ISSN 1870-4069

Customer Feedback Analysis Using Machine Learning Techniques

Anupam Jamatia, Kanishka Joshi, Kundan Kumar,
Shivam Kumar, Subrito Haldar

Department of Computer Science & Engineering,
National Institute of Technology Agartala,
India

{anupamjamatia, kanishkj34, kundanjnv11,
shivam.nita14, subrito996}@gmail.com

Abstract. Customer feedback are the representation of customers' opinions about the concerned products in today's business organization. Thus its analysis is essential in providing a company insights into what it has to do to render better customer experience in the future. Our work focuses on the automatic processing of customer feedback using machine learning approaches and subsequently analyzing them, which is otherwise an impossible task to do manually due to customer experience on sheer volume and variety of services, products. We compare the performance of different machine learning classifiers such as Naïve Bayes, Support Vector Machine, k-Nearest Neighbors and Random Forest on the collected corpus. The maximum accuracy is obtained using Random Forest classifier with an accuracy of 69.74%.

Keywords: Customer feedback, machine learning.

1 Introduction

The purpose of the customer feedback is that it provides marketers and business owners with insight that they can use to improve their business, products and overall customer experience. Classification of feedback is essential to gain the better perspective on the views of the customers. Customer feedback analysis measures how happy customers are with a company's products and services. With the ever-increasing size of feedback data, it has become an improbable task to manually inspect each review.

So it is necessary to automate the overall process to provide businesses with a better view of what it has to change, what it has to improve on, and what it has to do, to retain and grow revenue and profit. Customer feedback analysis has become an industry on its own. Hence many companies understandably want to automate customer feedback analysis system but a major hurdle is to deal with multilingual environment which exists in all over the world. There are several online survey-based companies who acquire customer data from their clients and do the analysis. Firstly, some commonly used categorizations include five-class viz. 'Excellent'- 'Good'- 'Average'- 'Fair'- 'Poor' by Yin *et. al* [20] and SurveyMonkey[1].

[1] https://www.surveymonkey.com/

Secondly, there are opinion and responsiveness based five-class viz. 'Positive'- 'Neutral'- 'Negative'- 'Answered'- 'Unanswered' by Freshdesk[2]. Lastly, there are seven-class 'Refund'- 'Complaint'- 'Pricing'- 'Tech Support'- 'Store Locator' - 'Feedback' - 'Warranty' by Sift[3]. These surveys are a vital tool for a variety of research fields, including marketing, social and official statistics research. A lot of work has been done in the field of sentiment analysis of feedback that classifies the sentiment polarity of the customer feedback into positive, negative, neutral and so on.

But interpretation of those reviews into bug, complaint, comment or request for better customer support is essential as well. Our work deals with classifying a customer review (in English) into one or more of the six predefined classes taken from Liu *et. al.* [13]. The classes are 'comment', 'request', 'bug', 'complaint', 'meaningless' and 'undetermined'. This paper can be viewed as multiclass classification [12, 21, 17, 4, 14] problem. We have used TF-IDF feature vectors and then used supervised machine learning techniques to train our dataset and subsequently test it.

The rest of the paper is organized as follows; in Section 2 we discussed the related research work on the customer feedback analysis. Data collection and preprocessing are described in Section 3. In Section 4, the description of the features used are given and the performance of four different machine learning methods are presented. Results, observations and error analysis are discussed in Section 5 and Section 6 respectively. Section 7 sums up with future research scope. The source code of our system can be found here.[4]

2 Related Work

There has been some significant work done in the area of customer feedback analysis, sentiment analysis of feedback and multiclass classification of feedback. For example, the work by Bentley and Batra [1] on Microsoft Office feedback describes how an engineer or a manager finds the signal in feedback to make business decisions by using classification, on-demand clustering and other machine learning techniques. The problem of sentiment polarity categorization has been tackled in Fang and Zhan [5].

In their experiment, random forest model performs the best on manually-labeled and machine-labeled sentences in case of sentence-level categorization but Support Vector Machines (SVM) model and Naïve Bayesian model perform better than Random Forest model in case of review-level categorization. Large feature vectors in combination with feature reduction can train linear support vector machine can achieve high classification accuracy, which was described by Gamon [6] in his paper on sentiment classification on customer feedback.

The paper suggests that the addition of deep linguistic analysis features to a set of surface level word n-gram features contributes consistently to classification accuracy in this domain. Mukherjee and Bhattacharyya [16] in their paper regarding feature specific sentiment analysis for product reviews have used dependency parsing method to identify relations among the opinion expressions.

[2] https://freshdesk.com/

[3] https://www.startupranking.com/sift

[4] https://drive.google.com/drive/folders/1d0w0yRbubqHC4R7ev7KilU3gybsDscDQ

Other such related research includes the paper by Chakankar *et al.* [3] which constitutes sentiment analysis of users' reviews and comments. They have used three different datasets and have classified the reviews/ comments as being positive or negative.

- The first dataset has movie reviews from IMDB [5]. They have used 25000 highly polar reviews for training purpose and 25000 reviews for testing purpose. For this dataset SVM model has obtained the best accuracy of 88.89%.
- The second dataset has 2000 processed movie reviews drawn from IMDB archive. For this dataset also, SVM model has outperformed other classifiers.
- The third dataset consisted of social commentary having insults; out of 3947 instances of social commentary 1049 are insults. For this dataset, Naïve Bayesian model has outperformed SVM model and Logistic Regression model.

A set of techniques has been proposed by Hu and Liu [8] to mine and summarize reviews based on data mining and natural language processing methods which is useful to common shoppers as well as product manufactures. They have performed the task in three steps.

- Mining product features that has been commented by the customers.
- Identifying opinion sentences in each reviews and deciding whether each opinion sentence is positive or negative.
- Summarizing the results by aggregating the results from the previous steps.

Two-class sentiment classification of movie reviews as positive or negative using machine learning techniques has been done by Pang *et al.* [18]. They have used Naïve Bayes, maximum entropy classification and SVM. They have taken n-grams as feature and SVM gives the best accuracy of 82.9% on unigrams feature.

3 Dataset

We received the dataset from the IJCNLP 2017 organizers of shared tasks for customer feedback analysis[6]. Each document in the dataset was pre-annotated into one of the classes, with a few documents (4.5%) being classified into more than one class. In the corpus, there are a total number of 3723 documents, which are distributed into six predefined classes, namely comment, request, bug, complaint, meaningless and undetermined. A few samples have been listed in table 1.

From them, 'comment' and 'complaint' classes have the maximum number of feedback. The class 'undefined' has the least number of feedback. About 4.5% of the feedback were annotated with multiple labels. The entire distribution of dataset into classes has been displayed in table 2.

[5] http://www.imdb.com

[6] https://sites.google.com/view/customer-feedback-analysis/

Table 1. Samples of feedback sentences from the dataset.

Statement	Qualifier
It is so wonderful to use.	Comment
Being a new Apple Developer, I needed a fast program that would work fast and has an easy User Interface.	Request
Phone froze as if the app had a virus.	Bug
Beautiful afternoon at the Bristol!	Meaningless
Even the accessories in the app look fake.	Complaint
Maybe old style clothing too from civil war era not just city slicker clothing.	Undetermined
It's nothing but it consumes a large amount a CPU and memory.	Complaint, Bug

Table 2. Class distribution in corpus.

Class	Number of feedback	Number of tokens
Bug	92	1553
Comment	2034	22099
Complaint	1096	15720
Meaningless	354	3600
Request	122	1827
Undefined	25	336
Total	3723	45135

The data provided by IJCNLP Shared Task 2017 organizer was raw in nature. That is, extra data (meta data) was present along with it. The raw data was in the format of: Raw Data = Text ID + Sentence + Classifier. Hence, pre-processing of raw data was necessary. First, the 'Text ID' was removed. Afterwards, stop-words, that is, common words which would appear to be of little value in helping select documents matching a user need, are excluded.

Further, words with frequency of exactly one were also removed, as they do not contribute to the overall classification process as well. Later the data was tokenized. We have used NLTK[7] sentence tokenizer for tokenizing sentences and then used NLTK word tokenizer for tokenizing words. After the above process, we have got refined data.

4 Experiments

We have used some of the popular supervised machine learning algorithms in our approach. We have used TF-IDF as features of the corpus to convert the textual representation of information into vector space model. Thereafter we divided the vector space into training and testing data using k-fold algorithm (k=10). Subsequently we implemented six classifiers, namely Gaussian Naïve Bayes classifier, Multinomial Naïve Bayes classifier, Bernoulli Naïve Bayes classifier, SVM, k-Nearest Neighbours (k-NN) classifier and Random Forest classifier. We then calculated the precision and accuracy score for each and compared them.

