INCITEST
International Conference on Informatics Engineering, Science & Technology

UNIKOM

Comparative Analysis of Logistic Regression and Support Vector Machine for Sentiment Classification of Shopee App Reviews on the Play Store

Maulvi Inayat Ali *, Sri Supatmi

Master of Information System, Universitas Komputer Indonesia, Bandung, Indonesia

*Corresponding E-mail: maulvi.75123016@mahasiswa.unikom.ac.id

Abstract. The rapid advancement of information technology has significantly transformed consumer shopping behavior on mobile e-commerce platforms, with Shopee standing out as one of the most popular choices. User reviews provide essential insights into customer sentiment and preferences, thereby highlighting the importance of automated sentiment analysis for capturing public opinion. This study aims to evaluate and compare the performance of two supervised learning algorithms Support Vector Machine (SVM) and Logistic Regression (LR) in classifying sentiments from 3,000 Shopee app reviews obtained from the Playstore. The research employed a 5-fold cross-validation method, with performance measured using accuracy, precision, recall, and F1-score. The findings indicate a clear trade-off between the models: SVM achieved a slightly higher test accuracy of 80.67% compared to 80.33% for Logistic Regression, but it showed greater susceptibility to overfitting. Conversely, Logistic Regression demonstrated more stable performance, maintaining a balanced trade-off between precision and recall without noticeable overfitting. This trade-off is likely due to SVM's complexity and sensitivity to high-dimensional feature spaces, whereas Logistic Regression's simpler linear model offers greater generalizability on this dataset. Based on these results, Logistic Regression is recommended for applications that prioritize consistency and robustness, while SVM may be preferable in contexts where maximizing precision is the primary objective.

Keywords: Sentiment Analysis, Support Vector Machine (SVM), Logistic Regression, 5-Fold Cross-Validation, Machine Learning.

1. Introduction

The acceleration of information technology has reshaped how consumers shop, with mobile e-commerce applications playing a central role. In Southeast Asia, Shopee stands out as a leading platform that has quickly become a preferred destination for online purchases across diverse product and service categories. By 2023, its user base numbered in the millions, and the extensive corpus of user-generated reviews offers rich evidence for gauging customer sentiment and preferences (M.J. Hossin 2022 et al.). Within this context, sentiment analysis is crucial because it automates the processing of large volumes of opinions, thereby furnishing actionable insights that can inform product refinement, marketing decisions, and customer support strategies (Lakshay Bharadwaj 2023).

At its core, sentiment analysis leverages machine-learning methods to categorize user content such as reviews into positive, negative, or neutral classes (Huang et al 2023). The effectiveness of such classification hinges strongly on model selection. Two algorithms frequently employed for this task are Support Vector Machine (SVM) and Logistic Regression (LR), both of which have shown robust performance across multiple domains (Premasudha & Rampalli 2024). SVM is particularly adept at coping with high-dimensional feature spaces typical of text data, whereas LR is valued for its simplicity and interpretability, making it a dependable option for binary classification problems.

Although numerous studies have explored these models in various settings, work specifically targeting Shopee reviews especially within the Southeast Asian context remains limited. This study addresses that gap by evaluating SVM and LR for classifying sentiments in Shopee reviews sourced from the Playstore. The comparison is based on accuracy, precision, recall, and F1-score, while also acknowledging the distinctive linguistic and cultural characteristics present in Southeast Asian user discourse.

2. II Related Work

This study evaluates SVM against LR for sentiment analysis using Shopee reviews collected from the Play Store, with labels of positive or negative and performance measured by accuracy, precision, recall, and F1-score. SVM is well-suited to high-dimensional text data and supports non-linear separations through kernel methods; in contrast, LR provides a compact, interpretable framework that models class probabilities from linear feature combinations, favoring use cases that prioritize clarity and explainability (M. Tusar & A.M Islam 2019).

A range of prior studies underscores the utility of these algorithms in sentiment tasks. A study by Indulkar & Patil compared multiple machine-learning models on Twitter data and reported that LR achieved the best accuracy (Indulkar & Patil 2021). In a similar vein, Tusar and Islam evaluated SVM, LR, and Random Forest for sentiment analysis of U.S. airline tweets, finding that SVM and LR both attained the top accuracy (77%) using a Bag-of-Words representation. Complementing these results, Styawati et al. highlighted SVM's superior performance for classifying sentiment related to Covid-19 vaccination policies.

