Multiple Approaches in Sentiment Analysis on Disneyland Reviews

Chun Onn Pam[1], Wen Eng Ong*[2], Amirah Rahman[3]

School of Mathematical Sciences, Universiti Sains Malaysia
[1]pamchun.o@gmail.com
[2]weneng@usm.my
[3]amirahr@usm.my

Article history

Received:
23 May 2022

Received in revised form:
2 June 2022

Accepted:
15 June 2022

Published online:
30 June 2022

*Corresponding author
weneng@usm.my

Abstract

The digital transformation in society has greatly allowed people to have more freedom of speech. The volume for online opinion sharing is growing explosively and irreversibly. Hence, maximizing the utilization of these assets is the key to the success of a business, and sentiment analysis was introduced to conduct studies on the opinions. Sentiment analysis is a great tool to analyze and understand the needs of the customer. There exist multiple approaches in Sentiment analysis including Lexicon-based approach which uses a pre-trained model for unlabeled data, and Learning-based approach which builds a supervised machine learning model for labeled data. Furthermore, there exist multiple techniques to conduct sentiment analysis with these approaches. To estimate the performance of the different techniques, a case study is carried out which focuses on TextBlob and VADER for the Lexicon-based approach and focuses on SVM and Naive Bayes for the Learning-based approach. This study uses reviews posted by visitors on Trip Advisor for three Disneyland Resort Theme Parks. The results indicate that for the Lexicon-based approach, complete sentences with parameter tuning performs better than cleaned sentences without parameter tuning. It was found that VADER and TextBlob perform well on different theme parks. For the Learning-based approach, SVM performed better than the Naive Bayes technique. The Learning-based approach performed better compared to the Lexicon-based approach.

Keywords: *Sentiment Analysis, Lexicon-based approach, Machine Learning, Disneyland reviews*

1. Introduction

Our daily life is full of opinions. Different opinions and perceptions sometimes have different impacts on decision making. Businesses nowadays require a volume of customers, and constructive opinions from customers are essential for businesses to be successful. Some consumers may rely on reviews before making any purchase decisions.

Natural Language Processing (NLP) is a field that allows the computer to understand and interact with human language with the aim to narrow the gap between them and make further interpretation or utilization. Up until the 1980s most NLP systems used the complex hand-written rules, while in the 1990s statistical NLP models arose. Nowadays, NLP systems learn and improve their accuracy using statistical methods such as machine learning or deep learning techniques [1].

Sentiment analysis in NLP allows humans to analyze the mass volume of customer reviews on the internet in a short time to improve business practices. It is used as a tool to analyze customer feedback on specific topics or products [2]. The analysis helps business owners make impactful marketing and business decisions.

Lexicon-based sentiment analysis uses a bag of lexicon words to calculate the polarity of a sentence using the score of each word in the bag. The score for each word is different and follows the rules that the score is higher if the word is stronger. For instances, “best” has higher magnitude compared to “good”. The sentence is tokenized and matched with the dictionary in the bag of lexicon words. A matching for a positive word causes an increment to the sentence score, whereas a matching for a negative word causes a decrement to the sentence score [3]. Different types of rules are needed to mitigate sentences that include elements such as negation and sarcasm. There may also be sentences that do not contain any lexicon words but do express sentiment. These challenges need to be handled with a larger bag of words [4]. Lexicon-based sentiment analysis is also known as unsupervised sentiment analysis [5].

In Learning-based sentiment analysis, a supervised machine learning model is trained to predict the sentiment of the sentence. This approach greatly depends on the training data. A trained model with high metrics scores in one domain might fail to detect the sentiment in another domain.

The dataset used for this study are Disneyland reviews. In 2019, Disneyland Theme Parks were ranked as the second most visited amusement park [6]. The reviews posted by visitors on Trip Advisor for three Disneyland Theme Parks were extracted and posted on Kaggle [7]. We determine the performance of the Lexicon-based sentiment analysis using the reviews from three different Disneyland Theme Park locations. Then, we identify the champion classification model for all three theme park locations. Lastly, the performance of the Lexicon-based approach and the Learning-based approach sentiment analyses for this dataset is compared.

The rest of the paper is organized as follows: Section 2 summarizes the existing literature related to this study. The data source and methodology are presented in Section 3. This is followed by the results and discussion in Section 4. Finally, Section 5 provides the conclusion of this study.

2. Literature Review

Sentiment analysis is greatly used on social data and opinions with different domains such as marketing, medical healthcare, and leisure. Almost every possible aspect of sentiment analysis has been explored. In the economics domain, economic sentiment has been incorporated into a time-series measure in [8]. In addition, [9] demonstrated that economic texts contain personal opinions despite the lack of explicit opinion indicators. Besides that, [10] analyzed the relevant methodological approaches, illustrated empirical results, and presented useful software. In the political domain, [11] presented a procedure for crowdsourcing fine-grained sentiment scores in a language and domain of the user's choice to generate a negative sentiment dictionary. In addition, [12] analyzed the Twitter data set related to the recent 14th Gujarat Legislative Assembly election. For other domains, [13] used sentiment analysis from multiple source domains in a new domain adaptation approach. Besides that, [14] generated context-driven features for the sentiment analysis with specific domain. Using published sentiment data, the sentiment polarity can be automatically determined at the end of news articles or reviews [15]. For the luxury domain, to analyze customer sentiment towards three Disneyland locations (Anaheim, Paris, and Hong Kong), sentiment analysis, emotion detection, and *n*-gram associations were used [16].

For Learning-based sentiment analysis, [17] began training a model with example sentences or statements that are manually annotated as either positive or negative in relation to a specific entity. Based on the difficulty of the task, the integrated combination of information retrieval, natural language processing, and machine learning produced good results. The research in [18] focused on using Support Vector Machine (SVM) and Naive Bayes algorithms to analyze FIFA cups tweets in the Portuguese language. Furthermore, the Naive Bayes and SVM algorithms were compared by collecting airline reviews based on sentiment analysis in [19]. Both studies showed that SVM outperformed the Naïve Bayes algorithm in sentiment analysis.

For Lexicon-based sentiment analysis, a comparison of sentiment lexicons, namely W-WSD, SentiWordNet, and TextBlob, is made to optimize sentiment analysis by identifying the most appropriate lexicon [20]. In 2015, a Lexicon-based sentiment analysis algorithm focusing on real-time Twitter content analysis was developed [21]. To estimate sentiment intensity rather than positive or negative labels, the mixed sentiment classification process has two key components: sentiment normalization and evidence-based combination functions. In this new approach, sentiment is normalized, allowing us to measure sentiment intensity rather than the strength of positivity or negativity. It was discovered that a mixed sentiment message can be improved by developing an evidence-based combining function. In 2018, [22] classified tweets into two categories: Positive and Negative. They calculated the semantic score from each tweet. A positive or negative score indicates whether the tweet is positive or negative. A variable called score is used to store the difference between positive and negative words in a sentence. In terms of polarity, the score indicates whether the sentence is positive or negative. A positive score indicates a positive sentence, otherwise it indicates a negative one.

Most of the previous studies focus on one of the approaches, either Lexicon-based, Learning-based or hybrids, but do not compare among the approaches. In this study, we will analyze and compare the results using both approaches and determine the best approach for Disneyland reviews dataset [7].

Data and Methodology

3.1 Data Source

This study analyzes 42000 reviews posted by visitors on Trip Advisor for three Disneyland Theme Park locations; Hong Kong, California and Paris. The reviews are partitioned into three datasets according to the locations and analyses are done separately. The attributes involved in this dataset are summarized in Table 1.

Table 1: The Summary of Disneyland Review Dataset

No	Attribute Name	Attribute Description
1	Review_ID	Unique ID for each review
2	Rating	Reviewer's rating of the theme park ranging from unsatisfied (1) to satisfied (5).
3	Year_Month	Year and months when the reviewer visited the theme park
4	Reviewer	Reviewer's origin country
5	Review_Text	Reviewer's comments about the visited theme park
6	Disneyland_Branch	Location of the theme park.

3.2 Data Processing, Visualization and Text Processing

To ensure a smooth analysis, data cleaning is performed to detect and correct corrupted or problematic data such as missing data or data duplication. Data visualization is performed using Seaborn, Matplotlib and WordCloud Python libraries to provide insights about the dataset. Text processing is performed to reduce the noise in the dataset. Case lowering is conducted to ensure the computer does not misclassify the capitalized and non-capitalized words. In addition, punctuation marks, hashtags, "@", "https", "http", and html formats are removed. Tokenization is used to break sentences into words, while lemmatization take words back to their base forms using a set of lexical databases for the English language called WordNet. Stop words are commonly used words or words that do not have significant meaning to the sentence. They are removed after tokenization and lemmatization to reduce the noise and reduce the time needed to build the model. Stop words are matched and removed using the NLTK library word list with some additional common words such as "Disney", "Disneyland", "hk", "California", "paris", "hongkong", "hong kong" and "one" [23].

3.3 Lexicon-based Sentiment Analysis

For Lexicon-based sentiment analysis, we examine the sentences with the open-source packages pre-built model in TextBlob and VADER library. We perform the classification of the sentiments based on the computed polarity scores. The results for before and after text processing are compared. Furthermore, parameter tuning is performed on the range for 'Neutral' sentiments classes.

Using a trained Lexicon-based model in TextBlob [24], a polarity score ranges from -1 to +1 will be returned. -1 indicates that the word is very negative whereas +1 indicates that the word is very positive. The sentence is tokenized and polarity score is computed by taking the average polarity on all the single words in the sentence. Furthermore, if negations exist in the text, the polarity is multiplied with -0.5. TextBlob handles grammatical modifiers and intensifiers, by ignoring the polarity if the modifiers do exist in the sentence [25].