[7] http://www.nltk.org/api/nltk.tokenize.html

We analyzed the results with the help of confusion matrix. After pre-processing of the data, we carried out feature selection and performed an analysis using TF-IDF.

$$\text{TF(word)} = F_{\text{count}}(\text{word})/N, \tag{1}$$

$$\text{IDF(word)} = log_e(N/E_{\text{count}}(\text{word})), \tag{2}$$

$$\text{TF} - \text{IDF} = \text{TF} \times \text{IDF}. \tag{3}$$

At first, we calculated the Bag-of-Words vector and using the same, we calculated the term frequency (TF) and later inverse document frequency (IDF) values for each unique word in each of the documents. Following that a 2-dimensional vector space was created.After the production of feature vectors, we then created the training and testing set using k-fold cross validation [11] algorithm, setting k=10, i.e. 90% of the dataset was kept in the training set and the remaining 10% in the test set.

After the division of vectors we implemented six supervised classifiers and analyzed the results. Naïve Bayes(NB) classifiers [22] are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. We have applied three types of Naïve Bayes [15] classifiers on the data. They are mentioned below. Gaussian NB supports continuous valued features and models each as conforming to a Gaussian (normal) distribution.

$$p(x = v|C_k) = \frac{1}{\sqrt{2\prod \sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}. \tag{4}$$

where μ_k is the mean of the values in x associated with class C_k, σ_k^2 be the variance of the values in x associated with class C_k and $p(x = v|C_k)$ is the probability distribution of v given a class C_k. Multinomial NB [10] estimates the conditional probability of a particular word given a class as the relative frequency of term t in documents belonging to a class. The variation takes into account the number of occurrences of term t in training documents from that class including multiple occurrences.

$$p(x|C_k) = \frac{\left(\sum_i x_i\right)!}{\prod_i x_i!} \prod_i p_{k_i}^{x_i}. \tag{5}$$

where, x is the feature vector, p_{ki} is the probability of class C_k generating the term x_i. Bernoulli NB generates boolean value/indicator about each term of the vocabulary equal to 1 if the term belongs to examining document, if not it marks 0. Non-occurring terms in document are takes into document and they are factored when computing the conditional probabilities and thus the absence of terms is taken into account.

$$p(x|C_k) = \prod_{i=1}^{n} p_{k_i}^{x_i}(1 - p_{ki})^{1-x_i}. \tag{6}$$

where p_{ki} is the probability of class C_k generating the term x_i. k-NN [9] algorithm is a non-parametric method used for classification. The input consists of the k closest training examples in the feature space.

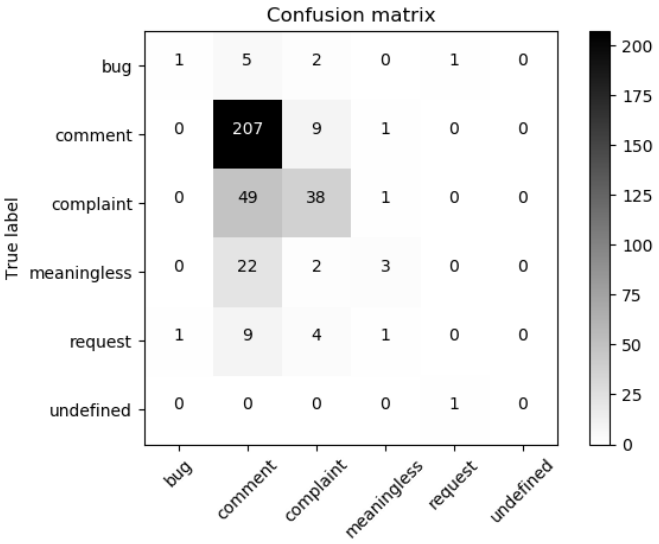

Fig. 1. Random Forest Confusion Matrix

The test sample is classified into a particular class as an output, depending upon the majority of the classes of its k-nearest neighbours. In plain words, if you are similar to your neighbours, then you are one of them. After sufficient experimenting, the value of k equals to 4 was taken. Binary SVM can be converted into a multiclass classifier using standard one versus one and one versus all.

