Advances and Applications in Mathematical Sciences
Volume 20, Issue 3, January 2021, Pages 361-366

SENTIMENTAL ANALYSIS ON TEXT DATA

RITIK GUPTA, PUSHPENDRA KUMAR and VIKASH YADAV

Department of Computer Science and Engineering
ABES Engineering College
Ghaziabad, India
E-mail: ritik.16bcs1176@abes.ac.in
pushpendra.16bcs1157@abes.ac.in
vikash.yadav@abes.ac.in

Abstract

The number of smart phones is increasing as well as the internet growing. The modern Internet provides a lot of opportunities for millions of people around the world to communicate with each other and they can share their ideas, views via email, social media sites like Facebook, twitter etc. It is the cheapest and easiest way of communicating with people. This social networking site has tons of text data. These texts data can be used to analyse the public opinions on certain topics, the emotion expressed on any online platform. There all are forms of sentimental analysis. The basic characteristic of sentimental is classifying the polarity in the text data. These polarities can be any finding of the form like positive, negative or neutral, happy or sad, good or bad. These all are classification problems. Sentimental analysis is also known as opinion mining. For developing the sentimental analysis system which can classify their polarity, it uses text analysis, NLP and various machine learning algorithms.

I. Introduction

Over the years the internet has been growing very fast. Now we all are habitual of the internet. It is now part of daily life we spend most of the time over it. The Internet provides us with many platforms for communication and getting user‟s reviews from sites like Facebook, Twitter, and Amazon. These sites help users to share their opinion worldwide on different types of topics. These opinions sometimes are used to analyse any business product, people or any organization. The business company uses text reviews to analyse the quality of their products. Today's companies rely on the

2010 Mathematics Subject Classification: 62N02, 91C20, 90C08.
Keywords: Polarity, Naive Bayes classifier, Text classification, NLP.
Received May 20, 2020; Accepted July 31, 2020

customer's feedback. There are tons of reviews and feedback is collected. The main problem of these reviews and feedbacks are how to analyse these huge amounts of data with less time with the minimum amount of time. This all can be done by fast and with less error with Machine learning and NLP. Here are some Text analysis algorithms are as follows:

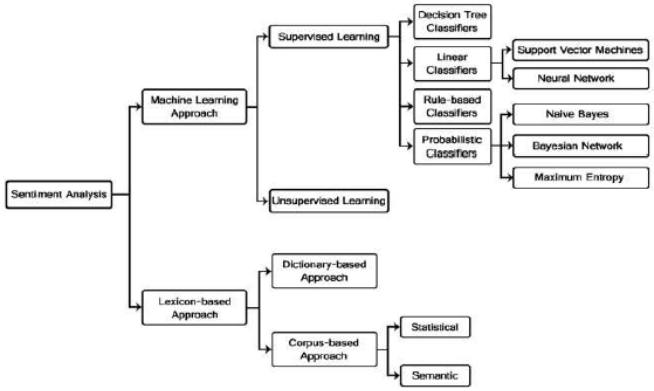

Figure 1. Sentimental Classification Techniques.

We can define sentiments like polarity. It can be positive, negative or neutral. It can be good or bad, and sad or happy any kind of feeling. Here are some examples:

(i) "This cafe is great, the staffs are really friendly and the coffee is delicious." -Positive.

(ii) "I would not recommend this cafe to anyone. Their coffee is terrible and is really expensive."- Negative.

Sometimes these reviews may not be clear as these are. These texts can be in any kind of form. It may be tweets, comments, feedback or sometimes it can be the heading of a newspaper.

II. Related Work

There are many papers are published regarding the sentimental analysis. In [1], the author worked on social media sites like Facebook. From where they gather the messages written by users. On these messages he applied the sentimental analysis and categorized the messages in the

positive, negative and neutral polarity. In [2] author used the corpus data and built the algorithm which can easily classify the data into the three categories which are positive, neutral and negative. The results show that his accuracy and performance on the data is better than previous techniques. In [3] author took the data from twitter. Their data all are tweets, which categorized them in the three parts positive, neutral and negative. He used two types of models. The first is the unigram model and Tree based model.

We collect the data of reviews from Kaggle. It consists of reviews of food from Amazon. The data is of 10 years from Oct 1999 to Oct 2012. It includes roughly 500,000 reviews of 74,258 products from 256,059 users. This dataset contains 10 attributes in which two of them are text which is Summary and Text. The summary is a brief summary of the review and Text is description of the reviews. These reviews contain ratings between 1 to 5. For our analysis we transfer these ratings into positive and negative. The reviews which have rating 4 and 5 we consider the positive and the reviews which have rating 1 and 2 we consider them negative and remove the reviews of ratings of 3 due to neutral.

III. Methodology

Our aim is to find out that that given review is positive or negative. So we use different types of text classification algorithms. These algorithms are as follows and their respective results.

