out, wherein slang terms are substituted with words that conform to the guidelines of EYD V. The elimination of slang words is accomplished through the utilization of a comprehensive slang dictionary containing a vast array of frequently employed slang, abbreviated, and nonstandard terms.

From testing using slangword replacement, there are 9419 neutrally labeled tweets, 4890 positively labeled tweets, and 2239 negatively labeled tweets. In this study, only positive labels and negative labels are used so that neutrally labeled tweets will be deleted and leave 7129 out of 16500 data.

2.3. Text Preprocessing

Text processing is an essential process conducted before performing sentiment classification. Its purpose is to transform unstructured text data into structured data, resulting in better input for data modeling and analysis. This stage has a significant impact, particularly in sentiment analysis, as it helps obtain clean data [10].

Prior to performing text preprocessing on the dataset, the initial step is to import the csv format file using the pandas library, which is an integrated python library. Next, do case folding on the imported sentence to convert all letters to lowercase. Subsequently, a data purification process will be conducted to eliminate hyperlinks, punctuation marks, numerical values, account mentions or references, and emojis present in the tweets. The subsequent step involves tokenizing the data that has been sanitized and subjected to case folding. Tokenization is the process of dividing phrases into individual word units, which are then gathered and stored in an array, with each word separated by commas. The subsequent stage involves performing stopwords, which is eliminating words that lack significant meaning in the sentence. The process of removing stopwords is facilitated by the Indonesian library of nltk, specifically nltk.corpus. The final step in text preprocessing in this study is stemming, which involves reducing or removing any affixes at the beginning or end of words, resulting in only the base form of the word.

2.4. TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) is a technique used to measure the weight of words in a document. This weight is used to determine the importance of a word in the document. The weight of a word is calculated by multiplying its term frequency (the number of times it appears in the document) by the inverse document frequency (the inverse of the frequency of the word across all documents). As a result (Pimpalkar and Retna Raj, 2020; Jalilifard *et al.*, 2021), words that frequently appear in a particular document but rarely appear in other documents will have a high weight, indicating their uniqueness and relevance to that specific document. Term Frequency (TF) Weighting words with Term Frequency is calculated using the following that shown in Equation 1 to Equation 3.

$$TF(t,d) = \frac{n_{ij}}{\sum_k n_{ik,j}} \quad (1)$$

Where:

TF(t, d) = Term Frequency of term t in document d

n_{ij} = The frequency of term d in document \(d \).

$\sum_k n_{ik,j}$ = Sum of the frequencies of term t in all positions k within document d.

Then calculate the Inverse Term Frequency (IDF) with following formula:

$$idf(t) = \log \frac{N}{df_j} \quad (2)$$

Where:

N = Total number of documents in the collection.

Then calculate the TF-IDF with following formula:

$$wij = tfij \times idf \quad (3)$$

Where:

wij = Represents the weight of the term in class j.

$tfij$ = Represents the total occurrences of term i in class j.

idf = Represents the inverse document frequency of the term.

After performing text preprocessing, the next step is to perform word weighting using the Tfidfvectorizer from the scikit-learn library. From the code that shown in Figure 2, the function fit_transform() is used to tokenize and calculate the word frequency matrix in each index of the tweet data. The result of calculating the word matrix frequency is used in the tfidftransformer object to compute the TF-IDF weights. The transform() function is then applied to convert the word frequency matrix into the actual TF-IDF matrix.

```
from sklearn.feature extraction.text import CountVectorizer
from sklearn.feature extraction.text import TfidfTransformer

#Menghitung matriks frekuensi katavectorizer=CountVectorizer()
X cv = vectorizer.fit transform(df['clean tweet'])

# Menghitung matriks TF-IDF
tfidf transformer = TfidfTransformer()
X tfidf = tfidf transformer.fit transform(X cv)

# Menghitung jumlah total bobot TF-IDF untuk setiap fitur (kata)feature names
=vectorizer.get feature names out()
total scores = X tfidf.sum(axis=0).A1

# Mengurutkan fitur berdasarkanjumlah total bobot TF-IDF secaramenurun
sorted indices = total scores.argsort()[::-1]
sorted features = [feature names[i] for i in sorted indices]sorted scores=tot
al scores[sorted indices]

# Cetak kata dan bobotnya
for feature, score in zip(sorted features, sorted scores):
print("Kata:", feature, "\tJumlah Total Bobot TF-IDF:", score)
```

Fig.2: TF-IDF implementation code snippet

Table 1 represents the cumulative sum of the calculated TF-IDF weights for all data in the document. The word "chatgpt" appears in the first position with a weight of 361.8658520824301.