We have used one versus all technique, in which a k-class problem is viewed as k many 2-class problem. In the training process, k binary classifiers are trained and each classifier tries to separate itself from $(k-1)$ classes.Random forests [2] operate by constructing a number of decision trees at training time and outputting the class that is the mode of the classes. After some trial-and-error and close examination, the maximum depth as 200 and random state as 2 was taken to employ this classifier.

5 Result and Observations

After the experiments, an analysis of the six classifiers was done for the baseline by calculating some parameters, namely accuracy, precision score, recall score and F1 score with the help of confusion matrix. A confusion matrix [19], also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one.

Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class (or vice versa). We constructed a confusion matrix for result analysis of our best performing Random Forest model. In our case, we put the instances of predicted classes in columns and instances of the actual classes in rows.

Hence a particular element of the matrix, say $CM[i][j]$ represents the number of feedback which were of class i but predicted as j. So when $i = j$, that is, the diagonal elements refer to the number of correctly predicted documents. The confusion matrix that we obtained is shown below. From the confusion matrix displayed in Figure 1, we infer that the maximum number of errors were found in differentiating between 'comment' and 'complaint' classes.

This was followed by the errors found in differentiating between 'meaningless' and 'comment'. Most correct predictions were from 'comment' class. The degree to which the result of a measurement, calculation or specification conforms to the correct value or a standard is called accuracy. It is the ratio of total number of correctly predicted documents to total number of documents.

$$\text{Accuracy} = \frac{\sum_{i=1}^{6} \text{CM}[i][i]}{\sum_{i=1}^{6}\sum_{j=1}^{6} \text{CM}[i][j]} \times 100. \tag{7}$$

where CM = Confusion Matrix. Precision value for a class is the ratio of related information out of retrieved information to total retrieved information. Here we have taken average precision value of all classes.

$$\text{Precision} = \frac{1}{6}\sum_{i=1}^{6} \frac{\text{CM}[i][i]}{\sum_{j=1}^{6} \text{CM}[i][j]}. \tag{8}$$

Recall value for a class is the ratio of related information out of retrieved information to total related information. Here we have taken average recall value of all classes.

$$\text{Recall} = \frac{1}{6}\sum_{i=1}^{6} \frac{\text{CM}[i][i]}{\sum_{j=1}^{6} \text{CM}[j][i]}. \tag{9}$$

The F_1 score can be interpreted as a weighted harmonic mean of the precision and recall,where an F-beta score reaches its best value at 1 and worst score at 0.

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{10}$$

6 Error Analysis

After conducting the aforesaid experiments, we found a few reasons for the occurrence of errors:

– Most of the documents were classified into a single class, but some of them (about 4.5 percent) were classified into more than one class, e.g "Its nothing but it consumes a large amount of CPU and memory" was assigned both 'bug' and 'comment' classes. This creates an ambiguity for the classifier during training.

Table 3. Observations.

Classifier	Accuracy	Precision	Recall	F1 Score
Gaussian NB	53.42	0.53	0.53	0.53
Multinomial NB	55.59	0.56	0.56	0.56
Bernoulli NB	55.09	0.55	0.55	0.55
SVM	58.49	0.59	0.59	0.59
k-NN	57.65	0.58	0.58	0.58
Random forest	69.74	0.68	0.68	0.68

- The dataset was highly imbalanced; 'bug' and 'undefined' classes have 92 and 25 feedback respectively. On the other hand, 'comment' and 'complaint' classes have 1096 and 2034 feedback respectively.
- Due to the uneven distribution of data, a couple of classes have very few documents. This affects the dataset division process (into train and test set) as those few documents might end up at either train set or test set. This results in ramifications.

7 Conclusion and Future Scope

Working on multiclass classification that too for six classification of unbalanced dataset is not an easy task in the field of natural language processing & machine learning. After preprocessing of corpus, we employed 10-fold cross-validation method for training and testing purpose. We employed various machine learning algorithms to get the best model. Initially, we achieved an accuracy of 53.42% using Gaussian Naïve Bayes algorithm. Finally we got an accuracy of 69.74% using Random Forest, followed by accuracy of 55.59% using Multinomial Naïve Bayes, 55.09% using Bernoulli Naïve Bayes, 58.49% using SVM and 57.65% using k-NN classifiers respectively. Seeing the advancement in sentiment and text classification by deep learning [7, 23], in future we wish to explore deep learning for better accurate model.

