SENTIMENT ANALYSIS FOR MOVIE REVIEWS

BY NIKHITA KANDIKONDA AND MANOJ GOOTAM

Abstract

Reviews are a store house of abundant information and opinions of people. Mining these data gives us an insight of the trends and opinions of various people. It is still presenting a challenging task because of the abundance of data. Hence, Opinion mining or sentiment analysis has become a topic of research in many fields from marketing to stocks. Movie reviews are one such opinion database and are an important way to gauge the performance of a movie. In this study, we explore various Natural Language Processing methods to pre-process the data and apply various classifiers like Naïve Bayes Classifier, Logistic Regression Classifier, Stochastic Gradient Descent Classifier and Support Vector machines to classify our reviews as positive or negative. Multinomial naïve Bayes classifier yields 83% accuracy on the Large Movie Review Dataset. Evidently, sentiment analysis can be used for a variety of applications like stock market predictions, recommender systems and customer service.

I. Introduction

Opinions are central to almost all human activities, with increase in the growth of social media, people now share opinions on the internet and it has become a part of everyday life. Companies often develop new products seeking the opinions of existing customers because they are key influencers of the products. Opinions are usually expressions of emotions on an object of interest which are subjective and user dependent. These opinions contain a lot of information that can be extracted and used for applications like predicting the rise and fall of stocks, movie reviews, marketing, recommendation systems, customer service, etc. Growth in social media coincides with growing importance in opinion mining.

Opinion is an emotional tone behind series of words that have a tone which can be classified as positive or negative, which sometimes depends on the context it is used in. This information on tone is usually referred to as sentiment of the opinion. The process of extracting the sentiment or classifying the tone into positive or negative categories is called sentiment analysis (or opinion mining). It is the process of determining the attitude of a speaker or the overall contextual polarity of an online mention. There are multiple applications of sentiment analysis that can be done today, due to the availability of a wide variety of opinion (text) data on the internet and one such source of opinion data is IMDb.

The IMDb (Internet Movie database) is an online database of information related to films, television programs and video games, including cast, production crew, fictional characters, biographies, plot summaries, trivia and reviews, operated by IMDb.com, Inc., a subsidiary of Amazon.com. It has been estimated that IMDb has about 50,000 movies with related information like the cast, crew, ratings and the user reviews. In this research study, we are applying sentiment analysis on movie reviews to extract the contextual polarity of user reviews and classify them into positive or negative classes.

II. Data Pre-Processing

We used 'Large Movie Review Dataset' which is a collection of 50,000 reviews from IMDB built by a Stanford research group. The data set does not allow more than 30 reviews per movie and contains an equal distribution of positive and negative reviews. The dataset contains only highly polarized reviews, where the negative reviews have a score of ≤ 4 out of 10 and a positive review has a score of ≥ 7 out of 10.

The positive and negative reviews are run through a series of Natural Language Processing methodologies using the NLTK package. A pipeline of text processing is done on the reviews as follows:

- **Cleaning:** White spaces, single letters and digits are replaced by a space.
- **Tokenization:** Each review is divided into collection of tokens by splitting the review with delimiters like space and commas.
- **Stop words removal:** Eliminate the common words such as 'a', 'an', 'of' etc. from these tokens.
- **Lemmatization:** Each word is converted into its root word. For example, sleeping is converted to sleep.
- **Filtering**: Any blank words that are created after all the preprocessing are removed.
- **Low frequency words:** All the words that repeated less than 100 times in 50000 reviews have been removed because these words do not express any sentiment at all. For example: actor or character names like peter parker, Tony Stark, Spiderman etc. these words repeat only a few times especially in reviews for one or two movies
- **High frequency words:** We have also removed the words that repeat a lot of times in both the positive reviews and the negative reviews which also do not express any sentiment for example: get, between, some, before, kind, rather, man, especially, although. These words are neutral words but still repeat a lot of times and removing them is tricky since you might lose some words that have sentiment. To do this, the probabilities of occurrence of each word in positive reviews and negative reviews are multiplied to make it a monotonically decreasing function. The words with the product greater than 0.24 are words that have approximately equal probability in positive and negative reviews implying no sentiment, while the words with products less than 0.24 will have at least (60% - 40%) occurrence in positive or negative reviews and they express a sentiment.

With these filtered words, we generated a bag of words which have a significant sentiment associated with it. This bag of words is used to construct a binary feature vector which marks the presence of these words in each review as binary, either true or false.

