Comparison of Naive Bayes and SVM Classification in Grid-Search Hyperparameter Tuned and Non-Hyperparameter Tuned Healthcare Stock Market Sentiment Analysis

KaiSiang Chong, Nathar Shah
Faculty of Computing & Informatics
Multimedia University, Cyberjaya, Selangor, Malaysia

***Abstract*—This paper compares the performance of Naive Bayes and SVM classifiers classification based on sentiment analysis of healthcare companies' stock comments in Bursa Malaysia. Differing from other studies which focus on the performance of the classifier models, this paper focuses on identifying the hyperparameters of the classifier models that are significant for sentiment analysis and the optimization potential of the models. Grid Search technique is used for the hyperparameters tuning process. The performance such as precision, recall, f1-score, and accuracy of Naive Bayes and SVM before and after hyperparameter tuning are compared. The results show that the important hyperparameters for Naive Bayes are alpha and fit_prior, while the important hyperparameters for SVM are C, kernel, and gamma. After performing hyperparameters tuning, SVM gave a better performance with an accuracy of 85.65% than Naive Bayes with an accuracy of 68.70%. It also proves that hyperparameter tuning is able to improve the performance of both models, and SVM has a better optimization potential than Naive Bayes.**

Keywords*—Machine learning; sentiment analysis; opinion mining, Naïve Bayes; SVM Classifier; grid search technique; hyperparameter tuning

I. Introduction

Sentiment analysis, often known as opinion mining, is a natural language processing (NLP) technique that determines the sentiment behind a body of text. This is a common method for businesses to determine and categorize customer views about a product, service, or idea. Data mining, machine learning (ML), and artificial intelligence (AI) are involved in analyzing the texts and finding out the sentiment.

There are too many ways to perform sentiment analysis by using different machine learning algorithms such as Naive Bayes, Support Vector Machine, K-Nearest Neighbour, and so on. This has made it difficult for the researchers to determine which classifier should be used as the performance of these algorithms is usually dependent on the datasets used. Most studies [7][9][3][6][4], concluded that the Naive Bayes and SVM classifiers outperform all other algorithms in evaluating the sentiment of the text. However, it seems that the performance of Naive Bayes and SVM is very similar. Depending on the datasets used, the performance of the classifiers is affected. In [7] and [9], SVM has a better performance than Naive Bayes. The dataset used in [7] is Amazon product reviews, and in [9] is Twitter reviews. However, in [3], Naive Bayes has a better performance than SVM, and the dataset used in [3] is about e-sport education. Most importantly, these researches only used default hyperparameters for the classification models, and so far the best performance of the classifier obtained is from [7], which is the SVM with 84% of accuracy. The problem of current research is that most of the papers use the default hyperparameter for the sentiment classification. The results might be good, but there should still be some potential for the models to perform better.

Hence, the purpose of this paper is to compare the performance of Naive Bayes and SVM classifiers based on sentiment analysis of healthcare companies' stock comments, justify which model is best suited for this case, and justify the optimization potential of the models by hyperparameter tuning using the Grid Search approach. The data is collected from the I3investor website and preprocessed by using text preprocessing techniques such as removing stopwords, lemmatization, tokenization, and so on. After that, the preprocessed data will be used to train the Naive Bayes and SVM, and the results will be evaluated. Section II will include some background studies of several similar works, Section III will be the methodology which includes the detailed steps of conducting the research, and Section IV will be the evaluation results of both classifications.

II. Background Study

Basically, this section of the paper includes the review of several research papers with similar works to ours. According to these research papers, data preprocessing like stopwords removal, stemming, and tokenization are necessary steps before performing the classification. First, the comparison study of Naive Bayes with SVM is reviewed. It appears that the performance of both models is dependent on the datasets used. Second, comes the review of the Naive Bayes classifier and it seems that Naive Bayes outperforms other classifiers. After that, the study about SVM is reviewed, and it shows that the performance of SVM is affected by the dataset used, and using the Grid Search approach for SVM optimization, the performance of SVM can be improved as well.

A. Comparison Study of Naive Bayes and SVM using Different Datasets

Firstly, this section will review the comparative study of Naive Bayes and SVM.

Sanjay Dey et al. [7] conducted a comparison study of two machine learning algorithms for sentiment analysis of Amazon product reviews. Naive Bayes and SVM were used in this paper. The preprocessing steps such as tokenization, removing stopwords, filling missing values, and feature extraction are applied. The result shows that SVM has a slightly better performance with 84 % accuracy than Naive Bayes with 82.875 % accuracy.

Abdul Mohaimin Rahat et al. [9] worked on a research paper to conduct sentiment analysis on the review from Twitter. The dataset collected is preprocessed by stop word removal, hashtag removal, POS tagging, and so on. Two algorithms, which are Naive Bayes and SVM were applied to classify the positive and negative sentiments. As a result, SVM gets a better accuracy of 82.48 % than the Naive Bayes with an accuracy of 76.56 %.

Rian Ardianto et al. [3] performed sentiment analysis toward e-sport education. The data was collected from Twitter. Naive Bayes and SVM are used in this research as a comparative study. Synthetic Minority Over-Sampling Technique (SMOTE) is used in the evaluation of the two algorithms. As a result, Naive Bayes with SMOTE has a better performance with an accuracy of 70.32 % as compared to SVM with SMOTE with an accuracy of 66.92 %.

B. Study on Naive Bayes

Second, this section will review the research on the Naive Bayes classifier.

The research done by Lopamudra Dey et al. [6] focuses on the comparison of two supervised machine learning approaches which are K-Nearest Neighbour and Naive Bayes based on the sentiment analysis of movie reviews as well as hotel reviews. The accuracy, precision, and recall of these models are evaluated. In short, Naive Bayes has a better performance for movie reviews with an accuracy of 82.43 % than K-NN with an accuracy of 69.81 %, while having a similar performance for hotel reviews with an accuracy of 55.09 % as compared to K-NN with an accuracy of 52.14 %. The researchers concluded that Naive Bayes performs better than K-NN in analyzing the movie reviews.

Achmad Bayhaqy et al. [4] focused on comparing three different classification algorithms, which are Decision Tree, K-NN, and Naive Bayes, by the sentiment analysis about the tweets/reviews of E-commerce in Tokopedia and Bukalapak on Twitter. Text preprocessing techniques are applied to the data collected. The comparison of the three algorithms is done with the assistance of Rapidminer. The results show that the accuracy of the Decision Tree is 80%, K-NN is 78%, and Naive Bayes is 77%. The results for precision for Decision Tree is 79.96%, K-NN is 85.67 %, and Naive Bayes is 88.50 %. Although the accuracy of Naive Bayes is 77 % which is the lowest among others, the researchers concluded the Naive Bayes as the most suitable classifiers for use with their datasets as it has the highest precision of 88.50 % which means it provided more accurate and precise predictions.

C. Study on Support Vector Machine

Lastly, this section will review the research on SVM classification.

Munir Ahmad et al. [1] have chosen to use SVM for the sentiment analysis with WEKA. There are two datasets included which are the tweets about self-driving cars and Apple products, and the data are pre-labeled with the sentiments. In short, the accuracy for the self-driving cars dataset is 59.91 %, and the accuracy for the Apple products dataset is 71.2 %. The outcomes are not very good, demonstrating the dependency of SVM performance on the input dataset. The habits of most Twitter users to use short forms or informal language might be the reason for the difficulty for the SVM to learn successfully.

Besides, using the Grid Search approach for SVM optimization, Munir Ahmad et al. [2] have achieved better results. The precision of SVM is increased from around 70% to 80%. With the Twitter data about the topics of Apple, Google, Microsoft, and Twitter, the potential of SVM optimization is highlighted in this paper.

III. Methodology

I3investor is a popular stock investment platform for independent stock traders and investors. Every month, the I3investor [5] community creates over 50K comments and posts. In order to be trained by the supervised learning algorithms, the datasets collected from I3investor needed to be preprocessed and labeled. After the preprocessing and labeling, a portion of data is selected from the dataset and is split into train-set and test-set. The words are vectorized by using the TF-IDF vectorizer, and the dataset is used in running both Naive Bayes and SVM classifiers. Moreover, the Grid Search technique is used for hyperparameter tuning to improve the accuracy of both classifiers. The experimental results are then evaluated. Fig. 1 shows the workflow of the methodology. The details of the process will be explained.