The lexical dictionary of VADER contains approximately 7500 sentiment features and is rated within the range -4 to +4, which indicates extremely negative to extremely positive [26]. VADER considers the negation that changes the polarity and intensifiers in the sentence. Furthermore, VADER considers capitalization of words. For instance, "HAPPY" is more positive than "happy". Also, punctuation marks such as "!!!!!!!" can increase the intensity magnitude. VADER computes the polarity of the sentence by taking the summation of sentiment score for all words in the sentence. Then, the final score of the polarity is then normalized to the range of -1 to +1 with $\alpha = 15$ as shown in equation (1) [27],

$$\frac{x}{\sqrt{x^2+\alpha}}, \tag{1}$$

where x is the summation of sentiment score for words in a sentence.

3.4 Learning-based Sentiment Analysis

For Learning-based sentiment analysis, the supervised machine learning models Support Vector Machine (SVM) and Naïve Bayes are compared. The ratings are classified into 3 different sentiments: 'Positive' for ratings 4 and 5, 'Neutral' for rating 3 and 'Negative' for ratings 1 and 2. The dataset for each location is split into 75% training data and 25% testing data. Then, the sentiments are converted from strings to integers: 'Positive' as 1, 'Neutral' as 0 and 'Negative' -1. Term frequency-inverse document frequency (TF-IDF) vectorization [28] is applied on the text data to obtain the frequency for model building. The formulae for TF and IDF are given in equation (2) and equation (3) respectively.

$$TF(w,d) = \frac{\text{Number of } w \text{ in } d}{\text{Number of words in } d}, \tag{2}$$

$$IDF(w) = \ln\left(\frac{N}{\text{Number of documents with } w}\right), \tag{3}$$

where w is the chosen word, d is the document and N is the total number of documents.

Naive Bayes is a probabilistic classifier with independent assumptions among the predictors based on the Bayes Theorem that determines the conditional probability. Equation (4) shows the formula for the Bayes Theorem.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \tag{4}$$

Naive Bayes can perform well for multiple class classification if the model satisfies the independence predictors and normally distributed assumptions. From equation (4), the probability of sentiments is calculated and normalized to obtain the likelihood of each sentiment. Then the sentiment is concluded for each text with the greatest percentage [29].

SVM [30] aims to obtain a hyperplane which can separate the data into different classes optimally by maximizing the distance from the hyperplane to the closest data from different classes. In SVM, there exist different type of Kernel functions such as linear, polynomial and radius basis function (RBF) that can separate the classes non-linearly. Hence, SVM performs well on complex datasets.

3.5 Performance Measures

Accuracy, recall, precision, and $F1$-score are used to evaluate the performance of the model. These metrics are computed based on the values of True Positive (TP), True Negative (TN), False Negative (FN), and False Positive (FP). TP is the outcome of the model predicted positive on the positive class, while TN is the outcome of the model predicted negative on the negative class. FN is the outcome of the model predicted negative on positive class, while FP is the outcome of the model predicted positive on the negative class. Equations (5), (6), (7) and (8) are the formulae for accuracy, recall, precision and $F1$-score respectively.

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP}. \tag{5}$$

$$Recall = \frac{TP}{TP+FN}. \tag{6}$$

$$Precision = \frac{TP}{TP+FP}. \tag{7}$$

$$F1 - score = \frac{2\times TP}{(2\times TP)+FN+FP}. \tag{8}$$

4. Results and Discussion

Figure 1 shows the count plots of sentiment ratings for the three Disneyland Theme Park locations. All three locations have different data sizes and are imbalanced with higher counts of positive sentiment and lower counts of negative sentiment. Hence, $F1$-score will be used to determine the best model as it sums up the predictive performance of a model by combining the recall and precision.

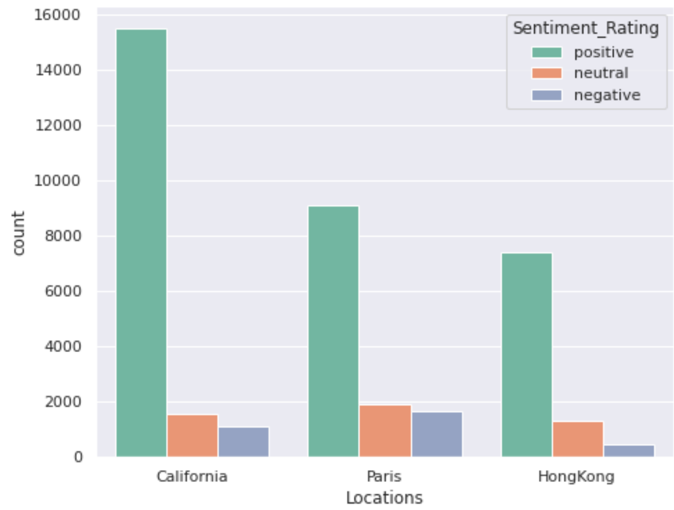


Figure 1: Count Plots for Different Branches with Sentiment Rating

4.1 Lexicon-based Sentiment Analysis

For Lexicon-based sentiment analysis, the counts for both TextBlob and VADER were obtained by tuning the parameter on the range of 'Neutral' sentiment. The parameter values chosen are similar to the level of significance in statistics: ± 0.01, ± 0.05, and ± 0.10. The range of polarity scores is $[-1, 1]$. If the range of 'Neutral' sentiment is $[0, 0]$, that means score within $[-1, 0)$ belongs 'Negative' sentiment, 0 is 'Neutral' sentiment, and $(0, 1]$ belongs to 'Positive' sentiment. If the range of 'Neutral' sentiment is $[-0.05, 0.1]$, that indicates score within $[-1, -0.05)$ is 'Negative' sentiment, $[-0.05, 0.1]$ is 'Neutral sentiment, and $(0.1, 1]$ is 'Positive' sentiment. The counts on the sentiment are compared between full sentences and text-processed sentences (cleaned sentences), and are given in Appendix 1 (TextBlob) and Appendix 2 (VADER).

