Predicting Rating Polarity through Automatic Classification of Review Texts

Gregorius Satia Budhi
The University of Newcastle, Australia;
Petra Christian University, Indonesia
gregorius.satiabudhi@uon.edu.au

Raymond Chiong
The University of Newcastle, Australia
raymond.chiong@newcastle.edu.au

Ilung Pranata
The University of Newcastle, Australia
ilung.pranata@newcastle.edu.au

Zhongyi Hu
Wuhan University, China
zhongyi.hu@whu.edu.cn

***Abstract*—Online reviews and ratings are important for potential customers when deciding whether to purchase a product or service. However, reading and synthesizing the massive amount of review data, which is often unstructured, is a huge challenge. In this study, we investigate the use of machine learning models to predict rating polarity (positive, neutral or negative) through automatic classification of review texts. We apply various single and ensemble classifiers to identify rating polarity of reviews from the 2017 Yelp dataset. Experimental results show that the linear kernel Support Vector Machine, Logistic Regression and Multilayer Perceptron are among the three best single classifiers in terms of accuracy, precision, recall and F-measure. Their performances can be further improved when used as base classifiers for ensemble models.**

Keywords—Big data; customer reviews and ratings; classification; machine learning; text mining.

I. Introduction

Post-purchase customer ratings and reviews have become pivotal elements for today's e-commerce. Many potential customers rely on information provided by ratings and reviews prior to making a purchase decision. Research shows that customers normally read product reviews provided by others to understand the reliability and usefulness of a product that they are about to purchase [1]. This, in turn, generates implied trust on the product [2, 3]. The ability to identify relevant content in a short time period helps both consumers and sellers to make proper decisions quickly. This is particularly useful when applied to online sites and social media [4]. The rapid proliferation of web and social media sites, e.g., review websites, forum discussions, blogs, micro-blogs, Twitter, and many others, produces a huge volume of data that is unprecedented. Nowadays, consumers do not solely rely on families or friends to get opinions about a product but visit various web and social media sites [5].

Reading and synthesizing all reviews provided by others, however, is a challenging task, mainly due to the massive amount of reviews found on various review websites and social media [6]. Having a system that is capable of automated review analysis is therefore vital in today's online environment [7, 8]. When a system that allows automated review analysis is in place, potential buyers can obtain a holistic view of a product in a quick and accurate manner. Automated review analysis generally involves training machines to capture discriminative features from user reviews. The quality of this training process would determine the accuracy of a model to perform analysis on review texts or to classify them [9]. Feature extraction, which determines how features are selected, is of critical importance for the accuracy of automated review analysis [10].

In this study, we investigate the use of several text processing and machine learning classification algorithms for classifying rating polarity of review texts obtained from a popular review website, i.e., Yelp! [11]. We consider three targets: negative, neutral and positive. We use bag-of-words to represent the feature space. We carry out computational experiments with varying sizes of features and samples to reduce the dimension of the review data. We compare a range of machine learning classifiers to obtain the best results based on metrics such as accuracy, precision, recall and F-measure. These classifiers include single [12-18] and ensemble [19-23] models.

The rest of this paper is organized as follows. In Section II, we review work related to this research. In Section III, we describe the methods used, which include the system design, data and targets, as well as classifiers to be investigated. Experimental results and discussions are presented in Section IV. Finally, we conclude the paper in Section V and highlight future research directions.