A. Data Pre-processing

Every comment may include some words that are neither significant nor beneficial for sentiment analysis. Hence, text preprocessing is a necessary step to obtain a clean dataset, and have better outcomes. The preprocessing steps included:

1) Removing of URL: The URL in the comments which basically links users to other websites is meaningless for sentiment analysis and is removed.

2) Removing of Other Languages: The data collected will include the comments from Malaysians which means there will be several languages such as Chinese, Malay, and English. Hence, the CLD3 package is used in this case to detect and remove the Chinese and Malay comments, only remain the English comments. Since the classifiers are not trained to assess the sentiment of comments in multiple different languages, the removal of other languages will then have a significant influence on the classification process outcomes.

3) Removing of Punctuation: Punctuation has no value for the sentiment analysis and is removed. It is also a step needed for the ease of tokenization.

4) Removing of Stopwords: Stop words are function words that have no sentiment yet are regularly used. If these terms are not eliminated, they will have no effect on the analysis's efficiency. These words are known as "noise." For example, frequently used terms are "a," "of," "the," "I," "it," "you," and "and."

5) Lemmatization: This stage condenses words into their stem or root forms. For example, “evaluate” and “evaluation”, the root of the word “evaluation” is “evaluate”, and having both terms in the data increases the algorithm's effort to interpret their sentiment. As a result, lemmatizing the token to its root type is required to minimize the complexity of the comment and reduce processing time, hence enhancing the model's performance.

6) Lower casing text: All the text in the datasets is changed to lower case to have a consistent format.

7) Tokenization: A method to divide the entire comment into many individual words for convenience of analysis.

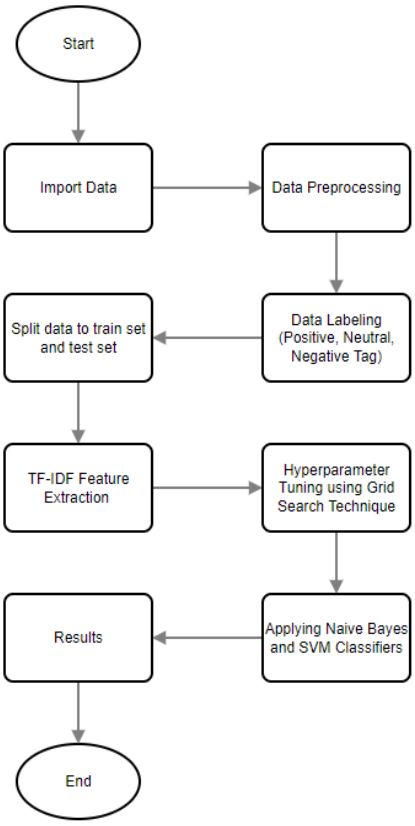

Fig. 1. Methodology workflow.

B. Data Labeling

It is impossible for a human being to manually label the data as the datasets consist of a large number of comments. Hence, the Opinion Lexicon created by Minqing Hu and Bing Liu [8] which contains positive and negative words is being prepared. The preprocessed data is next reviewed to see if the words match those in the positive or negative Opinion Lexicon. The number of positive and negative words for each row of data will next be calculated. The score for each row of data will be computed by subtracting the number of negative words from the number of positive words. As a consequence, data with scores more than 0 will be labeled as positive, data with scores less than 0 will be labeled as negative, and data with scores equal to 0 will be labeled as neutral.

C. Feature Extraction using TF-IDF

The term frequency-inverse document frequency (TF-IDF) was used to extract the feature of the dataset. When retrieving information, the TF-IDF technique weights the frequency of a phrase (TF) and the inverse frequency of documents (IDF). Each word or phrase is given a TF and IDF score. The TF and IDF product results of a word, on the other hand, correspond to the phrase's TF-IDF weight. As a result, the TF-IDF score (weight) rises in tandem with the phrase's rarity and vice versa. As a consequence, the TF of a term denotes its frequency, whereas the IDF denotes its importance across the corpus. If a term's content TFIDF weight is high, the content will always show among the top search results, allowing anybody to avoid stopwords while simultaneously finding words with higher search traffic and lower competition.