Other investigations demonstrate the benefits of specific feature engineering choices. Cahyanti et al. showed that pairing SVM with TF-IDF and Latent Dirichlet Allocation (LDA) features yields strong results for movie review classification. Conversely, Imamah and Rachman reported 94.71% accuracy in Twitter sentiment analysis on Covid-19 topics using LR with TF-IDF weighting. Collectively, these findings confirm that both SVM and LR are well-suited to sentiment classification across diverse domains (Imamah & Rahman 2021).

More recently, Bharathi et al. compared SVM and LR on e-commerce product reviews and observed that LR delivered more stable performance across datasets (Bharathi et al 2022). Building on traditional approaches, Kaur proposed a hybrid framework that integrates SVM with deep learning components for user-generated content, achieving higher accuracy than classical models alone (Kaur & Sharma 2023). These developments motivate continued exploration of hybrid strategies and the synergies among machine-learning techniques.

Against this backdrop, the present work systematically contrasts SVM and LR on Shopee review data to identify the more effective approach for mobile e-commerce sentiment classification, while paying particular attention to the linguistic and cultural nuances characteristic of Southeast Asian reviews.

3. Methodolody

3.1 Data Source

The dataset for this study consists of 3,000 reviews of the Shopee app, extracted from the Google Play Store via web scraping. We employed the google_play_scraper library, setting the count parameter to 3000. To ensure the data was contextually appropriate, we targeted the Indonesian app version (com.shopee.id) and set both language and country to 'id'. The key parameter sort=Sort.MOST_RELEVANT was used to curate a dataset that reflects the most significant user opinions. This is a standard and validated method for collecting review data for sentiment analysis (Park et al 2015).

Table 1. Shopee App Review and Rating Data

Review	Rating
After the update, there are many bugs, loading is unclear, pages not found, even though the Wi-Fi is fast. It's really bad.	1
Every time I shop on Shopee, I am really satisfied. The orders are as expected, and the delivery is always fast.	5
The Shopee app is very helpful for shopping. I really like it. The delivery is fast. Please increase my Shopee Pay limit.	5
The current version is slow with many bugs... Watching live, opening the store, viewing products, it's slow... All results lead me to another store which is smoother... Please fix it, thank you.	2
Shopee is getting worse. I have a lot of quota and a good network, but when I try to open the app, there's always a notification saying "network check failed." It's troublesome.	3

3.2 Data Labeling

After the data collection process, each review was automatically assigned a sentiment label using a binary method based on the rating given by the user. This process was carried out by setting the following criteria:

- Negative Label: Reviews with a rating between 1 and 3 were considered to express negative sentiment.
- Positive Label: Reviews with a rating of 4 or 5 were considered to express positive sentiment.

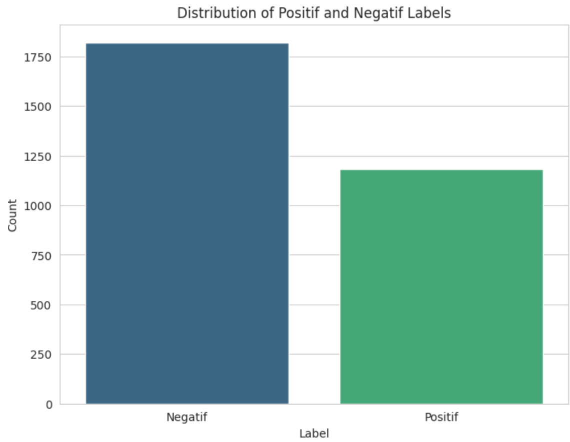

Figure 1. label distribution.

Figure 1 displays the distribution of positive and negative labels within the dataset. On this graph, the x-axis represents the available labels, "Positive" and "Negative," while the y-axis shows the count for each label in the dataset. The visualization reveals that the number of "Negative" labels is significantly higher than "Positive" labels, indicating an imbalance in the data distribution between these two categories.