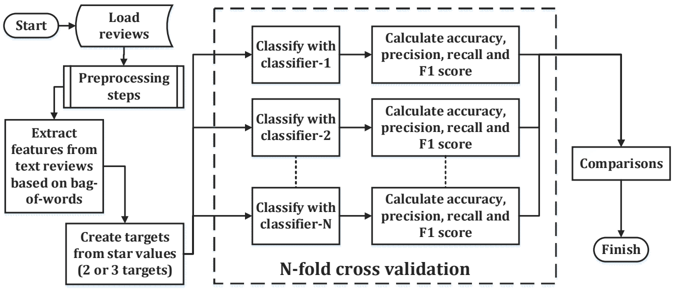
II. Related Work

A number of related studies on automatic rating assignment of product reviews exist in the relevant literature. Bagheri et al. [7] proposed an unsupervised model using heuristic rules, an iterative bootstrapping algorithm and aspect pruning to extract and detect different aspects of customer reviews. Tripathy et al. [24] grouped text polarity to negative and positive using an N-gram model with several classifiers, such as Naïve Bayes and Maximum Entropy, on IMDb movie review data. An algorithm using an intuitionistic fuzzy weighted averaging operator and preference ranking organization methods was developed by Liu et al. [25] to provide ratings (positive, negative, or neutral) to products based on their online reviews. Gui et al. [26] proposed a method to classify product reviews based on heterogeneous network representations, which include users (opinion holders), words, products (opinion targets) and polarities (positive and negative). They processed these network representations using different classifiers including deep learning, and found that Convolutional Neural Networks have the best results for the datasets they used, i.e., IMDb movie reviews, Yelp 2013, and Yelp 2014 [26]. Text mining of movie reviews and factors such as nationalities, ratings and other qualitative variables were used by Hur et al. [27] to do box-office forecasting. Three machine learning algorithms, i.e., the Classification and Regression Tree, Artificial Neural Network, and Support Vector Regression, were used as predictors.

Our work differs from others in that we consider many more machine learning models, including both single and ensemble classifiers. In terms of data, we use the latest 2017 Yelp review dataset, which is much larger than other datasets typically used in the literature.

III. Methods

To predict rating polarity from review texts, we applied a text processing algorithm and trained numerous classifier models. Fig. 1 details the proposed approach that we followed in our work.



Start
Load reviews
Preprocessing steps
Extract features from text reviews based on bag-of-words
Create targets from star values (2 or 3 targets)
Classify with classifier-1
Classify with classifier-2
Classify with classifier-N
Calculate accuracy, precision, recall and F1 score
Calculate accuracy, precision, recall and F1 score
Calculate accuracy, precision, recall and F1 score
N-fold cross validation
Comparisons
Finish

Fig. 1. The proposed approach for rating polarity prediction

A. Experimental Data & Labels

To validate our approach, we used consumer review data from the Yelp Dataset Challenge (Round 9) in 2017 [11] as our experimental dataset. Yelp is a leader in consumer ratings. It has grown rapidly since 2005. Yelp's users can review local businesses like restaurants, hair salons, bars, pubs, and many others. Users can write their own reviews and give star ratings of 1 to 5 to any businesses listed with Yelp [3]. The dataset used in this work contains 4.1 million review texts. With the massive amount of review texts, processing and experimenting them is a huge challenge. We utilized high-performance computing facilities from the University of Newcastle, Australia, which contain 2560 cores for 66 CPU and 4 GPU nodes, and up to 256 GB RAM can be assigned to each node.

To predict the polarity of a review text (input), we made use of the 1-5 stars rating given to each review as our (output) target label. We created three experimental output target types, by categorizing the reviews based on star ratings as follows: **Type A:** negative reviews are reviews with 1 & 2 star ratings while positive reviews are those with 3, 4 & 5 star ratings; **Type B:** negative reviews are reviews with 1, 2 & 3 star ratings while positive reviews are those with 4 & 5 star ratings; **Type C:** negative reviews are reviews with 1 & 2 star ratings, neutral reviews are those with 3 star ratings only, and positive reviews are those with 4 & 5 star ratings.

B. Preprocessing Steps

We pre-processed review texts prior to generating features for machine learning. The preprocessing steps involved removal of punctuations, numbers and English stop words, tokenization of words, and token lemmatization. Fig. 2 shows the preprocessing steps that we have applied to Yelp review texts. We used the natural language toolkit modules [28] to clean the review texts from punctuations and numbers as well as tokenize and lemmatize each word.

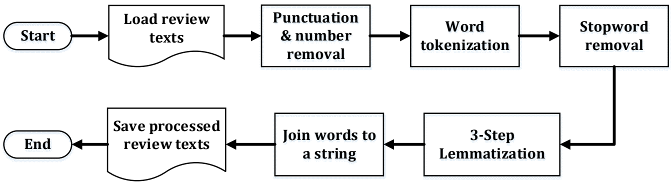


Fig. 2. Preprocessing steps

Our lemmatization approach has three steps: First lemmatizing each word as noun, then verb and after that adverb to reduce them to their basic form. Next the words are joined based on their original order and saved.

C. Feature Extraction

After the preprocessing steps, we created machine learning features from the processed review texts. We used Term Frequency (TF) to generate features for each preprocessed word token. The process to create features is as follows: First, a bag of words from all samples is created and their TF values are counted. Next, they are sorted based on their TF values. Features for each review text are extracted by checking the existence of each feature word. If a feature word does not exist in the review text then 0 is assigned, otherwise the feature word's TF value is assigned. A feature set can be created from all unique words found in the review texts or a subset of them above a certain threshold value.