D. Hyperparameters Tuning using Grid-Search Technique

Hyperparameters are variables whose values influence the learning process and affect the model parameters that a learning algorithm learns. Grid Search is a technique for optimizing hyperparameters. It prepares the machine learning algorithm for every potential combination of hyperparameters. Cross-validation is used to guide the training process, ensuring that the trained model can extract the majority of the patterns from the dataset. The best set of hyperparameter values from Grid Search is then used in the real model. In summary, the optimal hyperparameters are assured, and the model's accuracy can be enhanced.

IV. Experimental Result

The research is conducted using the Google Colab environment. A portion of the preprocessed data which consists of 20000 comments is used. There are 6219 positive comments, 6196 negative comments, and 7585 neutral comments.

Grid Search approach is used to find out the best hyperparameters of the models. Both original and tuned versions of Naive Bayes and SVM are trained and tested. The precision, recall, f1-score, and accuracy of each model are evaluated.

Table I shows the hyperparameters setting for Naive Bayes. There are three parameters which are alpha, fit_prior, and class_prior. Parameter alpha refers to the additive smoothing parameter, parameter fit_prior control whether to learn class prior probabilities or not, and parameter class_prior refers to the prior probabilities of the classes. The default hyperparameter for alpha is 1.0, and for fit_prior is ‘True.’ After performing the Grid Search, it appears that the best hyperparameter for alpha is 1.4, for fit_prior is ‘False’, and the parameter for class_prior is ‘remain unchanged.’

TABLE I. HYPERPARAMETERS SETTING FOR NAIVE BAYES

Parameters	alpha	fit_prior	class_prior
Default	1.0	True	None
Best	1.4	False	None

TABLE II. HYPERPARAMETERS SETTING FOR SVM

Parameters	C	kernel	gamma
Default	1.0	rbf	scale
Best	7.0	linear	auto

Table II shows the hyperparameters setting for SVM. Basically, SVM has a total of 15 hyperparameters. After performing Grid Search, there are only 3 hyperparameters that have changed which are C from a value of 1.0 to a value of 7.0, the kernel of 'rbf' to kernel of 'linear', and gamma of 'scale' to gamma of 'auto' while the other 12 hyperparameters showing default is the best option to choose. Parameter C refers to the regularization parameter, parameter kernel specified the kernel type to be used, and parameter gamma refers to the coefficient of the kernel. With the other hyperparameters for SVM remaining unchanged, this proves that SVM is already a good model for performing sentiment analysis without tuning the hyperparameter and can usually obtain good performance as in papers [7], [9], [3], and [1].

Precision is a metric used to quantify how many correct positive predictions have been made. It is derived by dividing the number of accurately predicted positive cases by the total number of positive examples predicted. The precision shows the model's accuracy in classifying samples as positive.

Table III shows the comparison of precision for each model before and after hyperparameters tuning. The precision of Naive Bayes has increased from 71.63% to 83.22%, and the precision of SVM has increased from 81.64% to 87.60%. In short, SVM has higher precision than Naive Bayes before and after tuning. Fig. 2 shows the bar chart of precision comparison for Naive Bayes and SVM.

The proportion of valid positive predictions made out of all feasible positive predictions is calculated as recall. The recall metric evaluates the model's ability to detect positive samples. The higher the recall, the more positive samples are discovered.

Table IV shows the comparison of recall for each model before and after hyperparameters tuning. The recall of Naive Bayes has decreased from 79.85% to 75.36%, and the recall of SVM has increased from 86.56% to 88.47%. SVM has a higher recall than Naive Bayes after tuning. Fig. 3 shows the bar chart of recall comparison for Naive Bayes and SVM.

TABLE III. PRECISION COMPARISON FOR NAIVE BAYES AND SVM

Model	Without Tuning	Tuned
Naive Bayes	71.63	83.22
SVM	81.64	87.60

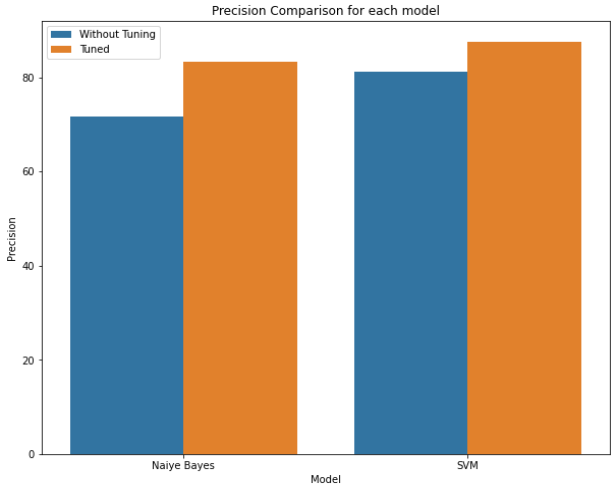

Fig. 2. Bar chart of precision comparison for Naive Bayes and SVM.

