An Effective Machine Learning Approach for Sentiment Analysis on Popular Restaurant Reviews in Bangladesh

S M Asiful Huda
Department of Computer Science & Engineering
East West University
Dhaka, Bangladesh
2015-1-60-166@std.ewubd.edu

Md. Mohiuddin Shoikot
Department of Computer Science & Engineering
East West University
Dhaka, Bangladesh
Shoikot.33@gmail.com

Md. Anower Hossain
Department of Computer Science & Engineering
East West University
Dhaka, Bangladesh
hossainanower4@gmailcom

Ishrat Jahan Ila
Department of Computer Science & Engineering
East West University
Dhaka, Bangladesh
ishratjahanila@gmail.com

Abstract— **Sentiment analysis or text mining is making a huge field of research in this cutting-edge period of social media. Different web journals and Social Media (Facebook, Twitter, Instagram) are the most prevalent stage for the consumers and users where most of the time they express their judgment about trending topics, different brands, restaurants, films, books and so on. Analyzing sentiment is an exceptionally brilliant and viable way to discover people views about news, place, restaurant, film, book, brand. It is helpful for both the owners and sellers. Researchers in the fields of natural language processing, machine learning, data mining, and others have tried an assortment of techniques for automating the process of sentiment analysis. In this study, we built a model using natural language processing techniques and machine learning algorithms to automate the approach of classifying a review on around 200 popular restaurants of Bangladesh as Satisfactory or Poor. This would greatly help the owners to gather a view about the consumers on their restaurant. In this paper, we developed an effective Machine Learning approach to build a model that can predict the sentiment by analyzing the customer's review of a restaurant. Our model achieved an accuracy of 95% using Support Vector Machine Classifier besides other classification models.**

Keywords—sentiment analysis, reviews of restaurant, satisfactory, poor, machine learning.

I. INTRODUCTION

Sentiment Analysis is the way toward deciding if a portion of writing is negative, positive or neutral. or on the other hand nonpartisan. This piece of writing could be a tweet, review about a book, film, movie, restaurant and so on. It tends to be utilized to recognize the client or consumers mentality towards a brand's crucial factors, for example, tone, context, emotion and so forth. These sorts of reviews are equally important for both the consumers and the betterment of the service. From consumers perspective having a view over that service from other consumers is useful for him to get an overall idea of the product. On the other hand, owners or service providers use sentiment analysis to have a view about the acceptance of their products or to analyze customer satisfaction and suggestions. But as we can assume it is a very lengthy and time-consuming approach to go through that huge number of reviews and manually analyze the sentiment of those contents. Using sentiment analysis, the overall result about the opinions and views can be obtained within seconds. It not only gives the owners an idea about the consumers, but it also gives them a better picture of how they stack up against their competitors'-Company has a 20% negative notion, is that terrible? It depends. In the event that its rivals have a generally half positive and 10% negative estimation, while X's is 20% negative, that merits more disclosure to comprehend the drivers of these feelings. Realizing the conclusions related to contenders help organizations assess their very own execution and scan for
Approaches to improve. [1]

II. RELEVANT WORK AND DATASET

Sentiment analysis, additionally called opinion or text mining, is a type of data extraction from the content of developing research and commercial interest. Various scientists endeavored to investigate this field and there are as of now substantial quantities of research papers particularly in this subject. Our work is motivated by the ongoing progressions using Machine Learning Techniques. Sentiment analysis can be achieved in several levels like word, sentence or document. In most of the reviews, people express different assessment about features. In this way, viewpoint/feature-based sentiment analysis would be the most reasonable method for our work. Aside from this, we look to find the sentiment extracted from any review which will help the restaurant owners as well as the consumers to get an overall idea.

There has been huge research so far in the field of text analysis or sentiment analysis [5] [6] [7]. G and et al., (2005) presented an overall picture of recommended systems. According to their work, there are three categories in recommendations methods, those are collaborative, content-based, and hybrid recommendation approaches. However, there are some limitations to these recommendation approaches [2]. Several extensions that can improve recommendation capabilities were proposed by them, as well as enlarging recommendation systems applicable to a broader range of application or systems.