Table 1. Cumulative TF-IDF calculation result of all documents

Word	TF-IDF
chatgpt	361.8658520824301
bisa	203.17641075230065
ai	139.28390497999126
enggak	130.87720620639735
banget	90.20885083024844
membantu	60.161022869324896
bagus	31.03900870245131
menarik	30.95513411869942

3. Result and Discussion

In In this experiment, two different ratio comparison scenarios were conducted. The first scenario used a 70:30 ratio, with 70% of the data allocated for training and 30% for testing, using the Multinomial Naive Bayes classification. The second scenario used an 80:20 ratio, with 80% of the data allocated for training and 20% for testing. In this case, the classification used was Logistic Regression.

3.1. Multinomial Naive Bayes

The first experiment compared the 70:30 ratio and the 80:20 ratio using Multinomial Naive Bayes. In the 70:30 ratio experiment, the training data consisted of 3,344 positive data and 1,537 negative data, while the testing data consisted of 1,434 positive data and 659 negative data. In the 80:20 ratio experiment, the training data consisted of 3,822 positive data and 1,757 negative data, while the testing data consisted of 956 positive data and 439 negative data. The comparison results between the 70:30 ratio and the 80:20 ratio with Multinomial Naive Bayes classification can be seen in Table 2.

Table 2. Comparison results of 70:30 and 80:20 ratios with Multinomial Naive Bayes

Metrics (%)	70:30		80:20	
Label	Positive	Negative	Positive	Negative
Acuracy	73%		74%	
Precision	72%	96%	72%	96%
Recall	100%	15%	100%	17%
F1-Score	84%	26%	84%	28%

Confussion matrix of the 70:30 ratio and 80:20 ratio with Multinomial Naïve Bayes can be seen in Table 3. Also Table 3 present confusion matrices for each ratio, depicting data pertaining to the actual and anticipated values.

Table 3. Confussion Matrix of Mutinomial Naive Bayes with 70:30 ratio and 80:20

		Predicted Value 70:30		Predicted Value 80:20	
Actual Value		Positive	Negative	Positive	Negative
	Positive	1430	560	953	366
	Negative	4	99	3	73

The accuracy of Multinomial Naïve bayes with 80:20 ratio is higher than the 70:30 ratio. Based on Table 2, it can also be concluded that the overall performance of the 80:20 ratio is higher than the 70:30 ratio.

3.2. Logistic Regression

In the experiment with logistic regression, two ratio comparisons were also conducted: 70:30 and 80:20, with the same number of training and testing data as in the previous experiment. The comparison results between the 70:30 ratio and the 80:20 ratio with logistic regression can be seen in Table 4.

Table 4. Comparison results of 70:30 and 80:20 ratios with Logistic Regression

Metrics (%)	70:30		80:20	
Label	Positive	Negative	Positive	Negative
Acuracy	86%		86%	
Precision	83%	98%	84%	98%
Recall	100%	56%	100%	58%
F1-Score	90%	71%	91%	73%

Confussion matrix of the 70:30 ratio and 80:20 ratio with Logistic Regression shown in Table 5. Based on Table 5, the accuracy for both ratio comparisons is the same, at 86%. However, when considering precision, recall, and F1-score, it shows that the 80:20 ratio performs slightly better, although the difference is not significant.

Table 5. Confussion Matrix of Logistic Regression with 70:30 ratio

	Predicted Value				
Actual Value		Positive	Negative	Positive	Negative
	Positive	1427	293	952	185
	Negative	7	366	4	254

3.3. Testing with SMOTE

The upcoming testing phase will involve the application of the Synthetic Minority Oversampling Technique (SMOTE) to the training data. The implementation of SMOTE is done to address the imbalance between positive and negative classes in the dataset and measure the performance of both algorithms with balanced data. In the 70:30 ratio of the training data, there are initially 3344 positive class instances and 1537 negative class instances. After applying SMOTE, the data will have 3344 instances for both the positive and negative classes. Similarly, in the 80:20 ratio, there are initially 3822 positive class instances and 1757 negative class instances. After applying SMOTE, both classes will have 3822 instances. Figure 3 and Figure 4 represent the results of testing with Multinomial Naive Bayes and SMOTE.

```
Classification Report:
              precision    recall  f1-score   support

    Positive       0.85      0.74      0.79       439
    Negative       0.89      0.94      0.91       956

    accuracy                           0.88      1395
   macro avg       0.87      0.84      0.85      1395
weighted avg       0.88      0.88      0.87      1395

Accuracy: 87.67025089605734
Precision: 0.8858267716535433
Recall: 0.9414225941422594
F1-score: 0.9127789046653144
True Positives (TP): 900
False Positives (FP): 116
False Negatives (FN): 56
True Negatives (TN): 323
```