3.1 Naive Bayes Classifier: Naive Bayes classifier [4] is one of the simplest and most commonly used probabilistic classifiers. It computes the posterior probability of the class which is based on the word distribution in the documents. Naive Bayes classifier uses the Naive Bayes Theorem to predict the probability that given features belong to a particular label.

$$P(X) = \frac{P(C)P(C)}{P(X)}$$

$P(c \mid x)$ = Posterior Probability

$P(x \mid c)$ = Likelihood

$P(c)$ = Class Prior Probability

$P(x)$ = Predictor Prior Probability

When we use Naive Bayes Classifier on the BOW, TFIDF, AvgW2V and TFIDF w2v then AUC are as follows 91%, 93%, 94% and 95%.

3.2 *K*-NN (*K*-Nearest Neighbour): *K*-NN (*K*-Nearest Neighbour) [5] algorithm is a simple supervised algorithm that is used in both regression and classification problems. When KNN uses for classification then it uses the class with the highest frequency from the *K*-most similar instances for the calculation of the output. In KNN each element of the data votes for their class and the class which has the highest votes that will take the predictions.

Let Problem is binary class classification:

$$P(class = 0) = \frac{count\ (class = 0)}{count\ (class = 0) + count\ (class = 1)}$$

$$P(class = 1) = \frac{count\ (class = 1)}{count\ (class = 0) + count\ (class = 1)}.$$

When we use KNN with the help of BOW (Bag of words), TFIDF, AvgW2V and TFIDF W2V it gave the AUC of 83%, 87%, 89% and 82% respectively.

3.3 Logistic Regression: Logistic Regression is also from supervised Learning. It is used for binary class classifications. The name of logistic Regression is taken from the logistic function which is of "*S*" shape graph. It is bounded within the range of 0 and 1.

$$y = \frac{e^{B0+B1\times x}}{1 + e^{B0+B1\times x}}$$

y = Predicted Output

$B0$ = Bias or intercept term

$B1$ = Constant for single value(x)

Our predicted output also has two class classification so we are able to

use Logistic Regression. When we use the Logistic regression with BOW, TFIDF, AvgW2V and TFIDF AvgW2V then the AUC are 93%, 95%, 89.7% and 84.5% respectively.

3.4 Decision Tree: Decision Tree is a predictive Machine learning algorithm. Decision tree is also known as CART (Classification and Regression trees). Modern variation of the decision tree is Random Forest. It is a binary tree in which each node represents the single input variable and these variablessplit into other nodes and the leaf nodes represent the output variables. It uses a greedy algorithm to split the tree.

When we use the Decision tree on our data with BOW, TFIDF, AvgW2V, TFIDFW2V then the AUC are 82%, 80.5%, 82.6 and 75.6% respectively.

3.5 SVM (Support Vector Machine): Support vector machine is a supervised machine learning algorithm. Which is used in Regression and classifications problems both. It is one of the most popular algorithms. In this algorithm data points are plotted in the n-Dimensions then we try to find a hyperplane that can separate the two classes very well.

Same as in the algorithm we train the model with the help of BOW, TFIDF, AvgW2v and TFIDFW2v then the AUC are as follows 93%, 96%, 92.5% and 87.6%.

3.6 Random Forest: Random Forest is a modern variation of the Decision Tree algorithm. It is a type of ensemble machine learning algorithm. Random forest is built with many decision trees and in this random sample of training data points is used for building the trees.

When we use the Random forest model it gives AUC which is as follows: 91.4%, 93.24, 88.6% and 83% with the help of BOW, TFIDF, AvgW2V and TFIDF W2v.

3.7 XGBOOST GBDT: After the launch in 2014 it is one of the most used algorithm. Which is mostly used in the hackathons and computations. It is an ensemble algorithm. The output of single model is dependent on several models

The AUC with the XGBOOST GBDT with the help of WOB, TFIDF, AvgW2V and TFIDF W2V are as follows 93%, 94%, 90% and 86%.

IV. Result Analysis

We use the different types of models and analyse the performance on the text data. After the deep analysis on the data the different model has different performance on the data which are in the table.

Sr No.	Approach		AUC		
		BOW	TFIDF	Avg W2Vec	TFIDF W2Vec
1.	Naive Bayes Classifier	91%	93%	94%	95%
2.	K-NN	83%	87%	89%	82%
3.	Logistic Regression	93%	95%	89.7%	84.5%
4.	Decision Tree	82%	80.5%	82.6%	75.6%
5.	SVM	93%	96%	92.5%	87.6%
6.	Random Forest	91.4%	93.24%	88.6%	83%
7.	XGBOOST GBDT	93%	94%	90%	86%

Figure 2. Performance Analysis on Data.

V. Conclusion

In the text data we have implemented different types of machine learning algorithms and also take the help of NLP. We find different algorithms have different performance on the text data but among them some work very well like when we use SVM with the help of TFIDF it gives the AUC of 96% and it is the best AUC among all the algorithms for our Text data.

References

[1] Ortigosa, Alvaro, José M. Martín and Rosa M. Carro, Sentiment analysis in Facebook and its application to *e*-learning, Computers in Human Behavior 31 (2014), 527-541.

[2] Pak, Alexander and Patrick Paroubek, Twitter as a Corpus for Sentiment Analysis and Opinion Mining, LREC. Vol. 10. 2010.

[3] Agarwal, Apoorv, et al., Sentiment analysis of twitter data, Proceedings of the Workshop on Languages in Social Media, Association for Computational Linguistics, 2011.

[4] https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[5] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm