Performance Analysis of Machine Learning Techniques for Sentiment Analysis

Muhamad Hariz Izzat Ahmad Hapez, Noor Latiffah Adam(✉), and Zaidah Ibrahim

Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, 40450 Shah Alam, Malaysia
{latiffah508,zaidah782}@uitm.edu.my

Abstract. Sentiment analysis determines the sentiment or opinion of a given text. A sentiment analysis model can classify whether a given text data is positive or negative by extracting meaning from the natural language. The growth of social media such as Twitter, forum discussions and reviews, contributed to the huge data repository in digital form. Analyzing these huge data manually is very time consuming and challenging. Thus, applying machine learning techniques can automatically classify the sentiment effectively. This research compares the performance of five popular machine learning techniques for sentiment analysis namely, Support Vector Machine (SVM), Logistic Regression, Naïve Bayes, Random Forest and K-Nearest Neighbor using a publicly available dataset from kaggle.com. Their classification performances are compared based on accuracy and training time where fine tuning of some of the hyperparameters are per-formed to improve the accuracy. Experimental analysis indicates that SVM with linear kernel function produces the highest accuracy but a slower training time. On the other hand, Naïve Bayes requires the shortest training time but with a slightly lower accuracy compared to SVM.

Keywords: Sentiment analysis · Support vector machine · Logistic regression · Naïves Bayes · Random Forest · K-nearest neighbor

1 Introduction

Sentiment analysis or also known as opinion mining is a natural language processing technique used to determine whether the data is positive, negative or neutral. People are eager to share their comments and reviews on social media. This kind of short review could highlight their preference on certain topic [1–3]. Sentiment analysis is often performed on textual data which may use slang phrases, misspellings, short forms, recurring characters, the use of dialects and modern emoticons [4, 5]. The same words and phrases can be used in a different context, thus making it difficult to be determined. Such analysis, for example, may help businesses on monitoring the brand and product sentiment in the customer feedback and understanding the customers' needs [6, 7]. It is extremely crucial because it helps businesses to quickly understand the overall opinions of their customers. By automatically sorting the sentiments behind those reviews, social media conversations, and more, you can make faster and more accurate decisions.

H. Badioze Zaman et al. (Eds.): IVIC 2021, LNCS 13051, pp. 205–213, 2021.
https://doi.org/10.1007/978-3-030-90235-3_18

An example of this application in the real world, is when we want to study the people's opinion on the Covid-19 [8, 9] vaccine. One way to do that is by doing a survey, interview, questionnaire, etc. However, these methods take an enormous amount of time and cost. Therefore, instead of doing these traditional methods, we can scrap people's opinions from the social media, and then running a sentiment analysis towards the scrapped text [10]. Not only that it will save time and any unnecessary costs, but we can also get up to millions of samples from all over the world. However, the accuracy of this prediction is also dependent on the models that were being applied. Some of the machine learning approaches that help classify the sentiments are Logistic Regression [11], Support Vector Machine [12], Random Forest [11], K-Nearest Neighbors [11] and Naïve Bayes [7, 13]. Thus, the goal of this study is to find the best machine learning algorithms out of these five models. We will be tuning the hyper-parameters of each model so that we can find an optimal combination of hyperparameters that minimizes a predefined loss function to give better results.

2 Literature Review

One good reason why opinion mining worth to be explored is that, we have massive data recorded in digital form which have potential to be examined [4, 11, 14–16]. The growth of social media such as Twitter, forum discussions and reviews, contributed to the huge data repository. To handle such big data, require intelligent approach such as machine learning. This section will elaborate on the machine learning approaches used to classify the sentiments.

Support Vector Machine (SVM) is a powerful machine learning model algorithm which is used for both classification and regression. But generally, it is used in the classification problem. The strength of an SVM rooted from its ability to learn the data classification patterns with balanced accuracy and reproducibility [17].

Logistic Regression is a regression model that utilizes binary on the targeted variables. In other words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no). Study in [18], reported that, the classifier has confidence when predicting the positive sentiments but biased when predicting negative reviews.

The Naïve Bayes classifier assumed that the presence of a particular feature in a class is unrelated to the presence of any other features of Naïve Bayes. It is widely used for text classification and spam detection. Despite the simplicity of this model, it works surprisingly well for document classification [19]. Naïve Bayes on text classification require a small data set for training [2]. Conditional probability can be used to classify words into their respective categories.

Random forest was introduced by Breiman [20]. It is a tree-based technique that uses a large number of decision trees built out of randomly selected sets of features. Contrary to the simple decision tree, it is highly uninterpretable, but it is generally produced good performance makes it a popular algorithm.

The K-Nearest Neighbor algorithm, commonly known as k-NN, is a nonparametric approach where the response of a data point is determined by the nature of its K-Nearest Neighbors from the training set. It is suitable to be used in both classification and

regression settings [21]. The higher the parameter k, the higher the bias, and the lower the parameter k, the higher the variance.

3 Methodology

3.1 Data Description

The data that we obtained is from Kaggle. It contains 1,600,000 number of rows and extracted using the twitter API. There are a total of 800,000 rows for Positive Sentiment and another 800,000 for Negative Sentiment which is perfectly balanced. However due to time complexity, we decided to take a sample of 40,000 number of samples with a balanced Positive and Negative ratio. The data contains 6 number of columns, and the description of the data is on Fig. 1.

1. target: the polarity of the tweet (*0* = negative, *2* = neutral, *4* = positive)
2. ids: The id of the tweet (*2087*)
3. date: the date of the tweet (*Sat May 16 23:58:44 UTC 2009*)
4. flag: The query (*lyx*). If there is no query, then this value is NO_QUERY.
5. user: the user that tweeted (*robotickilldozr*)
6. text: the text of the tweet (*Lyx is cool*)

Fig. 1. Data description

Figure 2 shows the snippet of the data that we obtained from Kaggle. As you can see, there are a total of six columns, namely Target, Ids, Date, Flag, User, and Text. However, we will only be using the Target and Text columns. As you can see from Fig. 3, there are a total 40,000 rows of data that we have selected with a balance between the positive and negative sentiment.

	Target	Ids	Date	Flag	User	Text
0	0	2323066693	Wed Jun 24 23:50:25 PDT 2009	NO_QUERY	iKarimah	talking to... no one
1	0	2323067072	Wed Jun 24 23:50:28 PDT 2009	NO_QUERY	PsychobillyCass	wanna listen to "my girl" but can't ...
2	0	2323067309	Wed Jun 24 23:50:30 PDT 2009	NO_QUERY	originald	OK, finally threw everything out. Trying some...
3	0	2323067331	Wed Jun 24 23:50:30 PDT 2009	NO_QUERY	pariahriot	My dumbass wiped the micro sd card I had in my...
4	0	2323067540	Wed Jun 24 23:50:31 PDT 2009	NO_QUERY	Rubyam	@HorseCrazyBoy1 oh apologies dear friend - i w...

Fig. 2. Data snippet

3.2 Data Pre-processing

One of the common steps that we did when we do data pre-processing is to remove null values if there is any. Null values can cause misleading results. Besides that, we also remove any duplicate values of the 'text' column. Duplicate values will only put unnecessary weight to a certain parameter in the model that might cause the model to be biased or overfit.

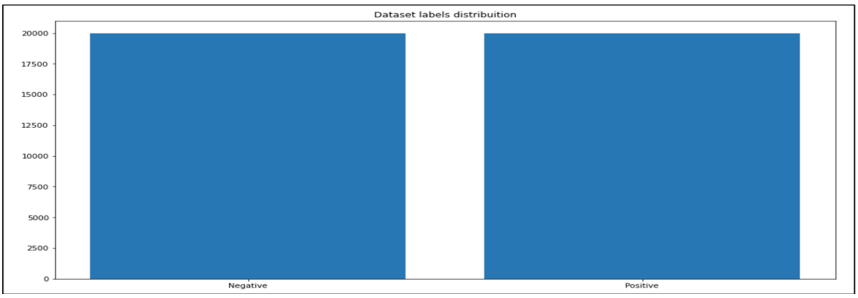

Fig. 3. Bar plot of the sentiment data

Then for the text analysis, the data cleaning step applied is very important for the data to be reliable. There are a lot of possible data noise for the text, such as the inconsistency use of the upper and lower cases, unexpected words such as symbols or emojis, also some useless words like nouns and others. This is expected since the text is one of the most unstructured data forms [22].