Acknowledgments. Thanks to the IJCNLP 2017 shared task Customer Feedback Analysis organizers who have made their datasets available and thanks also to an anonymous reviewer for extensive and useful comments.

References

1. Bentley, M., Batra, S.: Giving voice to office customers: Best practices in how office handles verbatim text feedback. In: Big Data (Big Data), 2016 IEEE International Conference on. pp. 3826–3832. IEEE (2016)
2. Breiman, L.: Random forests. Machine learning , vol. 45, no. 1, pp. 5–32 (2001)
3. Chakankar, A., Mathur, S. P., Venuturimilli, K.: Sentiment analysis of users' reviews and comments (2012)
4. Chaudhary, A., Kolhe, S., Kamal, R.: An improved random forest classifier for multi-class classification. Information Processing in Agriculture , vol. 3, no. 4, pp. 215–222 (2016)

5. Fang, X., Zhan, J.: Sentiment analysis using product review data. Journal of Big Data , vol. 2, no. 1, p. 5 (2015)
6. Gamon, M.: Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In: Proceedings of the 20th international conference on Computational Linguistics. p. 841. Association for Computational Linguistics (2004)
7. Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: A deep learning approach. In: Proceedings of the 28th international conference on machine learning (ICML-11). pp. 513–520 (2011)
8. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 168–177. ACM (2004)
9. Kataria, A., Singh, M.: A review of data classification using k-nearest neighbour algorithm. International Journal of Emerging Technology and Advanced Engineering , vol. 3, no. 6, pp. 354–360 (2013)
10. Kibriya, A. M., Frank, E., Pfahringer, B., Holmes, G.: Multinomial naive bayes for text categorization revisited. In: Australasian Joint Conference on Artificial Intelligence. pp. 488–499. Springer (2004)
11. Kohavi, R., et al.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: IJCAI. , vol. 14, pp. 1137–1145. Montreal, Canada (1995)
12. Li, H., Jiao, R., Fan, J.: Precision of multi-class classification methods for support vector machines. In: Signal Processing, 2008. ICSP 2008. 9th International Conference on. pp. 1516–1519. IEEE (2008)
13. Liu, C.-H., Moriya, Y., Poncelas, A., Groves, D.: Ijcnlp-2017 task 4: Customer feedback analysis. Proceedings of the IJCNLP 2017, Shared Tasks pp. 26–33 (2017)
14. Liu, G., Zhang, X., Zhou, S.: Multi-class classification of support vector machines based on double binary tree. In: Natural Computation, 2008. ICNC'08. Fourth International Conference on. , vol. 2, pp. 102–105. IEEE (2008)
15. McCallum, A., Nigam, K., et al.: A comparison of event models for naive bayes text classification. In: AAAI-98 workshop on learning for text categorization. , vol. 752, pp. 41–48. Citeseer (1998)
16. Mukherjee, S., Bhattacharyya, P.: Feature specific sentiment analysis for product reviews. In: International Conference on Intelligent Text Processing and Computational Linguistics. pp. 475–487. Springer (2012)
17. Pal, M.: Random forest classifier for remote sensing classification. International Journal of Remote Sensing , vol. 26, no. 1, pp. 217–222 (2005)
18. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. pp. 79–86. Association for Computational Linguistics (2002)
19. Townsend, J. T.: Theoretical analysis of an alphabetic confusion matrix. Perception & Psychophysics , vol. 9, no. 1, pp. 40–50 (1971)
20. Yin, D., Hu, Y., Tang, J., Daly, T., Zhou, M., Ouyang, H., Chen, J., Kang, C., Deng, H., Nobata, C., et al.: Ranking relevance in yahoo search. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 323–332. ACM (2016)
21. Yuan, P., Chen, Y., Jin, H., Huang, L.: Msvm-knn: Combining svm and k-nn for multi-class text classification. In: Semantic Computing and Systems, 2008. WSCS'08. IEEE International Workshop on. pp. 133–140. IEEE (2008)
22. Zhang, H.: The optimality of naive bayes. AA , vol. 1, no. 2, p. 3 (2004)

23. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Advances in neural information processing systems. pp. 649–657 (2015)