Table 2, Table 3, and Table 4 show the performance measure comparisons for Hong Kong, California, and Paris locations. All three tables show that cleaned sentence models have lower $F1$-score as compared to full sentence models for both TextBlob and VADER.

From Table 2, for the Hong Kong location dataset, models with parameter tuning have higher $F1$-score compared to models without parameter tuning, and VADER has better performance metrics compared to TextBlob. Text processing is not necessary as full sentences do have better performance and parameter tuning can greatly increase the $F1$-score for the model. Thus, the champion model for the Hong Kong location is VADER without text processing and with parameter tuning, which yields with the highest $F1$-score model, of 76.56%.

From Table 3, for the California location dataset, VADER has better performance without parameter tuning. Conversely, TextBlob has better performance with parameter tuning. The performance of parameter tuning is different for both

TextBlob and VADER in this case. The champion model here is TextBlob without text processing and with parameter tuning yielding the $F1$-score of 80.88%.

Table 2: Performance Measure for TextBlob and VADER Sentiment Analysis with Full and Cleaned Sentences and Parameter Tuning – Hong Kong Location

	TextBlob				VADER			
	Full		Cleaned		Full		Clean	
	[0, 0]	[−0.05, 0.1]	[0, 0]	[−0.05, 0.1]	[0, 0]	[−0.5, 0.5]	[0, 0]	[−0.1, 0.5]
1	0.7938	0.7493	0.7878	0.7408	0.7949	0.7689	0.7966	0.7639
2	0.7938	0.7493	0.7878	0.7408	0.7949	0.7689	0.7966	0.7639
3	0.7217	0.7645	0.7129	0.7573	0.7290	0.7629	0.7239	0.7412
4	0.7430	0.7565	0.7392	0.7486	0.7469	**0.7656**	0.7435	0.7515

** 1 = Accuracy, 2 = Recall, 3 = Precision, 4 = $F1$-score

Table 3: Performance Measure for TextBlob and VADER Sentiment Analysis with Full and Cleaned Sentences and Parameter Tuning – California Location

	TextBlob				VADER			
	Full		Cleaned		Full		Clean	
	[0, 0]	[−0.01, 0.05]	[0, 0]	[−0.01, 0.05]	[0, 0]	[−0.5, 0.5]	[0, 0]	[−0.05, 0.5]
1	0.8334	0.8144	0.8285	0.8099	0.8320	0.7870	0.8369	0.7879
2	0.8334	0.8144	0.8285	0.8099	0.8320	0.7870	0.8369	0.7879
3	0.7785	0.8040	0.7794	0.8008	0.7928	0.8136	0.7853	0.8006
4	0.8029	**0.8088**	0.8004	0.8049	0.8079	0.7992	0.8068	0.7950

** 1 = Accuracy, 2 = Recall, 3 = Precision, 4 = $F1$-score

Table 4: Performance Measure for TextBlob and VADER Sentiment Analysis with Full and Cleaned Sentences and Parameter Tuning – Paris Location

	TextBlob				VADER			
	Full		Cleaned		Full		Clean	
	[0, 0]	[0, 0.10]	[0, 0]	[0, 0.10]	[0, 0]	[−0.01, 0.50]	[0, 0]	[0, 0.50]
1	0.7361	0.7026	0.7314	0.6943	0.7390	0.7145	0.7342	0.7088
2	0.7361	0.7026	0.7314	0.6943	0.7390	0.7145	0.7342	0.7088

3	0.6302	0.7103	0.6274	0.6997	0.6516	0.6722	0.6382	0.6595
4	0.6687	**0.7057**	0.6654	0.6965	0.6788	0.6897	0.6667	0.6782

** 1 = Accuracy, 2 = Recall, 3 = Precision, 4 = $F1$-score

Similar to the California location dataset, TextBlob performed better after parameter tuning for the Paris location dataset, but VADER has performed better without parameter tuning. By considering models with parameter tuning and full sentences, TextBlob performed better compared to VADER. Thus, the champion model here is TextBlob without text processing and with parameter tuning, which yield the highest $F1$-score model of 70.57%.

In conclusion, all 3 champion models are without text processing and with parameter tuning. Furthermore, 2 out of 3 champion models are obtained from TextBlob. Hence, for this dataset, TextBlob is better in performance and text processing is unnecessary. Parameter tuning can greatly increase the performance for the uneven classes distributed dataset.