Recall that the Yelp review dataset has more than 4.1 million records of reviews. The total number of unique words in this dataset after preprocessing, which could become features, is more than 240K. A problem arises when all unique words are used as features – this would create more than 984 billion values, which require a huge memory allocation for model training, even with the high-performance grid computers we have. To find proper sizes of features and samples, we performed a series of experiments with various settings to reduce the size of features and samples (see Section IV).

D. Classifiers

We considered not just standard single classifiers but also ensemble models in our work, and compared their performances against each other. In total, 13 single and ensemble classifiers commonly used for classification and text mining tasks were examined. In the following, we first describe the single classifiers, followed by the ensemble models. All classifiers in this research were built using the Scikit-Learn [29] module.

1) Single Models

Naïve Bayes is often used in text classification [30, 31]. Even though it is the simplest form of Bayesian Network, Naïve Bayes is still considered as one of the top 10 data mining algorithms [32]. Here, we investigated Multinomial and Bernoulli Naïve Bayes [15] and Gaussian Naïve Bayes [33].

The Nearest Neighbor, widely used [31] and improved [34], estimates an unknown sample based on the closest instance [12]. Besides the standard K-Nearest Neighbor, we also investigated the Nearest Centroid classifier [35].

The Decision Tree is a hierarchical tree model of decisions and outcomes [17]. It has been widely used in previous studies [30, 36], and therefore we also included it in our study.

The Generalized Linear Model was invented to overcome some limitations of linear regression models [37]. Variants of the model have been used to solve a range of classification problems [38-40]. We investigated four of them, namely the Logistic Regression [16], Ridge Regression [41], Passive Aggressive [14], and Stochastic Gradient Descent [40].

The Support Vector Machine (SVM) has excellent generalization performance and was successfully applied in many areas [24, 30, 42, 43]. Here, we investigated the SVM with Linear and Radial Basis Function (RBF) kernels [13].

The Multilayer Perceptron is a feedforward Artificial Neural Network usually used as a supervised model for classification [27, 30, 44-46]. It works via minimizing errors of its results by computing all the weights in its networks. It has two steps, namely propagation and weight update [18].

2) *Ensemble Models*

Bagging Predictors are done by generating versions of the single predictor and using it to get a cluster of predictors. These predictors are trained on training sets created via bootstrapping [20]. Bagging has been used in many areas [4, 30].

Random Forests are an ensemble of Decision Tree predictors, in which each tree in the 'forest' is trained using a random vector that is sampled independently [19]. Random Forests have been used in many areas including text classification [4]. Besides the standard Random Forests, we also investigated Randomized Decision Trees [22].

Ada Boost stands for Adaptive Boosting [23]. It iteratively combines multiple weak base classifiers to get a better classifier. It was successfully used in past studies (e.g., see [30]).

Gradient Boosting is an ensemble of gradient boosted regression trees for classification of dirty data [21]. This algorithm has been used for classification problems [30].

E. *Evaluation Methods*

For evaluation purposes, four metrics including accuracy, precision, recall, and F-measure were used in our study. They are defined as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F_{measure} = \frac{2*Precision*Recall}{Precision+Rec} \quad (4)$$

Here, positive reviews are the positive class. TP is True Positive, which refers to the number of reviews that are correctly classified as the positive class; TN is the number of correctly classified negative reviews (and neutral); FP is the number of classified positive reviews that are actually not positive; and FN is the number of wrongly classified reviews that are actually positive reviews.

IV. Experimental Results And Discussions

A. *Experiments on Single Classifiers*

We investigated the performances of single classifiers in identifying the polarity of reviews from the 2017 Yelp dataset. A total of 13 classifiers, as seen in Table I, were tested using three experiment types discussed earlier.

TABLE I. SINGLE CLASSIFIERS USED IN EXPERIMENTS

No.	Classifier Name	Parameter
1	Multinomial Naïve Bayes (MNB)	alpha = 1.0
2	Bernoulli Naïve Bayes (BNB)	alpha = 1.0
3	Gaussian Naïve Bayes (GNB)	-
4	K-Nearest Neighbor (KNN)	K = 5, Euclidean
5	Nearest Centroid (NC)	Euclidean
6	Decision Trees (DT)	Gini index
7	Logistic Regression (LR)	max iterations: 100
8	Ridge Regression (RR)	alpha = 1.0
9	Passive Aggressive (PA)	Epochs = 5, PA-I formula
10	Stochastic Gradient Descent (SGD)	estim: Linear SVM, learning rate = 1.0 / (alpha * (t + t0))
11	RBF-kernel SVM (RSVM)	gamma = 1/n features
12	Linear-kernel SVM (LSVM)	max iteration = 1000
13	Multilayer Perceptron (MLP)	1 hidden layer - 100 neurons, rectified linear unit, α = 0.001