This approach is based on studies showing that user ratings significantly correlate with the sentiment perceived towards an application (Jadhav & Saha 2023). By using ratings as the primary indicator, the labeling process can be efficiently automated.

Table 2. Review and Rating Data with Labels

Review	Rating	Label
After the update, there are many bugs, loading is unclear, pages not found, even though the Wi-Fi is fast. It's really bad.	1	Negative
Every time I shop on Shopee, I am really satisfied. The orders are as expected, and the delivery is always fast.	5	Positive
The Shopee app is very helpful for shopping. I really like it. The delivery is fast. Please increase my Shopee Pay limit.	5	Positive
The current version is slow with many bugs... Watching live, opening the store, viewing products, it's slow... All results lead me to another store which is smoother... Please fix it, thank you.	2	Negative

Shopee is getting worse. I have a lot of quota and a good network, but when I try to open the app, there's always a notification saying "network check failed." It's troublesome.	3	Negative

3.1 Data Processing
The preprocessing process is a crucial step in text analysis, primarily to ensure that the review data is of high quality and uniform before feature extraction and modeling. The preprocessing in this study involves the following stages

3.3.1 Case Folding
The first stage is converting all text to lowercase. This step eliminates capitalization differences, so words like "Bagus" and "bagus" are treated as identical. Case folding is a standardization process that reduces data redundancy and improves the consistency of the input for the classification model (D Jurafsky & J.H. Martin 2021)

3.3.2 Text Cleaning
At this stage, irrelevant or noisy elements are removed, which includes:

- Removing Mentions and Hashtags: Deleting patterns that contain user references or topic labels (e.g., @username and #hashtag), as these elements typically do not provide significant semantic value for sentiment analysis.
- Removing URLs: Eliminating links or URLs that may be present in the reviews to avoid interference with the analysis.
- Removing Numbers and Symbols: Deleting numbers, special characters, and punctuation that do not contribute to the semantic interpretation of the review.
- Normalizing Spaces: Removing excess spaces and tidying up the text to produce a consistent representation.

3.3.3 Removing Emojis and Non-ASCII Characters
Although emojis can convey emotional information, non-ASCII characters often add unnecessary complexity. Therefore, emojis and non-ASCII characters are removed to simplify the text representation, ensuring only relevant characters are retained

3.3.4 Normalizing Duplicate Characters
To address excessive character repetition (e.g., "heyyy" or "wowww"), normalization is performed by converting sequences of characters repeated more than twice into a single character

3.3.5 Tokenization
Tokenization is the process of breaking down the cleaned text into individual word units (tokens). This process is fundamental in natural language processing as it allows the text to be represented as vectors or numerical features that can be further processed by classification algorithms.

3.3.6 Stopword Removal
Stopwords are common words that appear frequently but lack significant semantic value (e.g., "dan," "yang," "di"). Stopwords are removed to reduce noise and enhance the effectiveness of feature representation, allowing the model to focus on words that contribute to sentiment differences.

3.3.7 Stemming
Stemming is a technique that strips words down to their fundamental root. Taking the Indonesian words "berlari," "lari," and "pelari" as an example, the stemming process would

convert all of them to just "lari." This method is valuable because it groups together words that have the same core meaning, effectively simplifying the feature set for analysis.

3.3.8 Feature Extraction & Weighting with TF-IDF

For the feature extraction phase of this research, we used the Term Frequency-Inverse Document Frequency (TF-IDF) method to assign weights to the words. This is a well-regarded technique in text analysis, primarily used to translate raw text from reviews into a numerical vector format that classification algorithms can process (Salton & Bukcley 1988).

The fundamental principle of TF-IDF is to identify how important a word is to a document. A term receives a high TF-IDF weight if it shows up often in one specific document but is rare in all the others. Conversely, words that are extremely common across the entire collection of documents (like "and," "the," "in") are considered less meaningful and are therefore assigned a very low weight.

The first part of this calculation, Term Frequency (TF), is typically based on the proportional count of a term t within a single document d. The simplest way to figure out TF is:

$$TF(t,d) = \frac{f_t, d}{\sum_{t'} ft', d}$$

where:

- f_t, d is the number of times a term appears in a document.
- $\sum_{t'} ft', d$ is the total number of all terms that appear in the document *d*.

There are also other variations of TF, such as log normalization or boolean frequency (1 if the term appears, 0 if it does not). The choice of TF type is often adjusted based on the type of data and the needs of the analysis [12]

There are also other variations of TF, for instance, log normalization:

$$TF_{log}(t,d) = 1 + \log(f_t, d)$$

or boolean frequency (1 if the term appears, 0 if not). The choice of TF type is often adjusted according to the type of data and the needs of the analysis [12].

Inverse Document Frequency (IDF) measures the inverse of a term's frequency across a corpus. The more documents that contain a specific term, the lower its IDF value, as it is considered less "informative." The common formula for IDF is:

$$IDF(t,D) = \log\left(\frac{N}{n^t}\right)$$

where:

- N is the total number of documents in the corpus D.
- n_t is the number of documents that contain the term t.

By combining the two components above, the TF-IDF weight of a term t in a document d is defined as:

$$TF = IDF(t,d,D) = TF(t,d) \times IDF(t,D).$$

The result of applying TF-IDF is a feature matrix, where each row represents one document (a review) and each column represents a unique word from the entire corpus.

3.4 Machine Learning Algorithm

3.4.1 Support Vector Machine (SVM)

SVM is a classification algorithm based on the concept of a hyperplane, or a separating plane, in a high-dimensional space (Cortes & Vapnik 1995). SVM works by finding the optimal hyperplane that separates two classes of data with the maximum margin

Suppose we have a dataset $\{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in R^d$ and the class label is $y_i \in \{-1, +1\}$ .The equation for the hyperplane is:

$$w.x + b = 0$$

where $w \in R^d$ is the weight vector, and $b \in R$ is the bias (a scalar). To ensure an optimal separation, SVM requires that:

$$y_i(w.x_i + b) \geq 1 \ \forall i \in \{1, 2, \ldots, N\}$$

The margin separated by the hyperplane is defined as $\frac{2}{||w||}$. Thus, the optimization problem for SVM is to maximize the margin by minimizing $||w||^2$, which is mathematically formulated as:

$$min_{w,b} \frac{1}{2} ||w||^2$$

However, in practice, data isn't always perfectly separable. Therefore, a slack variable ε_i and a regularization parameter C are introduced to allow for margin violations:

$$min_{w,b,\varepsilon} \frac{1}{2} ||w||^2 + C \sum_{i=1}^{N} \varepsilon_i$$

with the constraint:

$$y_i(w.x_i + b) \geq 1 - \varepsilon_i, \quad \varepsilon_i \geq 0$$

The parameter C controls the trade-off between a large margin and classification error. A larger value for C will force the model to be stricter in avoiding classification errors on the training data, which runs the risk of causing overfitting

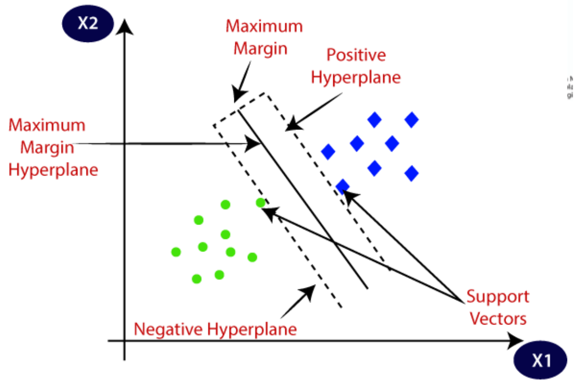

Figure 2. The Concept of Support Vector Machine

Figure 2 illustrates how SVM works in classifying two different classes of data using a hyperplane and margin.

3.4.2 Logistic Regression

Logistic Regression is a classification method used to predict the probability of an event occurring or not occurring based on input values. It models the probability of an event by using the logistic or sigmoid function. The sigmoid function is utilized because its output is a value between 0 and 1, which is suitable for representing probability (Hosmer et al 2013).

The Logistic Regression model for predicting the probability of the positive class $(y = 1)$ can be expressed as:

$$P(y = 1|x) = \frac{1}{1 + e^{-(w.x+b)}}$$

where w is the weight vector, x is the input or feature vector, and b is the bias. Meanwhile, the probability for the negative class $(y = 0)$ is calculated as:

$$P(y = 0|x) = 1 - P(y = 1|x)$$

A Logistic Regression model's classification is determined by a simple cut-off point, generally set at 0.5. Data is sorted into the positive class if its probability is greater than 0.5, and into the negative class if its probability is 0.5 or less. Beyond this binary outcome, the model's mechanics can also be understood by examining the odds and log-odds, which are mathematically linked via the logit function:

$$Log - Odds = \ln\left(\frac{P(y=1|x)}{1-P(y=1|x)}\right) = w.x + b$$

At the heart of Logistic Regression is the sigmoid function, which is responsible for converting the model's linear output into a probability constrained between 0 and 1. It is this probabilistic nature that allows the algorithm to be such a popular and effective tool for predicting outcomes that fall into two or more distinct classes.n to be widely used for predicting binary or multi-class outcomes.The sigmoid function is a key element in Logistic Regression because it transforms the output of the linear combination of input variables into a probability value between 0 and 1.

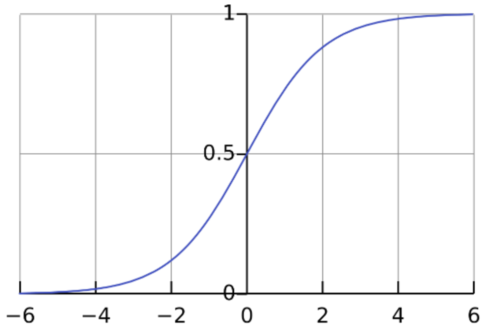

Figure 1 Sigmoid Function

The sigmoid function is defined as:

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

where:

- $z = w.x + b$
- w is the weight vector,
- x is the feature/input vector,
- b is the bias,
- e is Euler's number.

To arrive at a classification, the model uses the output from the sigmoid function. Because of its "S" shape, this function generates seamless probability predictions. A standard 0.5 threshold is then applied to these scores to make the final call: if the output probability is greater than 0.5, the instance is labeled as class 1; otherwise, it is labeled as class 0.

3.5 Model Evaluation

The Confusion Matrix plays a vital role in evaluating the performance of classification models, for both binary and multiclass classification problems. In general, a Confusion Matrix is a table that displays the frequency of a model's predicted outcomes against the actual classes. The rows typically represent the actual classes, while the columns represent the predicted classes (Han et al 2011).

Several relevant metrics derived from the Confusion Matrix, particularly for binary classification, include the following:

- Accuracy

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

Accuracy measures the percentage of correct predictions (both positive and negative) out of the total predictions made. This metric is effective when the dataset is balanced (i.e., the number of positive and negative samples is relatively equal).

- Precision (Positive Predictive Value)

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

Precision measures the proportion of positive predictions that were actually correct. The higher the precision, the fewer errors the model makes in labeling a negative sample as positive (false positives)

- Recall (Sensitivity / True Positive Rate)

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

Recall measures the model's ability to identify all actual positive samples. The higher the recall, the fewer positive cases are missed by the model (fewer false negatives)

- F1-Score

The F1-score is the harmonic mean of precision and recall. This metric is particularly useful when there's a need to balance both precision and recall, especially in the case of an imbalanced dataset.

4. Results and Discussion

This research pitted two machine learning algorithms, Support Vector Machine (SVM) and Logistic Regression, against each other in a classification task. The models were built using an 80% training and 20% testing split of the data, with 5-fold cross-validation employed to ensure a fair and objective performance measurement.

Table 3. Comparison of Test Results

Method	Label	Precision	Recall	F1-Score	Accuracy
SVM	Negative	78.00%	95.00%	86.00%	80.67%
	Positive	88.00%	59.00%	71.00%	
Logistic Regression	Negative	79.00%	93.00%	85.00%	80.33%
	Positive	85.00%	61.00%	71.00%	

The final test scores showed that the two models performed at a nearly identical level. The SVM algorithm registered a testing accuracy of 80.67%, just slightly ahead of the Logistic Regression model's 80.33%. When focusing on the "Negative" label, both were quite capable. Logistic Regression achieved a precision of 0.79 and recall of 0.93, while SVM posted a precision of 0.78 and a higher recall of 0.95, suggesting it was more sensitive in flagging all potential "Negative" instances.

The story was different for the "Positive" label. SVM demonstrated superior precision at 0.88, meaning its positive classifications were highly reliable. However, its recall was lower at 0.59. In contrast, Logistic Regression, with a precision of 0.85 and a recall of 0.61, proved more effective at capturing a larger portion of all the actual "Positive" cases

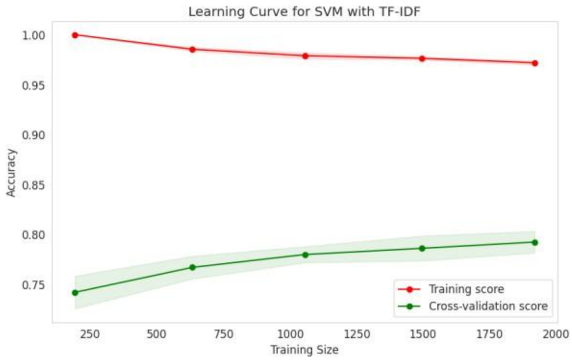

Figure 4. SVM Learning Curse

Figure 4 shows the learning curve for the Support Vector Machine (SVM) model with TF-IDF features, where:

- The Training Score (red line) indicates the model's accuracy on the training data. This accuracy is observed to decrease as the training set size increases, which suggests potential overfitting.
- The Cross-Validation Score (green line) measures the model's accuracy on the test data. Its curve is more stable and shows a slight increase, indicating that even as the model trains on more data, its performance on unseen data does not reach the high levels achieved on the training data.

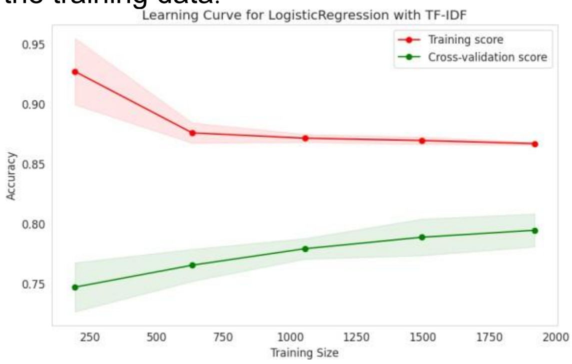

Figure 5. Logistic Regression Learning Curve

Figure 5 shows the learning curve for the Logistic Regression model with TF-IDF, where:

- The Training Score (red line) for this model is more stable, although it slightly decreases as the training data size increases. This indicates that the model is more stable compared to SVM.
- The Cross-Validation Score (green line) is also more stable and shows a slight increase, reflecting a balance between training and testing performance without significant signs of overfitting.

Overall, SVM tends to excel in precision for the "Positive" label, but Logistic Regression is more balanced between precision and recall across both labels. However, it should be noted that the SVM model shows signs of overfitting, with a much higher training accuracy (96.79%) compared to its testing accuracy. Meanwhile, Logistic Regression demonstrates better stability between its training accuracy (86.58%) and testing accuracy.

Therefore, although SVM has slightly better performance in terms of precision, Logistic Regression is superior regarding performance stability and the balance between precision and recall. The choice of the more appropriate model can be tailored to specific application needs, depending on whether the priority is to optimize precision or recall.

5. Conclusion

In comparing Support Vector Machine (SVM) and Logistic Regression for data classification using TF-IDF features, both models achieved strong and nearly identical testing accuracies with an 80/20 training-testing split, validated via 5-fold cross-validation. However, their performance profiles differed significantly. While SVM excelled in precision for the "Positive" class, it showed a tendency toward overfitting, as indicated by a substantial gap between training and testing accuracy. In contrast, Logistic Regression demonstrated more consistent performance, maintaining a better balance between precision and recall without significant overfitting. Based on these findings, Logistic Regression is the preferred model for scenarios prioritizing reliability and stability. SVM, however, is advantageous when the goal is to maximize precision for a specific class, albeit with the risk of overfitting.