In this study, we will be using the stopwords list from the NLTK library to remove any unnecessary words in the text such as 'a', 'and', 'how' and others. Next, we also normalise the text data by using the snowball stemmer from NLTK to convert the words into its root form. For example, from 'stemmed' into 'stem', from 'kicks, kicked, kicking' into 'kick', etc.

3.3 Machine Learning

We have tested five different types of machine learning algorithms to find the best model to classify the sentiments of a Twitter text, namely, SVM, Logistic Regression, Naïve Bayes, Random Forest and K-Nearest Neighbors. We have tuned the hyper-parameters in each model to find out the best model in this study. We compared the speed of the training time and the accuracy of each model. The following subsection will describe the parameters of each algorithm that will be tested.

SVM. The goal of support vector machines is to find the line that maximizes the minimum distance to the line. The parameters that will be tested will be as follows:

- Kernel Specifies the kernel type to be used in the algorithm. [linear, poly, rbf]
- Gamma Kernel coefficient. [scale, auto] 5
- Decision function shape: Whether to return a one-vs-rest ('ovr') decision function of shape (n_samples, n_classes) as all other classifiers, or the original one-vs-one ('ovo') decision function of libsvm which has shape (n_samples, n_classes * (n_classes - 1)/2). [ovo, ovr]

Logistic Regression. The parameters that will be tested will be as follows:

- Solver Algorithm to be used in the optimization problem. [liblinear, lbfgs, saga]
- Penalty Used to specify the norm used in the penalization. [l2, none]

- C Inverse of regularization strength; must be a positive float. [1, 4, 10]

Naïves Bayes. The parameters that will be tested will be as follows:

- Alpha Additive (Laplace/Lidstone) smoothing parameter (0 for no smoothing). [0, 1]
- Fit prior Whether to learn class prior probabilities or not. If false, a uniform prior will be used. [True, False]

Random Forest. The parameters that will be tested will be as follows:

- Criterion The function to measure the quality of a split. [gini, entropy]
- Number of estimators is the number of trees in the forest. [50, 100, 200]

K-Nearest Neighbors. The parameters that will be tested will be as follows:

- Number of neighbors Number of neighbors to use by default for kneighbors queries. [5, 15, 40]
- Weights Weight function used in prediction. [uniform, distance]
- Power parameter (p) for the Minkowski metric. When $p = 1$, this is equivalent to using manhattan_distance (l1), and Euclidean distance (l2) for $p = 2$. [1, 2]

4 Finding and Results

Table 1 shows the summary of the overall results by comparing the accuracy and training time. The first row of SVM described the comparison by using different Kernel parameters. Based on the results, we can say that the 'linear' kernel has a higher accuracy compared to 'rbf' and 'poly'. However, the 'rbf' kernel takes less time to train. The second row of SVM shows the results by using different Gamma parameters. The results illustrate that the 'scale' kernel coefficient has the higher accuracy compared to the 'auto' kernel coefficient. But the 'auto' kernel coefficient takes shorter time to train. The last row of SVM shows the results by using different parameters of the Decision Function Shape. The results show no significant difference on the accuracy and the time training of the model.

In this study, we have tested three parameters of Logistic Regression. The first one is the Solver parameters. It did not cause any significant different on the accuracy and the time training of the model. It takes only a few seconds of difference between these solver types. As for the different Penalty parameters, by using the 'l2' penalization, it has a higher accuracy compared to the 'none' penalization. Besides, the 'none' penalization also takes a bit longer to train the model. The C parameters exhibited that using the 'l0' regularization strength has a greater accuracy compared to '1' and '4' regularization strength. However, we also noticed that the greater the regularization strength is, the longer it takes for the model to be trained. We tested two parameters for the Naïve Bayes. The Alpha parameters, when using the additive smoothing one (1) has a greater accuracy compared to the no smoothing zero (0), while for the training time, they are only a few milliseconds apart. When using Fit Prior parameters, if we set the fit prior

to True or False, it would not cause any significant differences on the accuracy and the training time of the model. It only takes a few milliseconds in difference between these solver types.

Table 1. Summary of results.

Machine Learning Model	Parameter Tested				
SVM	Kernel		linear	poly	rbf
		Accuracy	0.76125	0.719625	0.756375
		Training time	12min 33s	11min 3s	7min 3s
	Gamma		scale	auto	
		Accuracy	0.756375	0.49825	
		Training time	8min 23s	2min 57s	
	Decision function shape		One versus one (ovo)	One versus rest (ovr)	
		Accuracy	0.756375	0.756375	
		Training time	6min 13s	6min 2s	
Logistic Regression	Solver		liblinear	lbfgs	Saga
		Accuracy	0.752875	0.752625	0.752875
		Training time	487 ms	3.64 s	641 ms
	Penalty		12	none	
		Accuracy	0.752625	0.735000	
		Training time	1.18s	3.85s	
	Maximum iteration	C	1	4	10
		Accuracy	0.752625	0.75975	0.761000
		Training time	1.34s	5.27s	7.45s
Naïve Bayes	Alpha Additive		0	1	
		Accuracy	0.710500	0.755875	
		Training time	57 ms	29 ms	
	Fit prior		True	False	
		Accuracy	0.755875	0.756125	
		Training time	29 ms	27 ms	
Random Forest	Criterion		gini	entropy	
		Accuracy	0.739250	0.739500	
		Training time	6min 11s	5min 49s	
	Number of estimators		50	100	200
		Accuracy	0.73575	0.736625	0.7425
		Training time	2min 49s	6min 19s	11min 53s
K-Nearest Neighbors	Number of neighbors		5	15	40
		Accuracy	0.546625	0.525375	0.519000
		Training time	5.25 ms	11.4 ms	8.84 ms
	Weights		uniform	distance	
		Accuracy	0.546625	0.5495	
		Training time	8.12 ms	8.69 ms	
	p		1	2	
		Accuracy	0.542875	0.546625	
		Training time	8.01 ms	8.63 ms	

Using Random Forest, we tested two parameters. The criterion parameters also show that, there is no significant difference in accuracy by setting the criterion to either ‘gini’ or ‘entropy’. The ‘gini’ criterion takes a slightly longer time to train the model compared

to the ‘entropy’ criterion. The Different Number of Estimators parameters also presented no significant difference in accuracy by setting the criterion to either ‘50, ‘100’ or ‘200’. However, there is a significant difference for the time taken to train the model, that is the greater the number of estimators, the longer it takes to train the model.

We tested three parameters of K-Nearest Neighbour. We have set a different number of the Neighbors parameters and it shows that, the ‘5’ neighbors have a higher accuracy compared to ‘15’ and ‘40’ neighbors. However, there is no actual pattern in the time taken to train the model, since the result is inconsistent when the number of neighbors increased. Weights parameters shows that when setting the weights to ‘uniform’ or ‘distance’, does not cause any significant difference on the accuracy and the training time of the model.

It takes only a few milliseconds of difference between these weight types. The P parameters shows that by setting P to ‘1 or ‘2, they do not cause any significant difference on the accuracy and the training time of the model. It only takes a few milliseconds of difference between these power parameter types.

5 Conclusion and Recommendation

The main purpose of this study is to find the best hyper parameters setup that can be used for the sentiment analysis for each model. By using SVM, we noticed that it has the highest accuracy compared to other models, however it also takes longer time to train. For the ‘kernel’ parameter, by setting it to linear, it has the highest accuracy compared to other kernel type. But it also causes the model to train longer. Next, for the ‘gamma’ parameter, by setting it to scale, it has a much higher accuracy when compared to the auto scale. However, it made the model to train longer. Furthermore, for the ‘decision function shape’, we do not notice any difference between the various decision function shapes. The difference between the time taken to train the model is also unnoticeable.

Besides, for the Logistic Regression model, we observed that it has one of the highest accuracies compared to other models and takes shorter time to train the model. For the ‘Solver’ parameter, there is no significant difference on the accuracy and the training time of the model. ‘lbfgs’ have a slightly longer time to train the model but in mere seconds. For the ‘penalty’ parameter, we can say that the ‘l2’ penalization has a greater accuracy compared to the ‘none’ penalization. It also causes the model to have a slightly less time to train the model. On top of that, for the regularization strength of the ‘C’ parameters, there is no significant difference on the accuracy performance between the value of the regularization strength. However, we noticed that the higher the regularization strength, the time it takes to train the model will also increase.

Furthermore, for the Naïve Bayes model, overall, we can say that it has a decent performance on the accuracy and the time taken to train the model. For the ‘alpha’ parameters, by setting it to 1 (additive smoothing), causes the model to be extra magnificent compared with no smoothing, 0. Not only that it causes the model to have better accuracy, but also a slight less time to train the model. Next, for the ‘fit prior’ parameters, there is no significant difference on the accuracy and time taken to train the model when setting it up to True or False.

Next, for the model Random Forest, it also has a decent performance on the accuracy, however it is quite the opposite for the time taken to train the model. For the ‘criterion’

parameters in the Random Forest model, even though there is a slight difference on the time taken to train the model, but there is no significant difference on the accuracy either when setting it up to gini or entropy. For the 'number of estimators' parameters, there is also no significant difference in accuracy when setting the criterion to either '50, '100' or '200'. However, there is a significant difference for the time taken to train the model, that is; the higher the number of estimators, the longer it takes to train the model.

Finally, for the model K-Nearest Neighbors, we can say that the accuracy performance of the model is quite terrible compared to other models even though it causes only a few milliseconds to train. For the 'number of neighbors' parameters, we noticed that by setting the number lower, it will cause the model to have a slightly better accuracy on the prediction. However, there is no real pattern on the time taken to train the model. Next for 'weights' parameters, by setting it up to either uniform or distance, there is still no significant difference on the performance of the model. Besides that, by setting the power parameter (P) to '1 or '2, it does not cause any significant differences on the accuracy and the time training of the model. It only takes a few milliseconds in difference between these power parameter types. Therefore, out of all 5 models, we can say that the Naïve Bayes and the Logistic Regression performed extremely well compared to other models. Not only the fact that they have good accuracy, the time they took to train the model are also nominal. However, if the time taken to train the model is not the main concern, the Support Vector Machine would be the best model due to its sharp accuracy. Additionally, we would also say that the K-Nearest Neighbors is not suitable for sentiment analysis due to its below-average accuracy despite having less training time.

However, the result of this study focuses only on one single text processing and vectorizing technique. Therefore, it is recommended for future researchers to try different vectorizing techniques of the text, such as different n-grams of the vector, etc. According to Subarno et.al. (2018), LSTM RNNs are more effective than Deep Neural Networks and conventional RNNs for sentiment analysis. Hence, we would also recommend future researchers to try LSTM RNNs and compare it with different models so that various results could be attained.

References

1. Naresh, A., Venkata Krishna, P.: An efficient approach for sentiment analysis using machine learning algorithm. Evol. Intell. **14**(2), 725–731 (2020). https://doi.org/10.1007/s12065-020-00429-1
2. Singh, J., Singh, G., Singh, R.: Optimization of sentiment analysis using machine learning classifiers. HCIS **7**(1), 1–12 (2017). https://doi.org/10.1186/s13673-017-0116-3
3. Maskat, R., Faizzuddin Zainal, M., Ismail, N., et al.: Automatic labelling of malay cyberbullying twitter corpus using combinations of sentiment, emotion and toxicity polarities. In: ACM International Conference Proceedings Series (2020). https://doi.org/10.1145/3446132.3446412
4. Liu, B.: Sentiment Analysis and Opinion Mining: A Survey. Morgan & Claypool
5. Adam, N.L., Rosli, N.H., Cik Soh, S.: Sentiment analysis on movie review using Naïve Bayes. In: 2nd International Conference on Artificial Intelligence and Data Sciences (AiDAS2021), pp 1–6 (2021)