4.2 Learning-based Sentiment Analysis

For Learning-based approach results, the confusion matrices are given in Appendix 3. For all three locations, Naïve Bayes model is better in predicting positive sentiment but bad at predicting negative sentiment. It predicts almost all text as positive. SVM has better prediction on neutral and negative sentiments compared to Naïve Bayes.

Table 5 shows the performance measures for the Hong Kong location. All performance measures for SVM are better than for Naïve Bayes. This indicates that generally SVM has better performance compared to Naïve Bayes. Furthermore, we observe that the precision for Naïve Bayes is low, which indicates that the count for false positives is high, we can infer that this model over classifies text into positive sentiments. Hence, the champion model for the Hong Kong location is SVM.

Table 5: Performances Metrics for Naïve Bayes and SVM – Hong Kong Location

	Naïve Bayes	SVM
Accuracy	0.8077	0.8257
Recall	0.8077	0.8257
Precision	0.6777	0.7839
$F1$-score	0.7235	0.7895

SVM also performed better in classification compare to Naïve Bayes for the California location dataset (See Table 6). Both models obtained approximately 80% for all measures except for Naïve Bayes' $F1$-score. The champion model for the California location dataset is SVM.

Table 6: Performances Metrics for Naïve Bayes and SVM – California Location

	Naïve Bayes	SVM
Accuracy	0.8538	0.8769
Recall	0.8538	0.8769
Precision	0.8029	0.8354
$F1$-score	0.7883	0.8468

All performance measures for the Paris location dataset are shown in Table 7. It can be observed that SVM outperforms Naïve Bayes here as well. By referring to Appendix 3, Naïve Bayes for the Paris location dataset has higher count on non-positive prediction compared to Naïve Bayes for Hong Kong and California location datasets, but the model performance in terms of $F1$-score is the worst. The champion model for the California location dataset is SVM.

Table 7: Performances Metrics for Naïve Bayes and SVM – Paris Location

	Naïve Bayes	SVM
Accuracy	0.7337	0.8059
Recall	0.7337	0.8059
Precision	0.7174	0.7766
$F1$-score	0.6387	0.7850

In a nutshell, for Learning-based approach sentiment analysis SVM is the champion model for all the three location datasets, where SVM surpassed Naïve Bayes performance wise.

4.4 Comparison between Lexicon-based and Learning-based approaches

To determine the best approach for our three datasets, a comparison using $F1$-score for the champion models is made. However, for Lexicon-based approach, the whole dataset is used for the performance measures. For Learning-based approach, the train-test splitting technique is necessary as part of the machine learning process. With the splitting of 75% training set and 25% testing set, we build the model using the training set and obtain the performance measures for the testing set.

Table 8: $F1$-score for Champion Models

	Lexicon-based		Learning-based	
	$F1$-score	Model	$F1$-score	Model
Hong Kong	0.7657	VADER, without text processing, parameter tuning	0.7895	SVM
California	0.8088	TextBlob, without text processing, parameter tuning	0.8468	SVM
Paris	0.7057	TextBlob, without text processing, parameter tuning	0.7850	SVM

Table 8 shows the comparison of $F1$-score for both Lexicon-based approach and Learning-based approach models. Learning-based models performs better compared to Lexicon-based models. Thus, we can conclude that supervised machine learning models outperform pre-trained Lexicon-based model for this Disneyland reviews dataset.

5. Conclusion

In conclusion, for Lexicon-based approach, TextBlob perform slightly better than VADER on this Disneyland reviews dataset. Furthermore, syntax in TextBlob is easier compared to VADER. It is more beginner-friendly, and has faster execution time. In general, using the full sentence for TextBlob and VADER yielded a slightly higher $F1$-score. Moreover, performing parameter tuning yield a better $F1$-score for the uneven distributed class, but sometimes will lower the accuracy of the prediction. Hence, we can conclude that text processing is unnecessary and parameter tuning is needed when performing sentiment analysis for this dataset.

For the Learning-based approach, SVM outperformed Naïve Bayes for all three locations dataset, and this indicates that the SVM is good at handling complex data. However, building an SVM model requires a longer execution time compared to Naïve Bayes.

Interestingly, the Learning-based approach performed better than the Lexicon-based approach. Lexicon-based approach packages such as VADER and TextBlob are good at handling unlabelled sentiment data but weaker at handling labelled sentiment data. If labeled sentiment data is given, Learning-based approach such as building a supervised classification model is better for future prediction. Hence, we can conclude that SVM is the champion approach for this Disneyland reviews.

One of the limitations of this study is that the comparison of $F1$-score between the Lexicon-based and Learning-based approaches is made based on different data. The Lexicon-based approach used the whole dataset, while the Learning-based approach used a partial dataset. Furthermore, execution time for model building was not considered as a factor when determining the champion model. For future work, TF-IDF vectorization could be implemented with different kinds of parameters and grid search is recommended to implement a more systematic parameter tuning for the model. Deep learning models such as neural network and its variations can be implemented for comparison purposes.

References

[1] Foote, K.D., 2019. A brief history of natural language processing (nlp).

