International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 08 | Aug 2019 www.irjet.net p-ISSN: 2395-0072

Sentimental Analysis for Online Reviews using Machine learning Algorithms

Babacar Gaye[1], Aziguli Wulamu [2]

[1]Student, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing CHINA
[2]Professor, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing CHINA

Abstract - *Sentimental analysis, as a general application of natural language processing, refers to extracting emotional content from text or verbal expressions. Online monitoring and listening tools use different approaches to construe and describe emotions with different performance and accuracy. The quantitative measurement of services and business products is measured with the help of market feedback, but the qualitative measurement of accuracy is complicated, which needs interpretation of consumer feedback using features extractions and machine learning algorithms such as support vector machine, random forest, XGBoost. The enlisting of accurate analysis and interpretation of sentiments. Considering the sentiment analysis XGBoost classifier has higher accuracy and performance than SVM, and random forest. that says the performance is better in case of sentiment analysis*

Key Words: Classification, SVM, Random forest, XGBoost, Sentiment Analysis.

1. INTRODUCTION

The sentiment analysis is mainly used for internal business needs (analytics, marketing, sales, etc.). Another application of sentiment analysis is the automation of recommendation modules integrated into corporate websites. These modules are used to predict the preferences of a given user and to suggest the most appropriate products.

Models based on machine learning for the detection of emotions require annotated corpora to lead a model that can take into account different specificities (including pragmatics). The development and deployment of such models reduces working time and the models themselves can achieve a good performance.

To train a machine learning model is to develop a set of automatically generated rules, which drastically reduces development costs. Textual cues and dependencies related to a feeling may not be visible at first human sight but are easily detected by a machine that encodes this information into a model.

To indicate that a given message expresses anger (which implies the prior annotation of a corpus by an expert) is sufficient for the algorithm to detect the "anger hints" automatically and saves them for future use.

E-commerce uses Reputation-based trust models to a greater extent. Reputation trust score for the seller is obtained by gathering feedback ratings. considering the fact that shoppers mostly express their feelings in free text reviews comments and by mining reviews, we propose Comment Trust Evaluation. They have proposed a model which is based on multidimensional aspects for computing reputation trust scores from user feedback comments. In this research paper we used TF-IDF, and we made a comparison of machine learning algorithms such as XGBoost, SVM and RandomForest and we made a data visualization of the result using Matplotlib.

This paper will present how to do text mining for product reviews using machine learning algorithms, the second part is the literature survey, the third part will be the methology, part four is the experiments and results of our research and the last part will be data visualization.

1.1 Literature reviews

In [1] the author wrote about hot opinions of the products comments using hotel comments dataset as the main research. They filtered the data from the length of the comments and the feature selection aspect by analyzing the characteristics of customer's reviews they have built a mathematical model for the preprocessing and adopt the clustering algorithm to extract the final opinions. They compared it with the original comments, the experiment results were more accurate.

In [2] in this paper the author categorized the descriptive and the predictive and separated them using data mining techniques. The statistical summary he made was mostly for the descriptive mining of the online reviews.

In [3] the main objective in this research is to extract useful information in case of a big data. Clustering: Cluster analysis is the task of grouping a set of objects in in a way that objects in the same group that you call cluster are more similar to each other than to those in other groups.

Clustering types are density based, center based, computational clustering, etc.

In [4] D. Tang et al. Did a learning continuous word representation for Twitter sentiment classification for a supervised learning framework. They learn word embedding by integrating the sentiment information into the loss functions of three neural networks. Sentiment-specific word embeddings outperform existing neural models by large margins. The drawback of this model this author used is that it learns sentiment-specific word embedding from scratch, which uses a long processing time.

The classifications algorithms have an impact on the accuracy on the result in polarity, and hence, a mistake in classification can result in a significant result for a growing business monitoring strategy [5].

2. IMPLEMENTATION

1. Dataset

Reviews can be downloaded using two methods, which are twitter streaming API and API. However, in this research, we used a balanced products reviews dataset with 4 columns and 3 classes negative, positive, and neutral reviews.

2. Sentiment Analysis

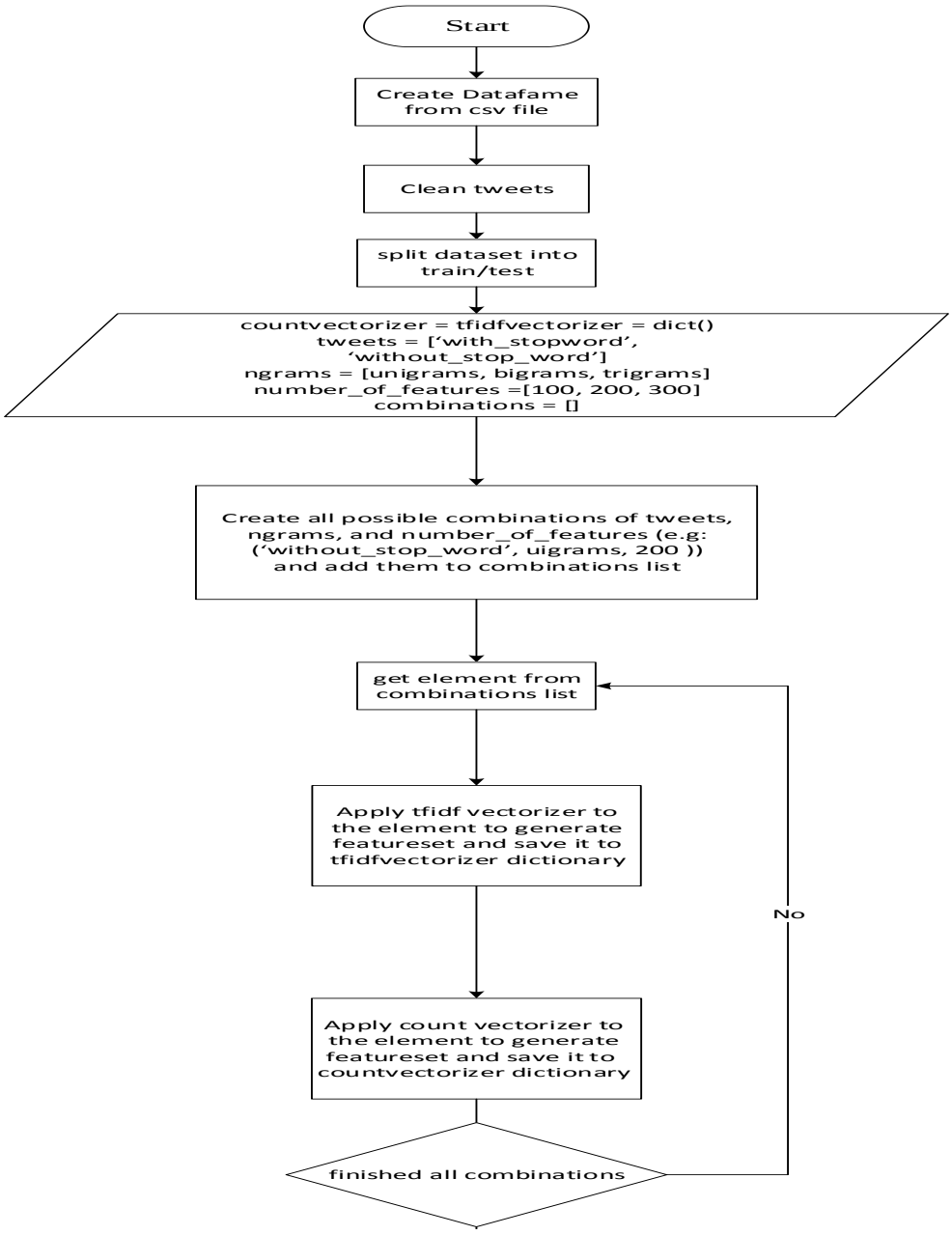

Chart -1 implementation procedure

2-1. Data preprocessing

The transformation of raw data into usable training data is referred to as data preprocessing. The steps we used to preprocess this data for our research is as follows:

We defined the preprocessing function self.data.columns = ['tweet', 'brand', 'label']

-Remove all I cannot tell

-Convert labels into one word

-Remove null tweets

After the first part we define the clean tweet function in our code

-remove web links (http or https) from the tweet text

-remove hashtags (trend) from the tweet text)

- remove user tags from tweet text

- remove re-tweet "RT"

-remove digits in the tweets

-remove new line character if any

-remove punctuation marks from the tweet

-convert text in lower case characters (case is language independent)

-remove extra-spaces

2.2 features extraction

Sklearn has several vectorizers to process and tokenize text in the same function, and it involves converting word characters into integers.