III. Sentiment Analysis

Once the word feature vectors of the dataset are derived, a model can be built using the word feature vectors as the input variables and the sentiment as the output variable. This is a binary classification problem because the sentiment has only two classes (positive or negative) in our dataset. Following are some of the issues with the word feature vectors.

- **High dimensional input space:** Because of the large number of words which are used as features, the input space tends to become very large.
- **Few irrelevant features:** To reduce the dimensionality, we could remove some features using feature selection techniques, but often there are very few irrelevant vectors.
- **Word feature vectors are sparse:** Because the number of words in a review (sample) are often only a handful, the feature vectors are highly sparse.

One of the major advantages in text mining problems are that they are linearly separable. Hence, we could apply all the linear classification techniques for sentiment analysis. But, literature suggests that the naïve Bayes classifier and support vector machines perform better when it comes to sentiment analysis. But after reading through the pros and cons of each technique and analyzing the issues with text classification, we have chosen the following binary classifiers from scikit-learn package in python for sentiment analysis.

- Naïve Bayes is a very simple classifier developed on Bayesian approach. Naïve Bayes models allow each attribute to contribute towards the final decision equally and independently from other attributes, in which it is more computational efficient when compared with other text classifiers. In this exercise, we have used two variants of naïve Bayes:
- Multinomial naïve Bayes classifier which implements Laplacian Smoothing
- Bernoulli naïve Bayes which implements classifier assuming multivariate Bernoulli distributions of the data; i.e., there may be multiple features but each one is assumed to be a binary-valued (Bernoulli, Boolean) variable
- Logistic Regression Classifier is a binary classification technique that assumes sigmoid function between the predictors and the output variable
- Stochastic Gradient Descent Classifier which is a simple and efficient approach to discriminative learning for binary classifiers under convex loss functions
- Support Vector machines (SVM) is a discriminative classifier and its ability to learn independent of the dimensionality of the feature space makes it remarkable. Since SVMs have the potential to handle the large feature spaces, it would be a good technique for sentiment analysis.

In the following section, we will discuss the results of the above classifiers on our dataset.

IV. Results

Cross Validation: We have performed 10-fold cross validation on the dataset for all the classifiers and the accuracy values are tabulated below

Classifier	**Accuracy**
Multinomial Naïve Bayes (with Laplacian Smoothing)	0.83740
Bernouli Naïve Bayes	0.83696
Logistic Regression Classifier	0.83640
Stochastic Gradient Descent	0.82096

Support Vector Machines	0.81112
Support Vector Machines (Linear Kernel)	0.83385
Support Vector Machines (Num of Support Vectors)	0.83720

Looking at the above results, it is very clear that the accuracy of all the models are all approximately equal to 0.83. There is not much difference between the results of the classifiers used. Multinomial Naïve Bayes with Laplacian smoothing performs a slightly better job than the rest of them. The following are some of the key metrics evaluated for the Multinomial Naïve Bayes classifier.

Classification metric	**Values**
Accuracy	0.8314
F1 – Score	0.8310
Precision Score	0.8325
Recall Score	0.8296

Confusion Matrix

Actual\Predicted	**Predicted positive**	**Predicted negative**
Positive	10415	2085
Negative	2130	10370

From the results in the above table, it is evident that the Multinomial Naïve Bayes classifier with Laplacian Smoothing performs a good job, as the classification metrics are good. Looking at the confusion matrix, True Positive Rate comes to around 83 % which is very good for a binary classifier. Precision, recall and the F1-measure are approximately equal to 0.83 confirming that the fit of the model is good. The word features are significant predictors of the sentiment of the movie reviews. Following are the top few words which were very significant compared with the rest of them

Most Informative Word Features	**Odds of Success**
stinker	Negative \ Positive = 18.2 : 1.0
incoherent	Negative \ Positive = 15.7 : 1.0
turkey	Negative \ Positive = 15.7 : 1.0
unfunny	Negative \ Positive = 14.6 : 1.0
waste	Negative \ Positive = 13.6 : 1.0
flawless	Positive \ Negative = 13.0 : 1.0
pointless	Negative \ Positive = 10.8 : 1.0
redeeming	Negative \ Positive = 10.2 : 1.0
lousy	Negative \ Positive = 9.6 : 1.0
worst	Negative \ Positive = 9.5 : 1.0
poorly	Negative \ Positive = 9.4 : 1.0
superbly	Positive \ Negative = 9.3 : 1.0