[2] Wolff, R. (2021), Voice of Customer (VoC) Analysis for Real-Time Results [Online], [Accessed 19th January 2022]. Available from the World Wide Web: https://monkeylearn.com/blog/voice-of-customer-analysis/

[3] Priyavrat, A.J., 2017. Sentiment Analysis: A Comparative Study of Supervised Machine Learning Algorithms Using Rapid miner. International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN, pp.2321-9653.

[4] El Alaoui, I., Gahi, Y., Messoussi, R., Chaabi, Y., Todoskoff, A. and Kobi, A., 2018. A novel adaptable approach for sentiment analysis on big social data. Journal of Big Data, 5(1), pp.1-18.

[5] VADER, IBM Watson or TextBlob: Which is Better for Unsupervised Sentiment Analysis? (2019) [Online], [Accessed 20th January 2022]. Available from the World Wide Web: https://intellica-ai.medium.com/vader-ibm-watson-or-textblob-which-is-better-for-unsupervised-sentiment-analysis-db4143a39445

[6] Rubin, J. (2018;2020), Global Attractions Attendance Report - aecom.com [Online], [Accessed 18th January 2022]. Available from the World Wide Web: https://aecom.com/wp-content/uploads/documents/reports/AECOM-Theme-Index-2020.pdf

[7] Chillar, A. (2021), Disneyland Reviews [Online], [Accessed 23th January 2022]. Available from the World Wide Web: https://www.kaggle.com/arushchillar/disneyland-reviews

[8] Shapiro, A.H., Sudhof, M. and Wilson, D.J., 2020. Measuring news sentiment. Journal of econometrics.

[9] Musat, C.C. and Trausan-Matu, S., 2009. Sentiment Analysis on Economic Texts. WOMSA'09, pp.71-78.

[10] Algaba, A., Ardia, D., Bluteau, K., Borms, S. and Boudt, K., 2020. Econometrics meets sentiment: An overview of methodology and applications. Journal of Economic Surveys, 34(3), pp.512-547.

[11] Haselmayer, M. and Jenny, M., 2017. Sentiment analysis of political communication: combining a dictionary approach with crowdcoding. Quality & quantity, 51(6), pp.2623-2646

[12] Bose, R., Dey, R.K., Roy, S. and Sarddar, D., 2019. Analyzing political sentiment using Twitter data. In Information and communication technology for intelligent systems (pp. 427-436). Springer, Singapore.

[13] Wu, F. and Huang, Y., 2016, August. Sentiment domain adaptation with multiple sources. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 301-310).

[14] Choi, Y., Kim, Y. and Myaeng, S.H., 2009, November. Domain-specific sentiment analysis using contextual feature generation. In Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion (pp. 37-44).

[15] Yan, L. and Zhang, Y., 2012, December. News Sentiment Analysis Based on Cross-Domain Sentiment Word Lists and Content Classifiers. In International Conference on Advanced Data Mining and Applications (pp. 577-588). Springer, Berlin, Heidelberg

[16] Cruz, M. D. (2021), Analyzing Disneyland Reviews using NLP [Online], [Accessed 24th January 2022]. Available from the World Wide Web: https://towardsdatascience.com/analyzing-disneyland-reviews-b916b6dcccf4

[17] Boiy, E. and Moens, M.F., 2009. A machine learning approach to sentiment analysis in multilingual Web texts. Information retrieval, 12(5), pp.526-558.

[18] Firmino Alves, A.L., Baptista, C.D.S., Firmino, A.A., Oliveira, M.G.D. and Paiva, A.C.D., 2014, November. A Comparison of SVM versus naive-bayes techniques for sentiment analysis in tweets: A case study with the 2013 FIFA confederations cup. In Proceedings of the 20th Brazilian Symposium on Multimedia and the Web (pp. 123-130).

[19] Rahat, A.M., Kahir, A. and 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.

[20] Gupta, N. and Agrawal, R., 2020. Application and techniques of opinion mining. In Hybrid Computational Intelligence (pp. 1-23). Academic Press.

[21] Jurek, A., Mulvenna, M.D. and Bi, Y., 2015. Improved Lexicon-based sentiment analysis for social media analytics. Security Informatics, 4(1), pp.1-13.

[22] Nigam, N. and Yadav, D., 2018, April. Lexicon-based approach to sentiment analysis of tweets using R language. In International Conference on Advances in Computing and Data Sciences (pp. 154-164). Springer, Singapore.

[23] Deepanshi (2021), Text Preprocessing NLP: Text Preprocessing in NLP with Python codes [Online], [Accessed 24th January 2022]. Available from the World Wide Web: https://www.analyticsvidhya.com/blog/2021/06/text-preprocessing-in-nlp-with-python-codes/

[24] Sloria (2014), TextBlob/en-sentiment.xml at eb08c120d364e908646731d60b4e4c6c1712ff63·sloria/TextBlob [Online], [Accessed 25th January 2022]. Available from the World Wide Web: https://github.com/sloria/TextBlob/blob/eb08c120d364e908646731d60b4e4c6c1712ff63/textblob/en/en-sentiment.xml

[25] Fahad, A. H. (2021), Sentiment Analysis-Let TextBlob do all the Work [Online], [Accessed 25th January 2022]. Available from the World Wide Web: https://medium.com/red-buffer/sentiment-analysis-let-textblob-do-all-the-work-9927d803d137