In our research, we used 2 methods:

Countvectorizer and Tf-IDF vectorizer

2. Count Vectorizer

We did a loop over n_gram to create unigram, bigram, trigram dataset

```
# get CountVectorizer instance
cv = CountVectorizer(min_df = 2, max_df = 0.5, ngram_range=n_gram[n],
                     max_features=f, preprocessor = lambda x: x,
                     tokenizer = lambda x: x)
```

Box -1: code snippet Tf-idf Vectorizer

2. TF-IDF Vectorizer

We converted our dataset to a matrix of token counts:

```
# get TfidfVectorizer instance
tfidf = TfidfVectorizer(min_df = 2, max_df = 0.5, ngram_range=n_gram[n],
            max_features=f, preprocessor = lambda x: x,
            tokenizer = lambda x: x)

x_train, x_test, y_train, y_test = train_test_split(tweets[t],
                                                    self.data.label,
                                                    test_size = 0.2)
```

Box -2: code snippet Tf-idf Vectorizer

3-Classifiers

We used 3 classifiers to do the sentiment analysis on our dataset:

3.1Support Vector Machine (SVM)

Support Vector Machine (SVM) It is a classifier that uses multi-dimensional hyperplanes to make the classification. SVM also uses kernel functions to transform the data in such a way that it is feasible for the hyperplane to partition classes effectively [8]. It's also a supervised learning algorithm that can analyze the data and recognize it's patterned [6]. You give an input set, SVM classifies them as one or the other of two categories. SVM can deal with non-linear classification and linear classification.

```
# SVM classifiers
elif cls == 'svm':
    param = {'C': [0.001, 0.01, 0.1, 1, 10],
             'gamma' : [0.001, 0.01, 0.1, 1]}

    tune = RandomizedSearchCV(estimator=classifiers[cls], param_distributions=param)

    tune.fit(countvectorizer[key]['x_train'],
             countvectorizer[key]['y_train'])

    classifiers[cls].set_params(C=tune.best_params_['C'],
                                gamma=tune.best_params_['gamma'])
```

Box -3: code snippet for SVM classifier

3.2 Random Forest

Random Forest classifier chooses random data points in the training dataset and creates a series of decision trees. The last decision for the class will be made aggregation of the outputs from all the trees [9] RandomForest is a supervised learning algorithm that can be used for regression and classification. Random forest generates a decision tree on randomly selected samples from the dataset and obtain the predictions from each tree and chooses the best solution by applying to vote. Random samples will be used to create decision trees and based on the performance of each tree. The best sub-decision trees will be selected [7].

```
# randomforest classifiers
elif cls == 'rf':
    param = {'n_estimators': [150, 170, 200],
             'max_depth' : [3, 5, 7, 10]}

    tune = RandomizedSearchCV(estimator=classifiers[cls], param_distributions=param)

    tune.fit(countvectorizer[key]['x_train'],
             countvectorizer[key]['y_train'])

    classifiers[cls].set_params(n_estimators=tune.best_params_['n_estimators'],
                                max_depth=tune.best_params_['max_depth'])
```

Box-4: code snippet random Forest classifier

3.3 Extreme Gradient Boosting

The third classifier we used in our research is Extreme gradient boosting (XGBoost) [10]. XGBoost is a scalable machine learning approach which has proved to be successful ina lot of data mining and machine leaning challenges [11]

```
# XGBoost classifiers
if cls == 'xgb':
    param = {
    'learning_rate': [0.001, 0.01, 0.1, 0.5],
    'n_estimators': [150, 170, 200],
    'max_depth': [3, 5, 7, 10],
    'subsample': [0.3, 0.5, 0.9]}

    tune = RandomizedSearchCV(estimator=classifiers[cls], param_distributions=param)

    tune.fit(countvectorizer[key]['x_train'],
             countvectorizer[key]['y_train'])

    classifiers[cls].set_params(learning_rate=tune.best_params_['learning_rate'],
                                n_estimators=tune.best_params_['n_estimators'],
                                max_depth=tune.best_params_['max_depth'],
                                subsample=tune.best_params_['subsample'])
```