TABLE IV. RECALL COMPARISON FOR NAIVE BAYES AND SVM

Model	Without Tuning	Tuned
Naive Bayes	79.85	75.36
SVM	86.56	88.47

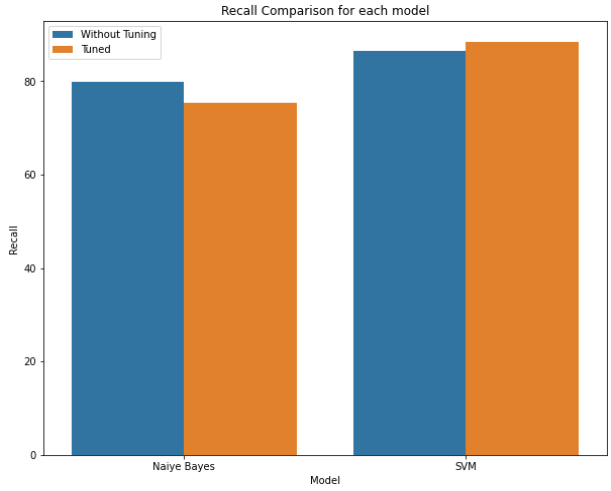

Fig. 3. Bar chart of recall comparison for Naive Bayes and SVM.

The f1-score is a method for combining precision and recall into a single metric that combines both characteristics. We might have good precision with poor recall or vice versa. The f1-score allows you to convey both concerns with a single score.

Table V shows the comparison of f1-score for each model before and after hyperparameters tuning. The f1-score of Naive Bayes has increased from 75.74% to 79.29%, and the f1-score of SVM has increased from 83.90% to 88.04%. SVM has a higher f1-score than Naive Bayes after tuning. Fig. 4 shows the bar chart of the f1-score comparison for Naive Bayes and SVM.

TABLE V. F1-SCORE COMPARISON FOR NAIVE BAYES AND SVM

Model	Without Tuning	Tuned
Naive Bayes	75.74	79.29
SVM	83.90	88.04

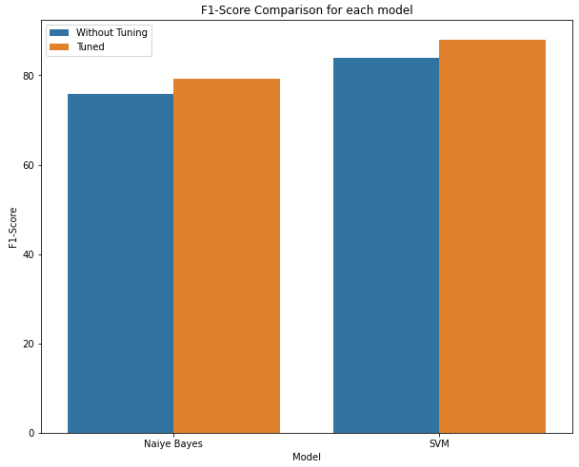

Fig. 4. Bar chart of F1-score comparison for Naive Bayes and SVM.

A model's accuracy is a metric that assesses how well it performs in all classes. This is advantageous when all of the classes are equally important. The ratio between the number of right predictions and the total number of predictions is used to evaluate it.

TABLE VI. ACCURACY COMPARISON FOR NAIVE BAYES AND SVM

Model	Without Tuning	Tuned
Naive Bayes	67.65	68.70
SVM	81.73	85.65

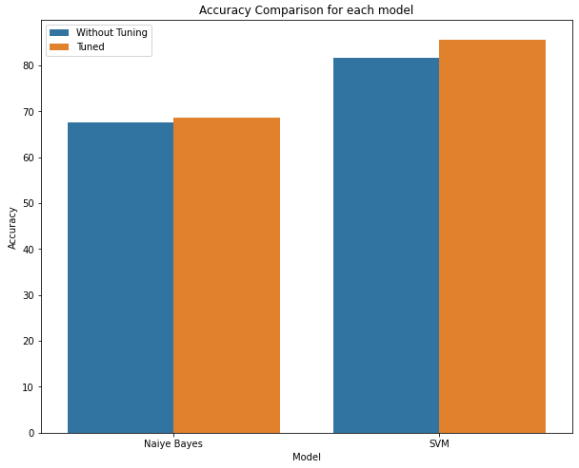

Fig. 5. Bar chart of accuracy comparison for Naive Bayes and SVM.