6. Wah, Y.B., Abdullah, N., Abdul-Rahman, S., Peng Tan, M.L.: Text mining and sentiment analysis on reviews of proton cars in Malaysia. Malaysian J. Sci. **37**, 137–153 (2018). https://doi.org/10.22452/mjs.vol37no2.5
7. Mohamed Shuhidan, S., Hamidi, S.R., Kazemian, S., et al.: Sentiment analysis for financial news headlines using machine learning algorithm. Adv. Intell. Syst. Comput. **739**, 64–72 (2018). https://doi.org/10.1007/978-981-10-8612-0_8
8. Das, S., Kolya, A.K.: Predicting the pandemic: sentiment evaluation and predictive analysis from large-scale tweets on Covid-19 by deep convolutional neural network. Evol Intell. (2021). https://doi.org/10.1007/s12065-021-00598-7
9. Ramya, B.N., Shetty, S.M., Amaresh, A.M., Rakshitha, R.: Smart simon bot with public sentiment analysis for novel Covid-19 tweets stratification. SN Comput. Sci. **2**(3), 1–11 (2021). https://doi.org/10.1007/s42979-021-00625-5
10. Du, J., Xu, J., Song, H., et al.: Optimization on machine learning based approaches for sentiment analysis on HPV vaccines related tweets. J Biomed. Semant. **8**, 1–7 (2017). https://doi.org/10.1186/s13326-017-0120-6
11. Shah, K., Patel, H., Sanghvi, D., Shah, M.: A comparative analysis of logistic regression, random forest and KNN models for the text classification. Augment. Hum. Res. **5**(1), 1–16 (2020). https://doi.org/10.1007/s41133-020-00032-0
12. Kharde, V.A., Sonawane, S.S.: Sentiment analysis of twitter data: a survey of techniques. Int. J. Comput. Appl. **139**, 5–15. (2016). https://doi.org/10.5120/ijca2016908625
13. Yuri, M.N., Mohd Rosli, M.: TelcoSentiment: sentiment analysis on mobile telecommunication services using Naive Bayes technique. In: 2nd International Conference on Information Security and Computer Technology (ICISCT2021), pp 1–8 (2021)
14. Yue, L., Chen, W., Zuo, W., Yin, M.: A survey of sentiment analysis on social media. Knowl. Inf. Syst. **60**, 617–663 (2019). https://doi-org.ezaccess.library.uitm.edu.my/10.1007/s10115-018-1236-4
15. Singh, J., Singh, G., Singh, R.: A review of sentiment analysis techniques for opinionated web text. CSI Trans. ICT **4**(2–4), 241–247 (2016). https://doi.org/10.1007/s40012-016-0107-y
16. Hemmatian, F., Sohrabi, M.K.: A survey on classification techniques for opinion mining and sentiment analysis. Artif. Intell. Rev. **52**(3), 1495–1545 (2017). https://doi.org/10.1007/s10462-017-9599-6
17. Pisner, D.A., Schnyer, D.M.: Support vector machine. Mach. Learn., 101–121 (2020)
18. Al Omari, M., Al Hajj, M., Hammami, N., Sabra, A.: Sentiment classifier: logistic regression for arabic services ' reviews in Lebanon. In: 2019 International Conference on Computer and Information Sciences (ICCIS), pp 1–5. IEEE , (2019)
19. Murphy, K.P.: Naive Bayes classifiers, pp. 1–8 (2006)
20. Breiman, L.: Random Forests. Mach. Learn. **45**, 5–32 (2001)
21. Dey, L., Chakraborty, S., Biswas, A., et al.: Sentiment analysis of review datasets using Naïve Bayes' and K-NN classifier. Int. J. Inf. Eng. Electron Bus. **8**, 54–62 (2016). https://doi.org/10.5815/ijieeb.2016.04.07
22. Jones, A.B.: Sentiment analysis of reviews: Text Pre-processing (2018). https://medium.com/@annabiancajones/sentiment-analysis-of-reviews-text-pre-processing-6359343784fb