We first performed 10-fold cross validation based on 10,000 randomly selected review texts using the 13 classifiers. In these experiments, we ran the classifiers with varying numbers of features from 250 features to the maximum (i.e., 245,071). These features were selected based on their TF values. The accuracies of the classifiers with different feature sets on the three experiment types are shown in Fig. 3. From the figure, we see that classifiers like the BNB and GNB, DT, KNN, NC and RSVM are not performing well compared to other classifiers. MNB performs well with a small amount of features, but the accuracy deteriorates as the number of features increases. The results also show that the accuracy of all classifiers, except for RR, does not increase any further when the number of features is beyond 5,000. RR reaches its peak accuracy at around 10,000 features. These experiments clearly showed that increasing the number of features, which increases training complexity, does not necessarily increase the accuracy of training models.

Next, we performed 10-fold cross validation using 500 features sorted by the TF values, on various review texts ranging from 10,000 records to the maximum of 4,133,088 records (i.e., the entire Yelp dataset). Fig. 4 shows the accuracies of these classifiers with an increasing number of records on the three experiment types. From the figure, we see that the accuracies of these classifiers increase quite substantially as the number of records increases, until about 500K records. After this point, the accuracy increase is marginal.

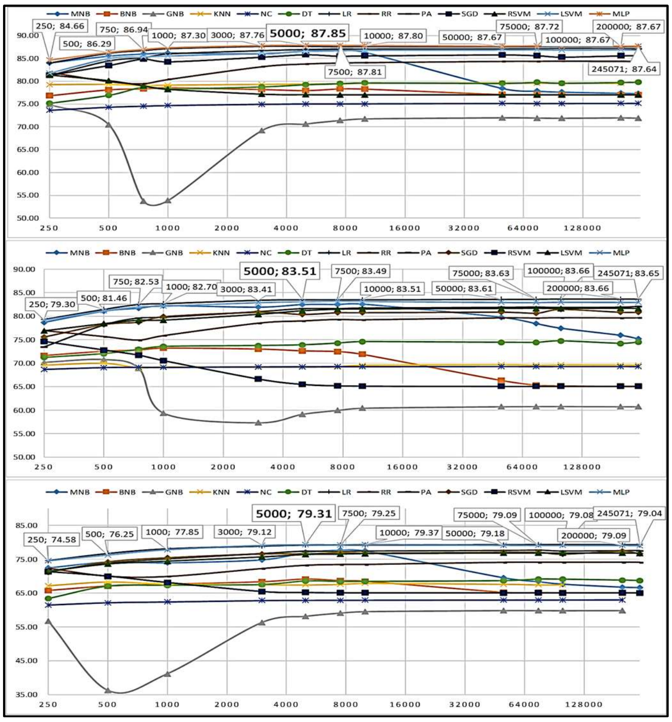


Fig. 3. Accuracy (y-axis) vs. the number of features (x-axis) for single classifiers; Experiment types A (top), B (middle), and C (bottom)

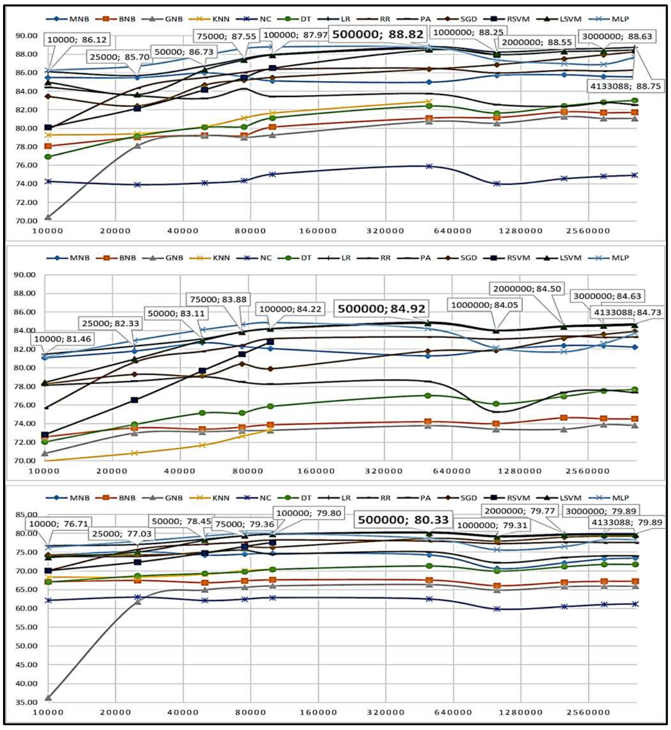


Fig. 4. Accuracy (y-axis) vs. the number of training records (x-axis) for single classifiers; Experiment types A (top), B (middle), and C (bottom)

From Fig. 3 and Fig. 4, we can see that the MLP, LR and LSVM are the three best classifiers in most cases. Additionally, we observe that experiment type A has the highest accuracy, followed by type B and type C. This implies that there are more positive reviews than negative ones.

To test the robustness of the three best classifiers, we performed further experiments with varying features (i.e., 1,000 to 10K feature sets) and training samples (i.e., 100K to 1M training samples) based on 10-fold cross validation. The results for each of the classifiers are reported in Table II. From the table, we see that the MLP has better results on type A and LR has better results for type B across different performance metrics. As for type C, the best accuracy and recall are acquired by LR while the MLP has the best precision and F-measure score.

TABLE II. RESULTS FOR THE 3 BEST SINGLE CLASSIFIERS

Classifier Name	Experi-ment Type	Max. Acc. (%)	Average (%)			
			Acc	Prec	Rec	F1
LSVM	A	90.57	89.26	89.13	89.26	89.14
	B	86.94	85.40	85.34	85.40	85.32
	C	82.25	80.83	79.09	80.83	79.37
LR	A	90.90	90.17	89.94	90.17	90.00
	B	87.23	86.46	86.34	86.46	86.37
	C	82.82	82.04	79.92	82.04	80.41
MLP	A	91.23	90.38	90.33	90.38	90.35
	B	87.54	86.12	86.13	86.12	86.12
	C	82.65	81.47	80.82	81.47	81.09

B. *Experiments on Ensemble Classifiers*

Besides the single classifiers, we also investigated the performances of 5 ensemble classifiers as seen in Table III. The experiments were conducted in the same manner as that of the three best classifiers with results presented in Table II.

By default, both Bagging and Ada Boost would use the Decision Tree as their base classifier [19, 21]. However, it is possible to change the base classifier to other classifiers. In this study, we further investigated the performance of Bagging and Ada Boost with the aforementioned three best single classifiers as base classifiers. Unfortunately, Ada Boost can only combine with classifiers that support sample weighting, thus the MLP cannot be combined with it. By comparing the results of Table II and the first 15 rows of Table IV, we observe that the three best single classifiers in Table II are better than all these five ensemble classifiers on all metrics. As can be seen in Section IV.A, the performance of Decision Tree, which is the default base classifier of these five ensemble models, is not as good compared to those classifiers listed in Table II. This could explain why the ensemble models' results are worse than that in Table II. Looking at the second part of Table IV on results of Bagging and Ada Boost with the three best single classifiers as base classifiers, we note that the performance of Bagging is improved but not Ada Boost's.

TABLE III. ENSEMBLE CLASSIFIERS USED IN EXPERIMENTS

No.	Classifier Name	Parameter
1	Randomized Decision Trees	10 estimators (DT), Gini index
2	Random Forest	10 estimators (DT), Gini index
3	Gradient Boosting	loss function: LR, 100 estimators (LR), mean squared error
4	Bagging Predictors	10 estimators (DT), bootstrap: true
5	Ada Boost	50 estimators (DT)