Table VI shows the comparison of accuracy for each model before and after hyperparameters tuning. The accuracy of Naive Bayes has increased from 67.65% to 68.70%, and the accuracy of SVM has increased from 81.73% to 85.65%. SVM has higher accuracy than Naive Bayes after tuning. Fig. 5 shows the bar chart of accuracy comparison for Naive Bayes and SVM.

In short, the hyperparameters of Naive Bayes that have a significant effect on sentiment analysis are “alpha” and “fit_prior”, while the hyperparameters of SVM that have a significant effect on sentiment analysis are “C”, “kernel”, and “gamma”. SVM has a better performance than Naive Bayes before and after hyperparameters tuning. It appears that SVM has a better potential for optimization with an increase in accuracy of about 4% than the Naive Bayes with an increase in accuracy of about 1%.

V. CONCLUSION

In conclusion, the research has done a comparative study for Naive Bayes and SVM and found out that SVM has a better performance than Naive Bayes based on sentiment analysis of healthcare companies' stock comments. Grid Search approach is used for hyperparameter tuning and is able to identify the hyperparameters of both models that are significant for sentiment analysis. The research is able to prove that hyperparameters tuning can increase the model’s accuracy, and SVM has a better potential for optimization as compared to Naive Bayes. There are still many things to be improved in the future such as adding more datasets, using different classifiers, and using different hyperparameter tuning techniques.

REFERENCES

[1] Ahmad, M., Aftab, S., & Ali, I. (2017). Sentiment analysis of tweets using svm. Int. J. Comput. Appl, 177(5), 25-29.

[2] Ahmad, M., Aftab, S., Bashir, M. S., Hameed, N., Ali, I., & Nawaz, Z. (2018). SVM optimization for sentiment analysis. Int. J. Adv. Comput. Sci. Appl, 9(4), 393-398.

[3] Ardianto, R., Rivanie, T., Alkhalifi, Y., Nugraha, F. S., & Gata, W. (2020). Sentiment analysis on E-sports for education curriculum using naive Bayes and support vector machine. Jurnal Ilmu Komputer dan Informasi, 13(2), 109-122.

[4] Bayhaqy, A., Sfenrianto, S., Nainggolan, K., & Kaburuan, E. R. (2018, October). Sentiment analysis about E-commerce from tweets using decision tree, K-nearest neighbor, and naïve bayes. In 2018 international conference on orange technologies (ICOT) (pp. 1-6). IEEE.

[5] Bursa Malaysia (KLSE) market summary. Bursa Malaysia (KLSE) Market Summary. (n.d.). Retrieved April 2, 2022, from https://klse.i3investor.com/web/index.

[6] Dey, L., Chakraborty, S., Biswas, A., Bose, B., & Tiwari, S. (2016). Sentiment analysis of review datasets using naive bayes and k-nn classifier. arXiv preprint arXiv:1610.09982.

[7] Dey, S., Wasif, S., Tonmoy, D. S., Sultana, S., Sarkar, J., & Dey, M. (2020, February). A comparative study of support vector machine and Naive Bayes classifier for sentiment analysis on Amazon product reviews. In 2020 International Conference on Contemporary Computing and Applications (IC3A) (pp. 217-220). IEEE.

[8] Minqing Hu and Bing Liu. "Mining and Summarizing Customer Reviews." Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004), Aug 22-25, 2004, Seattle, Washington, USA.

[9] Rahat, A. M., Kahir, A., & Masum, A. K. M. (2019, November). Comparison of Naive Bayes and SVM Algorithm based on sentiment analysis using review dataset. In 2019 8th International Conference System Modeling and Advancement in Research Trends (SMART) (pp. 266-270). IEEE.