Fig. 3: The results of testing Multinomial Naive Bayes with an 80:20 ratio and SMOTE

```
Classification Report:
              precision    recall  f1-score   support

    Positive       0.84      0.74      0.79       659
    Negative       0.89      0.94      0.91      1434

    accuracy                           0.87      2093
   macro avg       0.86      0.84      0.85      2093
weighted avg       0.87      0.87      0.87      2093

Accuracy: 87.43430482560917
Precision: 0.8869795109054858
Recall: 0.9358437935843794
F1-score: 0.9107567017305734
True Positives (TP): 1342
False Positives (FP): 171
False Negatives (FN): 92
True Negatives (TN): 488
```

Fig. 4: The results of testing Multinomial Naive Bayes with an 70:30 ratio and SMOTE

The testing outcomes of the Logistic Regression method and SMOTE technique are depicted in Figure 5 and Figure 6, respectively.

```
Classification Report:
              precision    recall  f1-score   support

    Positive       0.88      0.82      0.85       439
    Negative       0.92      0.95      0.94       956

    accuracy                           0.91      1395
   macro avg       0.90      0.89      0.89      1395
weighted avg       0.91      0.91      0.91      1395

Accuracy: 90.96774193548387
Precision: 0.9217479674796748
Recall: 0.948744769874477
F1-score: 0.9350515463917526
True Positives (TP): 907
False Positives (FP): 77
False Negatives (FN): 49
True Negatives (TN): 362
```

Fig. 5: The results of testing Logistic Regression with an 70:30 ratio and SMOTE

```
Classification Report:
              precision    recall  f1-score   support

    Positive       0.86      0.82      0.84       659
    Negative       0.92      0.94      0.93      1434

    accuracy                           0.90      2093
   macro avg       0.89      0.88      0.88      2093
weighted avg       0.90      0.90      0.90      2093

Accuracy: 90.15766841853798
Precision: 0.917687074829932
Recall: 0.9407252440725244
F1-score: 0.9290633608815427
True Positives (TP): 1349
False Positives (FP): 121
False Negatives (FN): 85
True Negatives (TN): 538
```

Fig. 6: The results of testing Logistic Regression with an 80:20 ratio and SMOTE

From the testing results with SMOTE, it was found that the accuracy and precision of Multinomial Naive Bayes significantly increased. There was an accuracy improvement of 14% in both the 80:20 and 70:30 ratio datasets. On the other hand, in the testing with Logistic Regression, there was a slight improvement in accuracy, reaching 5% in the 80:20 ratio and 4% in the 70:30 ratio. Overall, the

performance of Logistic Regression remains better than Multinomial Naive Bayes after applying SMOTE.

4. Conclusion

The experimental findings demonstrate that the Logistic Regression method outperforms the Multinomial Naive Bayes algorithm in datasets with both 70:30 and 80:20 ratios. After conducting a comparative evaluation of an 80:20 trial split, it was shown that both algorithms exhibit greater performance when trained on the 80% portion of the dataset. Specifically, LR demonstrates a superior accuracy of 86% compared to MNB, which reaches 74%. LR also exhibits superior accuracy, sensitivity, and overall performance as assessed by precision, recall, and F1 scores. The findings suggest that the logistic regression (LR) performance was enhanced by Twitter sentiment analysis. However, the sampling technique did not show any significant improvement. SMOTE enhances the accuracy of Multinomial Naive Bayes by up to 14%, although Logistic Regression only experiences a 5% gain. Nevertheless, Logistic Regression consistently outperforms Multinomial Naive Bayes in this sentiment analysis study.

Acknowledgements

Thank you to the Universitas Multimedia Nusantara, Indonesia which has become a place for researchers to develop this journal research. Hopefully, this research can make a major contribution to the advancement of technology in Indonesia

References

Antonakaki, D., Fragopoulou, P. and Ioannidis, S. (2021) 'A survey of Twitter research: Data model, graph structure, sentiment analysis and attacks', *Expert Systems with Applications*. Elsevier Ltd, 164(July 2020), p. 114006. doi: 10.1016/j.eswa.2020.114006.

Ardelia, V. and Istiono, W. (2021) 'Comparative Analysis of Brute Force and Boyer Moore Algorithms in Word Suggestion Search', *International Journal of Emerging Trends in Engineering Research*, 9(8), pp. 1064–1068. doi: 10.30534/ijeter/2021/05982021.

Cao, Y. *et al.* (2023) 'A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT', *Journal of the ACM*. Association for Computing Machinery, 37(4). Available at: http://arxiv.org/abs/2303.04226.