TABLE IV. RESULTS FOR THE ENSEMBLE CLASSIFIERS

Classifier Name	Experi-ment Type	Max. Acc. (%)	Average (%)			
			Acc	Prec	Rec	F1
Randomized Decision Trees	A	80.43	79.64	80.13	79.64	79.86
	B	74.57	73.74	74.08	73.74	73.89
	C	69.19	68.04	68.20	68.04	68.09
Gradient Boosting	A	86.98	86.64	86.42	86.64	85.15
	B	82.57	82.28	82.46	82.28	81.39
	C	78.52	78.23	75.85	78.23	74.55
Random Forest	A	87.95	87.19	86.73	87.19	86.86
	B	83.04	82.20	82.36	82.20	82.26
	C	79.27	78.52	76.48	78.52	76.19
Bagging (DT)	A	87.15	86.38	86.38	86.38	86.37
	B	82.16	81.35	81.71	81.35	81.48
	C	78.29	77.45	75.62	77.45	76.05
Ada Boost (DT)	A	87.26	86.75	86.07	86.75	85.99
	B	82.79	82.31	82.02	82.31	81.94
	C	77.88	77.59	74.30	77.59	74.55
Bagging (LSVM)	A	90.58	89.26	89.14	89.26	89.14
	B	86.96	85.40	85.34	85.40	85.33
	C	82.26	80.83	79.10	80.83	79.37
Ada Boost (LSVM)	A	90.57	89.26	89.14	89.26	89.14
	B	86.95	85.40	85.34	85.40	85.33
	C	82.25	80.83	79.10	80.83	79.36
Bagging (LR)	A	91.16	90.36	90.12	90.36	90.18
	B	87.54	86.65	86.53	86.65	86.55
	C	83.19	82.27	80.11	82.27	80.60
Ada Boost (LR)	A	89.51	88.78	88.80	88.78	88.78
	B	85.69	84.78	84.81	84.78	84.79
	C	80.56	79.44	78.49	79.44	78.89
Bagging (MLP)	A	92.11	91.21	91.08	91.21	91.13
	B	88.81	87.47	87.42	87.47	87.44
	C	84.27	83.39	82.00	83.39	82.45

C. *Experiment Types vs. Star Ratings*

Finally, we investigated each experiment type (A, B, and C) in more detail by conducting experiments using Bagging with the three best single classifiers as base classifiers. Table V shows the comparison between accuracies obtained based on experiment types and star ratings.

From Table V, we see that neutral ratings (3 stars) for experiment type C have the lowest accuracy. This indicates that the trained classifiers are not quite capable of recognizing neutral review ratings (3 stars). Predicting neutral ratings is always going to be challenging because neutral ratings may not have the equal composition of positive and negative comments. Most of the time, neutral rating comments tend to skew towards one of the target classes (positive or negative). This can be seen from the results of experiment types A and B, where we assumed the 3-star rating as either negative or positive. An interesting observation is that the 3-star rating has higher accuracy when it is classified as positive rating (type A). It can be inferred that users giving a 3-star rating tend to provide more positive comments than negative ones.

V. Conclusion And Future Work

In this paper, we presented a study to automatically determine the polarity of review texts as positive, neutral, or negative. Given that ratings and reviews are becoming pivotal in helping potential customers to make a purchase decision, the need for a study like this is clear. We used the 2017 Yelp review dataset and adopted the TF method to create features from this dataset. We found that having more features and data added into the training set does not necessarily enhance model performance. In fact, after exceeding a specific threshold, the model performance becomes stagnant. Our experiments showed that 5,000 features and 500,000 reviews are the cut-off points for polarity prediction in this context.

The computational results identified three single classifiers, i.e., LR, LSVM and MLP, as best-performing ones, obtaining accuracy, precision, recall and F-measure scores around 90% or more for experiment type A, above 85% for experiment type B, and above 80% for experiment type C. Further improvements can be obtained by using these three classifiers as base classifiers to build ensemble models. We found that classifying neutral ratings (3 stars) is more challenging due to the fact that neutral reviews tend not to have the equal composition of positive and negative comments. We also noticed that users who gave 3-star ratings have the tendency of giving more positive reviews.

In our future work, we intend to explore more advanced feature selection techniques to improve the performance of machine learning models considered in this study, especially in classifying neutral reviews. We also plan to use sentiment analysis methods to determine the polarity of customer reviews.

Acknowledgment

The first author would like to acknowledge financial support from the Indonesian Endowment Fund for Education (LPDP), Ministry of Finance; and the Directorate General of Higher Education (DIKTI), Ministry of Research, Technology and Higher Education, Republic of Indonesia; as well as travel funds from the Postgraduate Research Support Scheme provided by the Faculty of Engineering and Built Environment at the University of Newcastle, Australia.