Twitter data has been the center point of sentiment analysis research Nowadays [8] [9] [10]. There have been a significant number of researches going around Twitter as it is one of the largest social sites on the internet. Bac Le and et al, (2015) proposed a sentiment analysis model using around 200000 tweets and based on Support Vector Machine and Bayesian Algorithm [3]. Ziqiong Zhang and et al., (2011) used the SVM and naive Bayes and algorithms to automatically classify reviews as positive or negative into the domain of online Cantonese-written restaurant reviews. The impacts on the classification model of feature sizes and feature presentations are also discussed in the work. Gayatree Ganu and et al., (2009) did similar work in the field of text mining. A raw text format review is difficult for computers to understand, analyze, and aggregate, this paper proposes a regression-based recommendation method to identify the information in the text reviews that take the textual component of user reviews into consideration [4].

A. Dataset

In this work, we used restaurant reviews as our reference dataset. We collected almost 700 reviews from around 200 of the most popular restaurants in Bangladesh. As it is very much tough to collect this much reviews from only one restaurant, we collected reviews consisting of both positive and negative reviews of multiple restaurants. These reviews were manually collected one by one from a famous food review group in Bangladesh named “FOODBANK” where people all around the country share their opinions and views about foods and environments of the restaurants and give their personal ratings on that. Each review was labeled as satisfactory or poor. This labeling was done based on the kind of review. If the review had 7 or more rating out of 10 it was labeled as satisfactory and if it had a rating of 4 or less, the review was labeled as a Poor review. So, the dataset has two columns, the first column contains the sentiment which is either “Satisfactory” or “Poor” and the second column contains the raw text review corresponding to that sentiment. So, it can be easily understood if any review is in favor of that restaurant or not. Our task is to build a model that would be able to classify the sentiment by analyzing the text review.

poor	Ice cream machine is always down, staff is rude and ghetto, food is always old. I always try to avoid it and go to a different one.
Poor	Tasteless chicken and potato wedges almost burnt, there was less salt In vegetable. Rice was full of oil serving. Wont come to this place again. Poor service
Poor	The cashier had no care what so ever on what I had to say it still ended up being wayyy overpriced.
Poor	I was disgusted because I was pretty sure that was human hair.
Satisfacto	The selection on the menu was great and so were the prices.
Poor	Although I very much liked the look and sound of this place, the actual experience was a bit disappointing.
Satisfacto	Phenomenal food, service and ambiance.
Poor	Bland... Not a liking this place for a number of reasons and I don't want to waste time on bad reviewing.. I'll leave it at thatï¿½
Satisfacto	My fiancï¿½ and I came in the middle of the day and we were greeted and seated right away.
Poor	So don't go there if you are looking for good foodï¿½
Satisfacto	I had heard good things about this place, but it exceeding every hope I could have dreamed of.
Poor	Ordered a double cheeseburger & got a single patty that was falling apart (picture uploaded) Yeah, still sucks.
Satisfacto	Everyone is very attentive, providing excellent customer service.
Poor	We waited an hour for what was a lunch I could have done 100 times better at home.
Poor	This greedy corporation will NEVER see another dime from me!
Poor	in consistent Quality! Late Nights the food is bad, but during busy hours its good!
Poor	There is nothing authentic about this place. the spaghetti is nothing special whatsoever.
Satisfacto	The vegetables are so fresh and the sauce feels like authentic Thai.
Poor	I left with a stomach ache and felt sick the rest of the day.
Poor	This place deserves one star and 90% has to do with the food.

Figure 01: Glimpse of the Dataset

III. METHODOLOGY

From the above discussion, we already came to know that there are two classes in a restaurant review, whether it is satisfactory and poor. To classify a restaurant review, we need to understand the problem definition first, then we go for our model and evaluate the result. Machine Learning is very much reached with its algorithms but some of them are good for this kind of detection and some are on an average scale. Our focus is onto some natural language processing techniques which would help us to mine the raw text data and predict the final outcome that we are looking for. From the idea of sentiment analysis and previous studies, we find out that Word Vectorization or Bag of Word model can be a great feature as restaurant reviews may contain a lot of similar words like good, excellent, satisfactory and so on.