6. Conclusion

This study is constrained by the use of a single dataset, which may restrict the extent to which the findings generalize. The analysis also considers only two algorithms, leaving room to incorporate additional models (e.g., Neural Networks). Future investigations could evaluate performance across more varied datasets, address class imbalance, and examine advanced techniques such as deep learning to further enhance sentiment analysis accuracy.

References

Bharathi, R., Bhavani R., & -, P. B. R. (2022). Twitter Text Sentiment Analysis of Amazon Unlocked Mobile Reviews Using Supervised Learning Techniques. Indian Journal of Computer Science and Engineering, 13(4), 1242–1253. https://doi.org/10.21817/indjcse/2022/v13i4/221304100

Cahyanti, E. Yulianto, and H. Setiawan, "SVM-based sentiment analysis using TF-IDF and LDA features for movie review classification," Proc. 2020 Int. Conf. on Artificial Intelligence and Computer Vision (AICV), pp. 198-203, 2020.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

Han, J., Kamber, M., & Pei, J. (2011). Data mining: concepts and. Techniques (3rd ed), Morgan Kauffman, 68.

Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression. John Wiley & Sons.

Hossain, M. J., Joy, D. D., Das, S., & Mustafa, R. (2022, February). Sentiment analysis on reviews of e-commerce sites using machine learning algorithms. In 2022 International Conference on Innovations in Science, Engineering and Technology (ICISET) (pp. 522-527). IEEE.
Huang, H., Zavareh, A. A., & Mustafa, M. B. (2023). Sentiment analysis in e-commerce platforms: A review of current techniques and future directions. Ieee Access, 11, 90367-90382.

Imamah S. and R. Rachman, "Sentiment analysis of Twitter data related to Covid-19 using Logistic Regression with TF-IDF weighting," Proc. 2021 Int. Conf. on Machine Learning and Data Science (MLDS), pp. 82-87, 2021.

Indulkar, Y., & Patil, A. (2021, March). Comparative study of machine learning algorithms for twitter sentiment analysis. In 2021 International Conference on Emerging Smart Computing and Informatics (ESCI) (pp. 295-299). IEEE.

Jadhav, P., & Saha, A. (2023, December). Sentiment analysis of mobile app reviews using robotic process automation. In *2023 7th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)* (pp. 1-6). IEEE.

Jurafsky, D., & Martin, J. H. (2013). *Speech and Language Processing: Pearson New International Edition PDF eBook*. Pearson Higher Ed.

Kaur, G., & Sharma, A. (2023). A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis. Journal of big data, 10(1), 5.

Lakshay Bharadwaj. (2023). Sentiment Analysis in Online Product Reviews: Mining Customer Opinions for Sentiment Classification. *International Journal for Multidisciplinary Research*, *5*(5). https://doi.org/10.36948/ijfmr.2023.v05i05.6090. Premasudha, B. G., & Rampalli, V. (2024, October). A Comparative Study of Logistic Regression, Support Vector Machines, and LSTM Networks for Sentiment Classification in Academic Reviews. In 2024 First International Conference on Innovations in Communications, Electrical and Computer Engineering (ICICEC) (pp. 1-11). IEEE.

Park, D. H., Liu, M., Zhai, C., & Wang, H. (2015, August). Leveraging user reviews to improve accuracy for mobile app retrieval. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 533-542).

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, *24*(5), 513-523.

Sharma, A., Mishra, S. K., & Srivastav, V. K. (2023). The Evolution and Impact of E-Commerce. Journal of Namibian Studies, 33.

S. Styawati, F. S. Handoko, and D. Yuliana, "Sentiment analysis of Covid-19 vaccination policy using Support Vector Machine (SVM) approach," Int. J. Inf. Technol. vol. 13, no. 2, pp. 567-574, 2021

Tusar, M. T. H. K., & Islam, M. T. (2021, September). A comparative study of sentiment analysis using NLP and different machine learning techniques on US airline Twitter data. In *2021 International conference on electronics, communications and information technology (ICECIT)* (pp. 1-4). IEEE.