TABLE V. ACCURACY COMPARISON BETWEEN TARGET TYPES

Classifier	Experiment Type	Average Accuracy (%)	Avg. accuracy of targets (%)			Avg. accuracy of stars (%)				
			Negative	Neutral	Positive	1	2	3	4	5
Bagging (LSVM)	A	89.26	72.45	-	93.96	81.78	57.78	79.84	95.31	97.26
	B	85.40	76.56	-	89.94	89.21	82.44	58.51	82.63	94.23
	C	80.83	76.65	25.90	92.30	84.81	63.79	25.90	86.60	95.64
Bagging (LR)	A	90.36	73.40	-	95.09	83.39	57.69	81.12	96.65	98.24
	B	86.65	77.52	-	91.33	90.84	83.85	58.37	84.02	95.63
	C	82.27	78.16	25.35	94.05	86.86	64.48	25.35	88.76	97.15
Bagging (MLP)	A	91.21	77.32	-	95.09	86.96	62.14	81.79	96.59	98.09
	B	87.47	80.43	-	91.08	92.89	86.05	62.72	84.01	95.24
	C	83.39	81.02	34.67	93.08	89.51	67.65	34.67	87.37	96.43

REFERENCES

[1] A. Y. K. Chua and S. Banerjee, "Helpfulness of user-generated reviews as a function of review sentiment, product type and information quality," *Computers in Human Behavior,* vol. 54, pp. 547-554, 2016.

[2] I. Pranata, G. Skinner, and R. Athauda, "A survey on the usability and effectiveness of web-based trust rating systems," in *Proceedings of the IEEE/ACIS 12th International Conference on Computer and Information Science (ICIS)*, Niigata, Japan, 2013, pp. 455-460.

[3] I. Pranata and W. Susilo, "Are the most popular users always trustworthy? The case of Yelp," *Electronic Commerce Research and Applications,* vol. 20, pp. 30-41, 2016.

[4] S. L. Lo, R. Chiong, and D. Cornforth, "Ranking of high-value social audiences on Twitter," *Decision Support Systems,* vol. 85, pp. 34-38, 2016.

[5] B. Liu, *Sentiment Analysis and Opinion Mining*. Morgan & Claypool, 2012.

[6] M. Salehan and D. J. Kim, "Predicting the performance of online consumer reviews: A sentiment mining approach to big data analytics," *Decision Support Systems,* vol. 81, pp. 30-40, 2016.

[7] A. Bagheri, M. Saraee, and F. de Jong, "Care more about customers: Unsupervised domain-independent aspect detection for sentiment analysis of customer reviews," *Knowledge-Based Systems,* vol. 52, pp. 201-213, 2013.

[8] E. Fersini, E. Messina, and F. A. Pozzi, "Expressive signals in social media languages to improve polarity detection," *Information Processing & Management,* vol. 52, no. 1, pp. 20-35, 2016.

[9] M. D. Devika, C. Sunitha, and A. Ganesh, "Sentiment analysis: A comparative study on different approaches," *Procedia Computer Science,* vol. 87, pp. 44-49, 2016.

[10] C. Bhadane, H. Dalal, and H. Doshi, "Sentiment analysis: Measuring opinions," *Procedia Computer Science,* vol. 45, pp. 808-814, 2015.

[11] Yelp. (2017, February 5). *Yelp dataset challenge: Round 9 of the Yelp dataset challenge: Our largest yet!* Available: https://www.yelp.com.au/dataset_challenge

[12] M. Bramer, "Nearest neighbour classification," in *Principles of Data Mining.* London: Springer-Verlag, 2007, pp. 31-38.

[13] C. Campbell and Y. Ying, *Learning with Support Vector Machines*. Morgan & Claypool, 2011.

[14] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *Journal of Machine Learning Research,* vol. 7, pp. 551-585, 2006.

[15] C. D. Manning, P. Raghavan, and H. Schuetze, "Naïve Bayes text classification," in *Introduction to Information Retrieval*: Cambridge University Press, 2008, pp. 234-265.

[16] S. Menard, *Logistic Regression: From Introductory to Advanced Concepts and Applications*. Los Angeles: SAGE, 2010.

[17] L. Rokach and O. Maimon, *Data Mining with Decision Trees: Theory and Applications*. World Scientific Publishing Company, 2007.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1: MIT Press, 1986, pp. 318-362.

[19] L. Breiman, "Random forests," *Machine Learning,* vol. 45, no. 1, pp. 5-32, 2001.

[20] L. Breiman, "Bagging predictors," *Machine Learning,* vol. 24, no. 2, pp. 123-140, 1996.

[21] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics,* vol. 29, no. 5, pp. 1189-1232, 2001.

[22] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning,* vol. 63, no. 1, pp. 3-42, 2006.

[23] J. Zhu, H. Zou, S. Rosset, and T. Hastie, "Multi-class AdaBoost," *Statistics and Its Interface,* vol. 2, pp. 349-360, 2009.

[24] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Systems with Applications,* vol. 57, pp. 117-126, 2016.