Giachanou, A. and Crestani, F. (2016) 'Like it or not: A survey of Twitter sentiment analysis methods', *ACM Computing Surveys*, 49(2). doi: 10.1145/2938640.

Hill-Yardin, E. L. *et al.* (2023) 'A Chat(GPT) about the future of scientific publishing', *Brain, Behavior, and Immunity*. Elsevier Inc., 110(March), pp. 152–154. doi: 10.1016/j.bbi.2023.02.022.

Jalilifard, A. *et al.* (2021) 'Semantic Sensitive TF-IDF to Determine Word Relevance in Documents', *Lecture Notes in Electrical Engineering*, 736 LNEE, pp. 327–337. doi: 10.1007/978-981-33-6987-0_27.

King, M. R. (2023) 'A Conversation on Artificial Intelligence, Chatbots, and Plagiarism in Higher Education', *Cellular and Molecular Bioengineering*. Springer International Publishing, 16(1), pp. 1–2. doi: 10.1007/s12195-022-00754-8.

Kristiyanti, D. A. *et al.* (2019) 'Comparison of SVM Naïve Bayes Algorithm for Sentiment Analysis Toward West Java Governor Candidate Period 2018-2023 Based on Public Opinion on Twitter', *2018 6th International Conference on Cyber and IT Service Management, CITSM 2018*. IEEE, (Citsm 2018), pp. 1–6. doi: 10.1109/CITSM.2018.8674352.

Lund, B. D. and Wang, T. (2023) 'Chatting about ChatGPT: how may AI and GPT impact academia

and libraries?', *Library Hi Tech News*, 40(3), pp. 26–29. doi: 10.1108/LHTN-01-2023-0009.

McGee, R. W. (2023a) 'How Would History Be Different If Karl Marx Had Never Been Born? A Chatgpt Essay', *SSRN Electronic Journal*, (April). doi: 10.2139/ssrn.4413422.

McGee, R. W. (2023b) 'What Will the United States Look Like in 2050? A Chatgpt Short Story', *SSRN Electronic Journal*, (April). doi: 10.2139/ssrn.4413442.

Ollivier, M. *et al.* (2023) 'A deeper dive into ChatGPT: history, use and future perspectives for orthopaedic research', *Knee Surgery, Sports Traumatology, Arthroscopy*. Springer Berlin Heidelberg, 31(4), pp. 1190–1192. doi: 10.1007/s00167-023-07372-5.

Philips, W. and Istiono, W. (2021) 'Analysis of MinFinder Algorithm on Large Data Amounts', *International Journal of Emerging Trends in Engineering Research*, 9(6), pp. 627–632. doi: 10.30534/ijeter/2021/04962021.

Pimpalkar, A. P. and Retna Raj, R. J. (2020) 'Influence of Pre-Processing Strategies on the Performance of ML Classifiers Exploiting TF-IDF and BOW Features', *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 9(2), pp. 49–68. doi: 10.14201/adcaij2020924968.

Ray, P. P. (2023) 'ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope', *Internet of Things and Cyber-Physical Systems*. The Author, 3(March), pp. 121–154. doi: 10.1016/j.iotcps.2023.04.003.

Sallam, M. (2023) 'The Utility of ChatGPT as an Example of Large Language Models in Healthcare Education, Research and Practice: Systematic Review on the Future Perspectives and Potential Limitations', *medRxiv*, p. 2023.02.19.23286155. Available at: http://medrxiv.org/content/early/2023/02/21/2023.02.19.23286155.abstract.

Setyawan, M. Y. H., Awangga, R. M. and Efendi, S. R. (2018) 'Comparison Of Multinomial Naive Bayes Algorithm And Logistic Regression For Intent Classification In Chatbot', *Proceedings of the 2018 International Conference on Applied Engineering, ICAE 2018*. IEEE, pp. 1–5. doi: 10.1109/INCAE.2018.8579372.

Wickramasinghe, I. and Kalutarage, H. (2021) 'Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation', *Soft Computing*, 25(3), pp. 2277–2293. doi: 10.1007/s00500-020-05297-6.

Wongkar, M. and Angdresey, A. (2019) 'Sentiment Analysis Using Naive Bayes Algorithm Of The Data Crawler: Twitter', *Proceedings of 2019 4th International Conference on Informatics and Computing, ICIC 2019*, (July), pp. 3–8. doi: 10.1109/ICIC47613.2019.8985884.

Wu, T. *et al.* (2023) 'A Brief Overview of ChatGPT: The History, Status Quo and Potential Future Development', *IEEE/CAA Journal of Automatica Sinica*, 10(5), pp. 1122–1136. doi: 10.1109/JAS.2023.123618.