A. Algorithms

In our model, we implemented 4 unique machine learning algorithms and for that purpose, we used Python 3.7 as our programmable language. The algorithms are Naïve Bayes, Logistic Regression, Support Vector Machine and Decision Tree. These algorithms are useful for various purposes and they got their very own properties and performance measures depending on various type datasets. As we said before Naïve Bayes, Logistic Regression and SVM are most commonly used algorithms for classification purpose, In our model Logistic Regression gave an accuracy of 88% which is the highest among the other algorithms.

We will implement the following different ideas in order to obtain relevant features from our dataset. [11] [12] Those are as follows,

1.Count Vectors
2.TF-IDF
  a. Word Level

Count Vectors: Count Vector is a data set matrix notation in which each row represents a corpus document, each column represents a corpus term, and each cell represents the frequency count of a particular term in a particular document.

TF-IDF: In the document and in the whole corpus, the TF-IDF score represents the relative importance of a term. The TF-IDF score consists of two terms: the first calculates the standardized Term Frequency (TF), the second term is the Inverse Document Frequency (IDF), which is calculated as the logarithm of the number of documents in the corpus divided by the number of documents in which the term appears.

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)
IDF(t) = log_e(Total number of documents / Number of documents with term t in it)

TF-IDF Vectors can be generated at different levels of input tokens (word level)

a. **Word Level TF-IDF** : Matrix in different documents representing tf-idf scores of each term.

B. Preprocessing the dataset

To apply these algorithms, first, we need to process our dataset. We used Pandas, NumPy, Scikit-Learn, to clean and preprocess our dataset. As we are dealing with text data, we need to make a sequence over our text data so that our raw text data will be transformed into a clean collection of text which we call Corpus. Then it would be much easier to extract feature Vectors from that Corpus array and then we will be able to train our model with those features.

i. **Odd Characters Removal:** As the dataset contains raw text collected directly from People's food review, those review contains various emojis, hashtags, and some irrelevant space and characters. To have a clean text that contains only the words from those sentences we cleaned the whole dataset with the help of regular expression and NLTK library.

ii. **Lower Characterization:** If the same word remains in the format of capital and small letter, the corpus may contain the same word twice. To avoid that situation, we converted all the letters only to small letters.

iii. **Stop Words Removal:** A stop word is a word that is used very commonly in a sentence such as "the", "a", "an", "in".In general, any search engine is programmed in such a way so that it avoids these characters while search query. We would not want these words in taking up space in our corpus or increasing valuable processing time. For this, we can remove them easily, by storing a list of words that you consider to stop words. NLTK (Natural Language Toolkit) in python has a list of stop words stored in 16 different languages.

Stemming: The idea of stemming is kind of normalizing method. There are many variations of a word which carry the same meaning. Every word Having individual dictionary entries would be highly expensive, redundant and inefficient especially since, once we convert to numbers, the "value" for every word is going to be identical. For this reason, we stem the word to shorten the length of lookup and so normalize the sentence. One of the most popular stemming algorithms is Porter Stemmer that we used to stem on our dataset.

C. Feature Extraction

After being done with the several preprocessing methods applied over the whole dataset, we now have a corpus that contains the reviews having only the root words. Now we are ready to extract feature vectors for our model. We choose the Bag of Words model as the main feature to build the model. The bag of words model is explained below.

Bag of words (BoW): A bag of words or simply known as (Bow) is a popular way of vectorizing the words for use in modeling. This technique is widely used in language processing or text mining techniques. This approach is very much simple and easy to implement and can be used in various ways for extracting features from documents.

A bag of words simply represents the appearance of words in a document. The model is only concerned with whether some known words appear in the document and not the position of the words in the document. It means we look at the histogram of the words within our preprocessed corpus by assuming each word count as a feature. This whole technique works in three steps,

1. **Collecting the data**: The first and foremost task is to prepare the data for implementing the Bag of words model which we already have as a corpus where the documents contain only the root words of each review.
2. **Designing the vocabulary**: The second step is to design the vocabulary. This simply means we are making a list of words out of the whole corpus to easily extract the number of appearances of known words in a document.

For example, if there are some reviews like

"The environment, as well as the food, was quite excellent"
"The environment was not up to our expectation"

The corresponding vocabulary for these two reviews would be,

- "The"
- "environment"
- "as"
- "well"
- "food"
- "was"
- "quite"
- "excellent"
- "not"
- "up"
- "our"
- "expectation"

3. **Create document vectors**: Now at the final stage, the goal is to turn every document of free text that we have in our vocabulary into a vector that we can use as input or output of a machine learning model. The simplest method is to mark the appearance of words as a Boolean value. 0 for absent, 1 for the present. For example, using the arbitrary list of words shown above in our vocabulary we can sequence over the first document ("The environment, as well as the food, was quite excellent") and get a binary vector out of it.

- "The" = 1
- "environment" = 1
- "as" = 1
- "well" = 1
- "food" = 1

- “was” = 1
- “quite” = 1
- “excellent” = 1
- “not” = 0
- “up” = 0
- “our” = 0
- “expectation” = 0

As a binary vector, this would look as follows
[1, 1, 1, 1, 1, 1, 1, 1, 0 ,0 0,0]

All ordering of the words is nominally tossed out and then we now have a compatible way of extracting features from the corpus from each of the document and ready for use in modeling. If any of the New lines or document overlap with the vocabulary of known words, but also may contain words outside of the vocabulary, can still be encoded, where only the appearance of known words is marked as scored and unknown words are avoided.

IV. IMPLEMENTATION

After having completed the word vectorization and TF-IDF vectorizer with the help of Bag of words model we are now ready to implement our model. Five different classification was used in here. Based on the comparison we have also considered the precision and recall value between the models. A combination of five different models is tested sequentially in order to improve the accuracy of prediction. The word vectors are used as input or train data in the model and the corresponding sentiment that was either “Satisfactory” or “Poor” as the output of the model. The task is going to be whenever a new document comes to analyze it and predict the sentiment of it whether the category of that review is “Satisfactory” or “Poor”. Using scikit learn, these models are implemented to learn from the training data using 70% data as train data and then tested upon the rest 30% data. To ensure the efficiency of our model the reviews were shuffled all over the dataset. Then we have evaluated the performance of the individual models based on their classification metrics and accuracy.

V. RESULTS

Firstly, **Naive Bayes Model** was tested on the feature vectors mentioned above. It gave an overall accuracy of 94% on word count vectorizer and 84% accuracy on TF-IDF vectorizer on feature.
Then A **Decision Tree** was performed on those same features. This time the performance was slightly underperforming than before, giving an accuracy of 78% and 83% respectively in Word Count Vectorizer and TF-IDF vectorizer.
Thirdly **Support Vector Machine** was performed for observing any further enhancement over the previous result so far. A mentionable enhancement was observed in this case over Decision tree and Naïve Bayes and Decision Tree. The SVM model gave an accuracy of 88% using word count vectorization and 95% using TF-IDF vectorizer. This is the best accuracy we have got over our dataset so far.
Last and finally **Logistic Regression Classifier** was implemented by following the same criterions. Out of the four-classification model that we have used so far, Logistic Regression gave an accuracy of 91% and 88% respectively using Word Count Vectorization and TF-IDF vectorization technique. The graph below shows the accuracy comparison between four classification models using those two feature vectorization techniques.

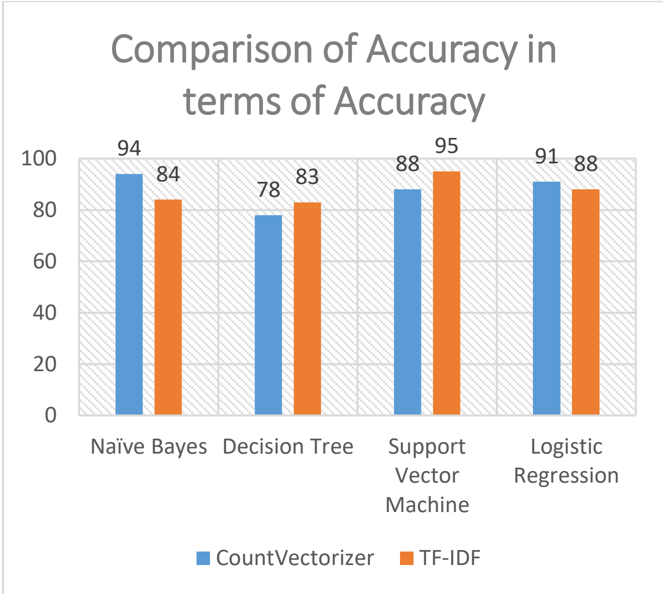

Figure 02: Comparison between models in terms of accuracy

Now as we know Only the Accuracy metric doesn’t express the overall picture. Despite having a higher accuracy, the model may be underperforming if the actual costs that we assign to every error of its prediction are not equal. For example, while dealing with a rare but fatal disease, the cost of failing to predict the disease of a real sick person is much higher than the cost of sending a healthy person to more tests for diagnosis. Therefore, we also look at some other classification metric to get a clear idea of how our model is performing.
The table below shows the Precision, Recall and F1 Score of each of the model that has been implemented over our dataset.

Models	Class Labels	Precision	Recall	F1 Score	Accuracy
Naïve Bayes	**Satisfactory**	**0.81**	**0.93**	**0.87**	**94%**
	Poor	**0.98**	**0.94**	**0.96**	
Decision Tree	Satisfactory	0.66	0.75	0.70	78%
	Poor	0.93	0.89	0.91	
SVM	Satisfactory	0.68	0.89	0.77	88%
	Poor	0.97	0.88	0.92	
Logistic Regression	Satisfactory	0.79	0.82	0.81	91%
	Poor	0.95	0.94	0.94	

Table 01: Precision, Recall & F1 Score of different classification models using Word Count Vectorizer.

Models	Class Labels	Precision	Recall	F1 Score	Accuracy
Naïve Bayes	Satisfactory	0.86	0.40	0.55	84%
	Poor	0.84	0.98	0.90	
Decision Tree	Satisfactory	0.62	0.81	0.70	83%
	Poor	0.93	0.84	0.88	
SVM	**Satisfactory**	**0.88**	**0.94**	**0.91**	**95%**
	Poor	**0.98**	**0.96**	**0.97**	
Logistic Regression	Satisfactory	0.90	0.57	0.70	88%
	Poor	0.88	0.98	0.93	

Table 02: Precision, Recall & F1 Score of different classification models using TF-IDF Vectorizer.

As we can observe from the above tables in terms of precision, recall and F1 score, The Bayesian model is the best performing model out of the rest algorithms when word count vectorization is considered as the feature vectorization technique and Support Vector Machine performs even better than the Naïve Bayes Model when TF-IDF vectorizer is used for generating the feature vectors to train the model. **As the accuracy of Support Vector Machine is the maximum out of the rest three classification models it can be concluded that SVM performs better compared to the other classifiers implemented over our dataset using TF-IDF vectorizer.**

VI. FUTURE WORK

There are several interesting options for future work. One is to make use of other strong and powerful features available in the NLTK Library. Moreover, the dataset only contains nearly 700 reviews, it can be extended to more reviews to improve the performance of the model. Our future work includes focusing on cases harder to classify, in particular when the review is Satisfactory, but the overall scenario of the review is somewhat misleading.

The classification accuracy probably would be significantly improved by means of using a more complex model. It is worth noting, that even with the given dataset, only part of the information was used. Deep learning models like LSTM can come handy while talking about such sentiment analysis problem which we look to implement in the near future.