[25] Y. Liu, J.-W. Bi, and Z.-P. Fan, "Ranking products through online reviews: A method based on sentiment analysis technique and intuitionistic fuzzy set theory," *Information Fusion,* vol. 36, pp. 149-161, 2017.

[26] L. Gui, Y. Zhou, R. Xu, Y. He, and Q. Lu, "Learning representations from heterogeneous network for sentiment classification of product reviews," *Knowledge-Based Systems,* vol. 124, pp. 34-45, 2017.

[27] M. Hur, P. Kang, and S. Cho, "Box-office forecasting based on sentiments of movie reviews and independent subspace method," *Information Sciences,* vol. 372, pp. 608-624, 2016.

[28] NLTK. (2017, January 25). *Nltk Package*. Available: http://www.nltk.org/api/nltk.html

[29] Scikit-learn. (2017, April 3). *API Reference*. Available: http://scikit-learn.org/stable/modules/classes.html

[30] Z. Hu, R. Chiong, I. Pranata, W. Susilo, and Y. Bao, "Identifying malicious web domains using machine learning techniques with online credibility and performance data," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 5186-5194.

[31] Z. Lungan, J. Liangxiao, L. Chaoqun, and K. Ganggang, "Two feature weighting approaches for Naïve Bayes text classifiers," *Knowledge-Based Systems,* vol. 100, pp. 137-144, 2016.

[32] J. Liangxiao, L. Chaoqun, W. Shasha, and Z. Lungan, "Deep feature weighting for Naïve Bayes and its application to text classification," *Engineering Applications of Artificial Intelligence,* vol. 52, pp. 26-39, 2016.

[33] T. F. Chan, G. H. Golub, and R. J. LeVeque, *Updating Formulae and a Pairwise Algorithm for Computing Sample Variances*. New Haven: Stanford University, 1979.

[34] P. Zhibin, W. Yidi, and K. Weiping, "A new general nearest neighbor classification based on the mutual neighborhood information," *Knowledge-Based Systems,* vol. 121, pp. 142-152, 2017.

[35] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, "Diagnosis of multiple cancer types by shrunken centroids of gene expression," *Proceedings of the National Academy of Sciences of the United States of America,* vol. 99, no. 10, pp. 6567-6572, 2002.

[36] Z. Xu, P. Li, and Y. Wang, "Text classifier based on an improved SVM decision tree," *Physics Procedia,* vol. 33, pp. 1986-1991, 2012.

[37] G. H. Dunteman and M.-H. R. Ho, *An Introduction to Generalized Linear Models*. SAGE Publications, Inc., 2011, pp. 2-6.

[38] D. D. A. Bui, G. D. Fiol, and S. Jonnalagadda, "PDF text classification to leverage information extraction from publication reports," *Journal of Biomedical Informatics,* vol. 61, pp. 141-148, 2016.

[39] L. Jing, Z. Peilin, and H. Steven C. H., "Online passive-aggressive active learning," *Machine Learning,* vol. 103, no. 2, pp. 141-183, 2016.

[40] L. Ruhwinaningsih and T. Djatna, "A sentiment knowledge discovery model in Twitter's TV content using stochastic gradient descent algorithm," *TELKOMNIKA,* vol. 14, no. 3, pp. 1067-1076, 2016.

[41] K. P. Murphy, *Machine Learning*. MIT Press, 2012.

[42] S. L. Lo, R. Chiong, and D. Cornforth, "Using support vector machine ensembles for target audience classification on Twitter," *PLoS ONE,* vol. 10, no. 4, e0122855, 2015.

[43] S. L. Lo, D. Cornforth, and R. Chiong, "Identifying the high-value social audience from Twitter through text-mining methods," in *Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES 2014)*, Singapore, 2014, pp. 325-339.

[44] R. Adipranata, G. S. Budhi, and B. Setiahadi, "Automatic classification of sunspot groups for space weather analysis," *International Journal of Multimedia and Ubiquitous Engineering,* vol. 8, no. 3, pp. 41-54, 2013.

[45] G. S. Budhi and R. Adipranata, "Java characters recognition using evolutionary neural network and combination of Chi2 and backpropagation neural network," *International Journal of Applied Engineering Research,* vol. 9, no. 22, pp. 18025-18036, 2014.

[46] G. S. Budhi and R. Adipranata, "Handwritten Javanese character recognition using several artificial neural network methods," *Journal of ICT Research and Applications,* vol. 8, no. 3, pp. 195-212, 2015.