wonderfully	Positive \ Negative = 8.5 : 1.0
remotely	Negative \ Positive = 8.5 : 1.0
wasting	Negative \ Positive = 8.5 : 1.0
laughable	Negative \ Positive = 8.2 : 1.0
horrid	Negative \ Positive = 8.2 : 1.0
lame	Negative \ Positive = 8.1 : 1.0
insult	Negative \ Positive = 7.8 : 1.0
blah	Negative \ Positive = 7.7 : 1.0
refreshing	Positive \ Negative = 7.4 : 1.0
atrocious	Negative \ Positive = 7.1 : 1.0
wasted	Negative \ Positive = 7.0 : 1.0
gripping	Positive \ Negative = 6.9 : 1.0
drivel	Negative \ Positive = 6.9 : 1.0
uninteresting	Negative \ Positive = 6.6 : 1.0
stupidity	Negative \ Positive = 6.6 : 1.0
breathtaking	Positive \ Negative = 6.6 : 1.0
awful	Negative \ Positive = 6.5 : 1.0
beautifully	Positive \ Negative = 6.5 : 1.0
stink	Negative \ Positive = 6.4 : 1.0
perfection	Positive \ Negative = 6.3 : 1.0
amateurish	Negative \ Positive = 6.1 : 1.0
alright	Negative \ Positive = 5.9 : 1.0
delightful	Positive \ Negative = 5.9 : 1.0
pile	Negative \ Positive = 5.8 : 1.0
nonexistent	Negative \ Positive = 5.8 : 1.0
worse	Negative \ Positive = 5.6 : 1.0
underrated	Positive \ Negative = 5.5 : 1.0
extraordinary	Positive \ Negative = 5.5 : 1.0
unconvincing	Negative \ Positive = 5.5 : 1.0
uninspired	Negative \ Positive = 5.4 : 1.0
finest	Positive \ Negative = 5.3 : 1.0
chilling	Positive \ Negative = 5.2 : 1.0
badly	Negative \ Positive = 5.2 : 1.0
appalling	Negative \ Positive = 5.2 : 1.0
ripoff	Negative \ Positive = 5.2 : 1.0
idiotic	Negative \ Positive = 5.1 : 1.0
delight	Positive \ Negative = 5.1 : 1.0
pathetic	Negative \ Positive = 5.0 : 1.0

The above table consists of the top word features and their probabilities of odds of success. It is very reaffirming to see that the words like Flawless, Superbly, Wonderfully are classified as Positive sentiment words and words like Pointless, Lousy, Waste, Poorly are classified as Negative sentiment words. Overall, word features seem to be predictive of the sentiment of the movie reviews which now can be used to classify IMDb reviews, tweets, Facebook posts etc.

V. Conclusion

From the results discussed in the earlier section, it is very clear that Multinomial Naïve Bayes classifier with Laplacian Smoothing turns out to be a very good classifier that yields the best accuracy of 83%. Upon applying various Natural Language Processing techniques on the dataset, the model is picking the best word features for text classification. Applying sentiment analysis on the Large Movie Review dataset reveals that IMDb reviews can be classified into positive or negative categories with good accuracy. These classifiers can now be used to classify movie reviews, on a variety of platforms like IMDb, twitter or Facebook posts, etc. This can, in fact, be extended to stock market predictions, classification of tweets, policy making for politics, recommendation systems and customer service.

One way to improve the results is to include bigrams (pairs of words like "not bad", "very good") in the word features. Another improvement could be to use large amounts of reviews for training the classifier. The disadvantage of Naïve Bayes classifier is that it does not involve morphological relation among the features or terms. This drawback can be overcome by Naïve Bayes variants such as Weighted Naïve Bayes, and NB with semantic probability or ontology analysis.

VI. References

1.9. Naive Bayes. (n.d.). Retrieved December 19, 2016, from http://scikit-learn.org/stable/modules/naive_bayes.html

1.4. Support Vector Machines. (n.d.). Retrieved December 19, 2016, from http://scikit-learn.org/stable/modules/svm.html

Python Programming Tutorials. (n.d.). Retrieved December 19, 2016, from https://pythonprogramming.net/sklearn-scikit-learn-nltk-tutorial

Movie on Tweets. (n.d.). Retrieved December 19, 2016, from http://chenzhe142.github.io/nu-eecs349/

R or Python on Text Mining. Retrieved December 19, 2016, from https://datawarrior.wordpress.com/2015/08/12/codienerd-1-r-or-python-on-text-mining/

Improve Your Model Performance using Cross Validation (in Python and R). Retrieved December 19, 2016, from https://www.analyticsvidhya.com/blog/2015/11/improve-model-performance-cross-validation-in-python-r/

Sentiment analysis. (n.d.). Retrieved December 19, 2016, from https://en.wikipedia.org/wiki/Sentiment_analysis