[26] Ma, Y. (2020), NLP: How does NLTK.Vader Calculate Sentiment? [Online], [Accessed 25th January 2022]. Available from the World Wide Web: https://medium.com/ro-codes/nlp-how-does-nltk-vader-calculate-sentiment-6c32d0f5046b

[27] Beri, A. (2020), SENTIMENTAL ANALYSIS USING VADER [Online], [Accessed 26th January 2022]. Available from the World Wide Web: https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664

[28] Jayaswal, V. (2020), Text Vectorization: Term Frequency-Inverse Document Frequency (TFIDF) [Online], [Accessed 26th January 2022]. Available from the World Wide Web: https://towardsdatascience.com/text-vectorization-term-frequency-inverse-document-frequency-tfidf-5a3f9604da6d

[29] Ray, S. (2017), Learn Naive Bayes Algorithm: Naive Bayes Classifier Examples [Online], [Accessed 26th January 2022]. Available from the World Wide Web: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

[30] Ray, S. (2017), SVM: Support Vector Machine Algorithm in Machine Learning [Online], [Accessed 26th January 2022]. Available from the World Wide Web: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector- machine-example-code/

Appendix 1

Table A: TextBlob Sentiment Count

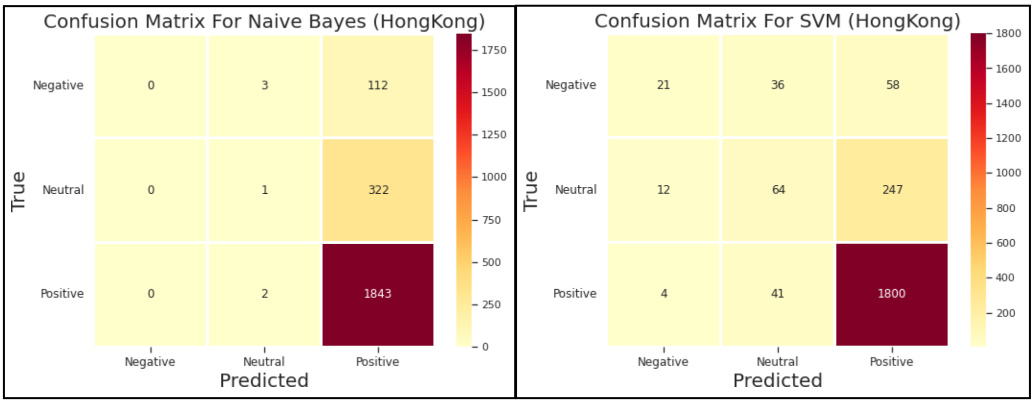
Hong Kong						
Full Sentences			Neutral Range	Cleaned Sentences		
Negative	Neutral	Positive	Parameter	Negative	Neutral	Positive
459	1294	7378	**Original**	459	1294	7378
742	61	8328	**[0, 0]**	738	95	8298
690	173	8268	**[−0.01, 0.01]**	687	200	8244
467	763	7901	**[−0.05, 0.05]**	**482**	793	7856
282	1701	**7148**	**[−0.10, 0.10]**	316	1683	**7132**
467	**1516**	**7148**	**[−0.05, 0.10]**	**482**	**1517**	**7132**
California						
Full Sentences			Neutral Range	Cleaned Sentences		
Negative	Neutral	Positive	Parameter	Negative	Neutral	Positive
1135	1551	15506	**Original**	1135	1551	15506
1405	153	16634	**[0, 0]**	1425	216	16551
1277	418	16497	**[−0.01, 0.01]**	**1306**	450	16436
843	1606	**15743**	**[−0.05, 0.05]**	857	1627	**15708**
525	3270	14397	**[−0.10, 0.10]**	519	3315	14358
1277	**1172**	**15743**	**[−0.01, 0.05]**	**1306**	**1178**	**15708**
Paris						
Full Sentences			Neutral Range	Cleaned Sentences		
Negative	Neutral	Positive	Parameter	Negative	Neutral	Positive
1672	1933	9086	**Original**	1672	1933	9086
1403	50	11238	**[0, 0]**	**1436**	68	11187
1271	309	11111	**[−0.01, 0.01]**	1326	319	11046
828	1515	10348	**[−0.05, 0.05]**	849	1547	10295
481	3196	**9014**	**[−0.10, 0.10]**	488	3153	**9050**
1403	**2274**	**9014**	**[0, 0.10]**	**1436**	**2205**	**9050**