Box-5: code snippet for XGBoost classifier

For each of this classifier we used random search in order to choose the best hyper parameters, we have multiples for loops that are intersected such as Different classifiers, with and without stop words, numbers of features. This in total gave us all the possible keys.

4. RESULTS

The results are evaluated on comparison for the best classifier accuracy among the 3 classifiers we used on this research such as naïve Bayes, random forest and XGBoost. For each given machine learning algorithm, we did the classification by choosing 100, 200, 300 features for the unigram, bigram and trigram with and without stop words. After we compared the accuracy between the 3 classifiers by fixing the numbers of features, afterwards we draw the best of the best results using MatPlotlib.

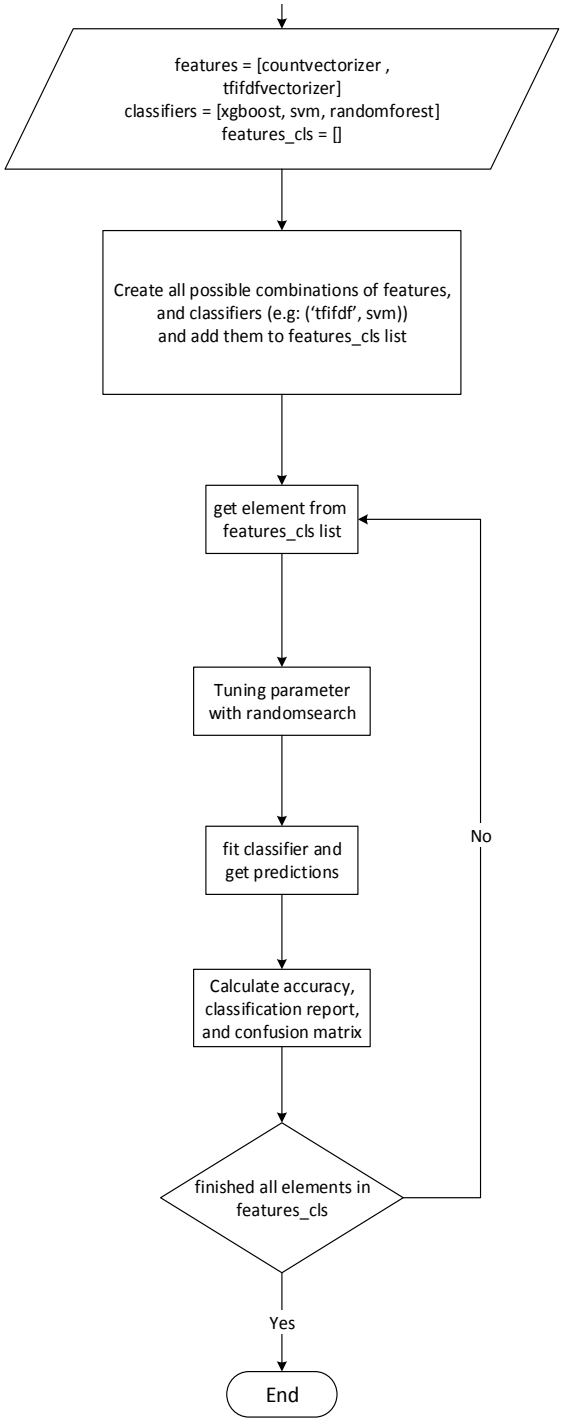

Chart2: Prediction Procedure

This chart is the reference of all the steps we have used to do the sentimental analysis for this dataset and here are the results of 3 algorithms used on this research. We have used Matplotlib bar graphs to show the results of the experiments.