The research showed, that even quite simple artificial intelligence algorithm may show a good result on such an important problem as Sentiment Analysis. Therefore, the results of this research suggest that artificial intelligence techniques may be successfully used to tackle this kind of important problem.

VII. CONCLUSION

Numerous nearby restaurant entrepreneurs in Bangladesh run out of business due to the absence of an effective method for examining their overall customer feed. In order to secure their business investments and comprehend the general consumer satisfaction, we came up with the idea of automating this procedure using machine learning techniques, one of the powerful and strong approach at present in this kind of problems. In this paper, we implemented an automated model for classifying the sentiment from a user's review over a restaurant, which provides general solutions for the restaurant owners to get an overview from huge reviews of their customers. As mining that much of the review and getting an overview is a time-consuming lengthy process. By using this model, they will be able to get an idea about their restaurant and how well they stack up to their competitors within seconds. This work demonstrates an automated system for detecting Sentiments over the restaurant reviews. Such a system could be valuable to social media users as at present people tend to judge any restaurant by seeing other people's views. From that perspective, this kind of research can have a significant impact on future research purposes in the fields of Sentiment Analysis.

REFERENCE

[1] IPULLRANK blog, http://ipullrank.com/step-step-twitter-sentiment-analysis-visualizingunited-airlines-pr-crisis/, Accessed on 10th August 2017.

[2] E. Boiy, M. F. Moens," A machine learning approach to sentiment analysis in multilingual Web texts", Springer-Information Retrieval Journal, 2008, Belgium.

[3] B. Le, H. Nguyen, "Twitter Sentiment Analysis Using Machine Learning Techniques","Springer", 2015.

[4] G. Ganu, N. Elhadad, and A. Marian, "Association for Computational Linguistics", beyond the stars: Improving rating predictions using review text content. In WebDB, volume 9, pages1–
6.2009.

[5] M. Govindarajan, "Sentiment Analysis of Restaurant Reviews Using Hybrid Classification Model", "IRF International Conference", 9th February 2014, Chennai India.

[6] D. V. N. Devi, C. K. Kumar, S. Prasasd ,"A feature Based Approach for Sentiment Analysis by Using Support Vector Machine", "IEEE 6 International Conference on Advanced Computing", 2016, India .

[7] Y. Woldemariam, "Sentiment analysis in a cross-media analysis framework," *2016 IEEE International Conference on Big Data Analysis (ICBDA)*, Hangzhou, 2016, pp. 1-5. doi: 10.1109/ICBDA.2016.7509790

[8] X. Fan, X. Li, F. Du, X. Li and M. Wei, "Apply word vectors for sentiment analysis of APP reviews," *2016 3rd International Conference on Systems and Informatics (ICSAI)*, Shanghai, 2016, pp. 1062-1066. doi: 10.1109/ICSAI.2016.7811108

[9] V. Ikoro, M. Sharmina, K. Malik and R. Batista-Navarro, "Analyzing Sentiments Expressed on Twitter by UK Energy Company Consumers," *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, Valencia, 2018, pp. 95-98.doi: 10.1109/SNAMS.2018.8554619

[10] P. Porntrakoon and C. Moemeng, "Thai Sentiment Analysis for Consumer's Review in Multiple Dimensions Using Sentiment Compensation Technique (SenseComp)," *2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Chiang Rai, Thailand, 2018, pp. 25-28. doi: 10.1109/ECTICon.2018.8619892

[11] Towards Data Science. (2019). *Process Text using TFIDF in Python*. [online] Available at: https://towardsdatascience.com/tfidf-for-piece-of-text-in-python-43feccaa74f8 [Accessed 31 Mar. 2019].

[12] Towards Data Science. (2019). *Natural Language Processing: Count Vectorization with scikit-learn*. [online] Available at: https://towardsdatascience.com/natural-language-processing-count-vectorization-with-scikit-learn-e7804269bb5e [Accessed 31 Mar. 2019].