Appendix 2

Table B: VADER Sentiment Count

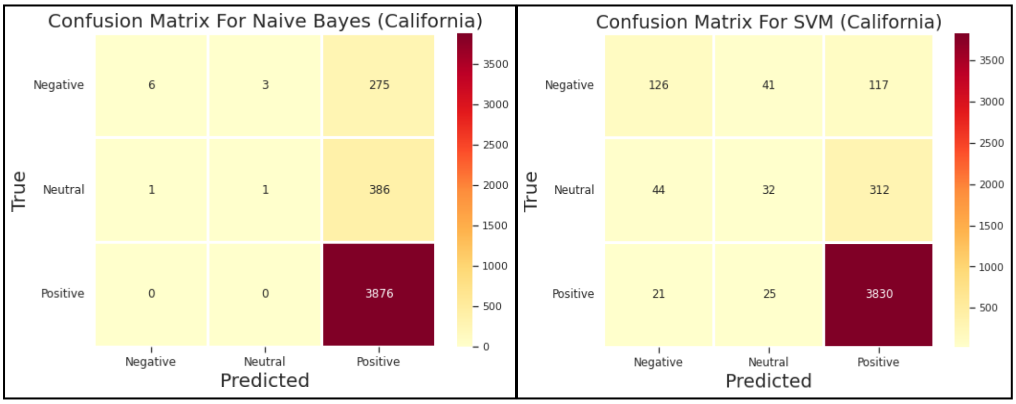
Hong Kong						
Full Sentences			**Neutral Range**	**Cleaned Sentences**		
Negative	**Neutral**	**Positive**	**Parameter**	**Negative**	**Neutral**	**Positive**
459	1294	7378	**Original**	459	1294	7378
736	105	8290	**[0, 0]**	530	112	8489
731	114	8286	**[−0.01, 0.01]**	527	120	8484
713	161	8257	**[−0.05, 0.05]**	499	169	8463
678	228	8225	**[−0.10, 0.10]**	**464**	236	8431
351	**1292**	**7488**	**[−0.50, 0.50]**	208	1204	**7719**
678	965	7488	**[−0.1, 0.5]**	**464**	**948**	**7719**
California						
Full Sentences			**Neutral Range**	**Cleaned Sentences**		
Negative	**Neutral**	**Positive**	**Parameter**	**Negative**	**Neutral**	**Positive**
1135	1551	15506	**Original**	1135	1551	15506
1675	233	16284	**[0, 0]**	1249	245	16698
1668	244	16280	**[−0.01, 0.01]**	1237	264	16691
1622	348	16222	**[−0.05, 0.05]**	1169	374	16622
1574	473	16145	**[−0.10, 0.10]**	**1118**	532	16542
972	**2307**	**14913**	**[−0.50, 0.50]**	573	2375	**15244**
1574	1705	14913	**[−0.10, 0.50]**	**1118**	**1830**	**15244**
Paris						
Full Sentences			**Neutral Range**	**Cleaned Sentences**		
Negative	**Neutral**	**Positive**	**Parameter**	**Negative**	**Neutral**	**Positive**
1672	1933	9086	**Original**	1672	1933	9086
1685	107	10899	**[0, 0]**	1230	109	11352
1679	116	10896	**[−0.01, 0.01]**	1226	120	11345
1646	176	10869	**[−0.05, 0.05]**	1196	189	11306
1607	262	10822	**[−0.10, 0.10]**	1147	301	11243
1121	1583	9987	**[−0.50, 0.50]**	712	1598	10381
1679	**1025**	**9987**	**[−0.01, 0.50]**	1226	1084	10381
1685	1019	9987	**[0, 0.50]**	**1230**	**1080**	**10381**

Appendix 3



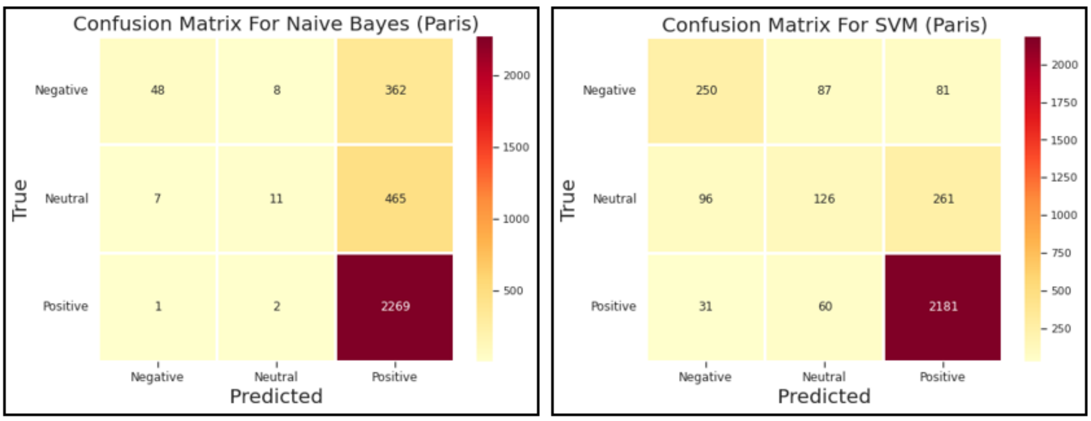
(a) (b)

Figure A: 3×3 Confusion Matrix for Hong Kong, (a) Naïve Bayes and (b) SVM.



(a) (b)

Figure B: 3×3 Confusion Matrix for California, (a) Naïve Bayes and (b) SVM.



(a) (b)

Figure C: 3×3 Confusion Matrix for Paris, (a) Naïve Bayes and (b) SVM.