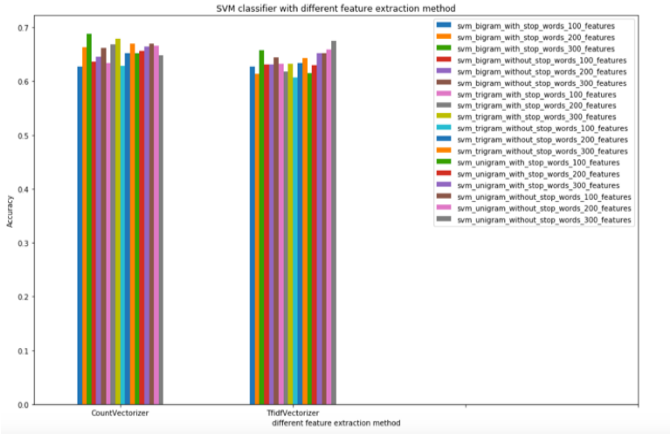

Fig. 1 support vector machine feature extraction results

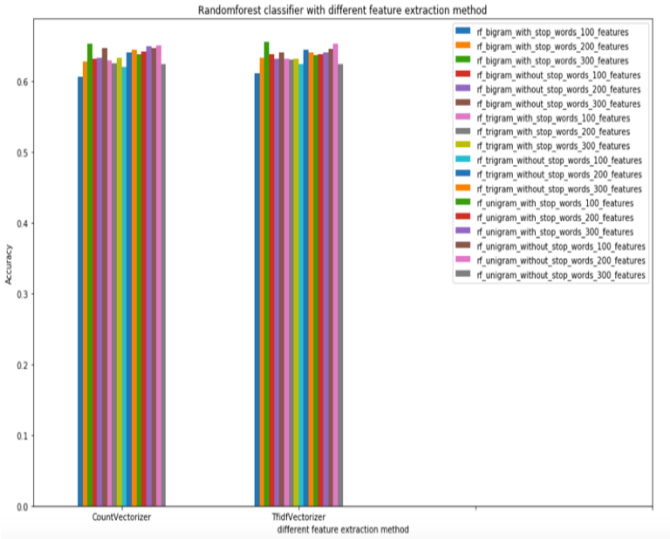

Fig. 2 Random Forest feature extraction results

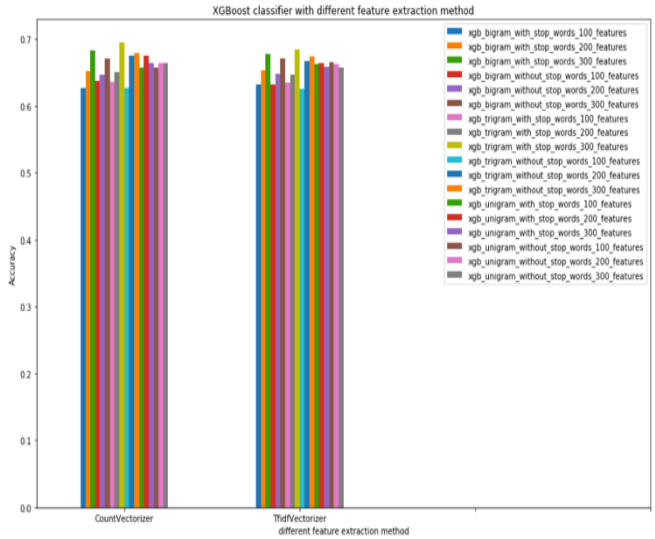

Fig. 3 XGBoost feature extraction results

According to our results we can say that if accuracy is your priority, we should consider a classifier like XGBoost that uses high has the best accuracy. If processing and memory are small, then Naïve Bayes should be used due to its low memory and processing requirements. If less training time is available, but you have a powerful processing system and memory, then Random Forest can be considered

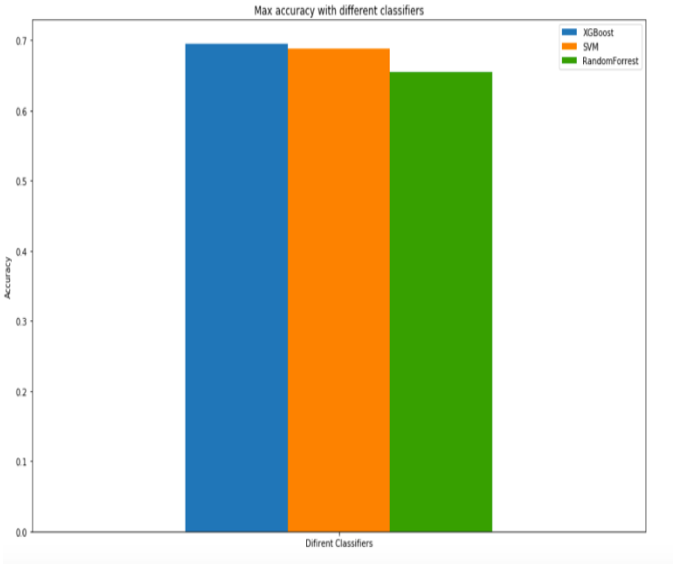

Fig. 4 comparison results of the 3 classifiers

CONCLUSION

Based on results, in conclusion we can that for the context of sentiment analysis, XGBoost has a better performance because it has a higher accuracy.

In sum, we can see that every classification 1algorithm has drawbacks and benefits. Considering the sentiment analysis XGBoost classifier has higher accuracy and performance than SVM, and random forest. That says the performs better in case of sentiment analysis. Random Forest implementation also works very well. The classification model should be chosen very carefully for sentimental analysis systems because this decision has an impact on the precision of your system and your final product. The overall sentiment and count based metrics help to get the feedback of organization from consumers. Companies have been leveraging the power of data lately, but to get the deepest of the information, you have to leverage the power of AI, Deep learning and intelligent classifiers like Contextual Semantic Search and Sentiment Analysis.

REFERENCES

[1] Vijay B. Raut, D.D. Londhe, “Opinion Mining and Summarization of Hotel Reviews”, Sixth International Conference on Computational Intelligence and Communication Networks, Pune, India,2014.

[2] Betul Dundar, Suat Ozdemir, Diyar Akay "Opinion Mining and Fuzzy Quantification in Hotel Reviews", IEEE TURKEY, 2016

[3] Apurva Juyal*, Dr. O. P. Gupta "A Review on Clustering Techniques in Data Mining" International Journal of advanced computer science and software engineering Volume 4. Issue 7, July 2014

[4]. D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin "Learning sentiment-specific word embedding for twitter sentiment classification", 2014

[5] Bo Pang and Lillian Lee "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts" in ACL '04 Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, 2004, Article No. 271

[6] Xu, Shuo & Li, Yan & Zheng, Wang. (2017). Bayesian Gaussian Naïve Bayes Classifier to TexClassification. 347-352. 10.1007/978-981-10-5041-1_57.

[7] https://www.datacamp.com/community/tutorials/rand om-forests-classifier-python

[8] Ben-Hur, Asa, and Jason Weston. " A users guide to support vector machines."

[9] Louppe, Gilles. " Understanding random forests: From theory to practice." arXiv preprint arXiv:1407.7502 2014.

[10]. Chen, T.; Guestrin, C. Xgboost: A Scalable Tree Boosting System. *arXiv* **2016**, arXiv:1603.02754.

[11]Phoboo, A.E. Machine Learning wins the Higgs Challenge. *ATLAS News*, 20 November 2014.

[12]. Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis lectures on human language technologies, 5(1), 1-167.

[13]. M Fırat. C Twitter Sentiment Analysis 3-Way Classification: Positive, Negative or Neutral.[2]. "IEEE International Conference on Big Data, 2018. [14]. Md. Daiyan, Dr. S.K.Tiwari , 4, April 2015, "A literature review on opinion mining and sentiment analysis", International Journal of Emerging Technology and Advanced Engineering, Volume 5.

BIOGRAPHIES

Babacar Gaye is currently pursuing a PhD in computer science and technology. His research area is Data science and machine learning at the university of science and technology Beijing.

Aziguli Wulamu is a professor at the school of computer and communication engineering, university of science and technology Beijing.