Sentiment Analysis on Movie Review

Elison Tuscano **Sai Sravan Peddi**
The University of Texas at Arlington
701 S Nedderman Dr , Arlington , TX 76019
elisonmarshal.tuscano@mavs.uta.edu saisravan.peddi@mavs.uta.edu

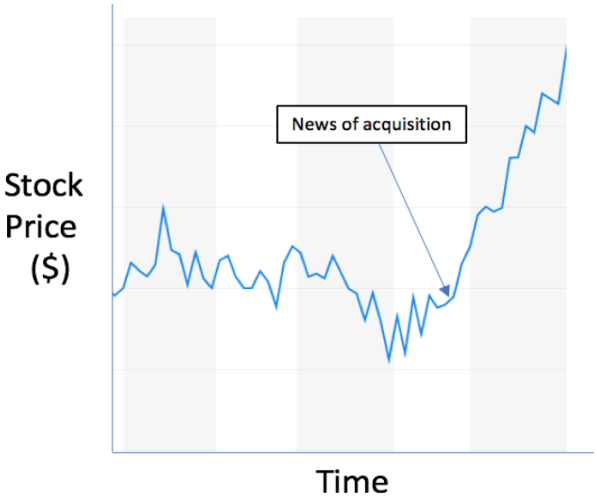
Abstract

Sentimental analysis is one of the widespread applications of the machine learning that is used for the understanding of the opinion or sentiment for a given text or review. In this project we aim to build a sentiment analysis for review data to classify whether the review was a good review or a bad review .We aim to build this project using naive-Bayes classification algorithm. Enhancement was provided by combining TF-idf and artificial neural network.

Introduction

In today's world where internet has become an important part in our life, we usually face the large amount of data being targeted at us. In this confusion of huge data that a user is being subjected sentiment analysis is used to filter the amount of data and from that decisions can be made by the user.

The task of sentimental analysis to measure the score for a review or text or a document whether its positive or negative. Sentiment analysis is used in different cases like in customer reviews or in stock trading companies where sentiment algorithms used to detect companies show positive sentiments.



Here in this project we have done sentiment analysis on movie reviews whether the given review is positive or negative using naive-Bayes classifier algorithm.

Data set Description

The data set we are going to use is obtained from the large movie review data set provided by the Stanford university. The data set consists of total 25000 movie reviews for testing and another 25000 reviews for training. Each review in the data set is given as separate text file. One folder is related to positive reviews and the other is for negative reviews.

Data preparation process:

In this preparation process we created a file which is used to take each review text file in both positive and negative folders and combine them into csv files for training the model and or testing the model. We have to create one more file to preprocess before giving it to the naive Bayes algorithm. In this file we will remove the stop words like (the, this. . . .) which do not give crucial information for recognizing whether the review is positive or negative. Then we remove the unnecessary special characters such as (!, @,. . .) from the reviews . We convert the strings of reviews into lowercase. Then we will do stemming on each review to bring the words to their base form. Finally, we convert the review column into vector.

Project Description

1. Description:

During the age of internet, the access to information has become very easy. That has created change in the traditional method of doing things. Blogs, online discussions, website for review/ rating have become a big part of our day to day life which have an influence over our decision consciously or subconsciously. Whether it may be a product to buy, carrier path to choose or movie to watch internet is the first step to gather information in order to make decision. Movie rating, recommendations are all dependent of the reviews the particular movie has on movie rating websites. In such times it is necessary to recognize whether a particular review is in favor of the movie or against it. The aim of our project is to create an application to determine the sentiment of a particular review has towards the movie. Based on

the user review the application with predict whether it is a positive review or negative review.

As the data[3] contains each review in a separate file.Each review is read and stored in review column and labelled according to the folder it belongs. The review in positive folder are labelled as 1 and reviews in negative folders are labelled as 0. Accordingly the train and test Data is read and saved. The review column will be the input for the model and the label column will be the output. In the application the user will enter the review and sentiment of the review will be predicted whether it is positive or negative review. Before providing the review as input in the system following task needs to be achieved.

- Removal of HTML tags: As the data is scrapped from various website it sometimes contains HTML tags inside the review which need to be removed.
- Removal of special Characters such as punctuation mark, hashtag,etc which does not help in predicting the sentiment which should be removed from each review.
- Tokenization :It is a method that divides the variety of document into small parts called tokens.
  Ex: I like this movie.
  After tokenization the sentence will be shown as
  "I","like","this","movie".
- Removal of Stop wards: Stop words are the commonly occurring words such as "and","the",etc. They occur in almost all of the document but does not help in predicting the sentiment as removing them creates an efficient model
- Finally reviews are ready to be converted into bag of words to provide as an input to our classification model.

a. Bag of words:

A bag of word is representation of word with its frequency count in the document. Each word is represented with its occurrence in number of documents in the training data. This list will help us to calculate probability of that word occurring and probability of that word occurring in a particular sentiment. But traditional approach to use frequency count as per mentioned in paper[1] also takes into consideration words that occur in majority of the document. As these words occur a lot only taking frequency count will give them a higher weight-age even thou these words are not a crucial part of the review. Hence we will use TF-IDF to create bag of words in our application which also take inverse document frequency into consideration to give weight-age to the word.

TF-idf (term frequency-inverse document frequency) is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. Two metrics are multiplied to achieve this: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents. With this we take the first 10000 crucial words from the data into consideration and represent them in vectorized format to give as input to our classification model.

b. Classification model

Now that we have our input cleaned and ready we are ready to provide it to our classification model. Many machine learning models are suitable for this task such as naive Bayes, Support vector machine , Random forest, neural network. We will see the implementation for naive Bayes compare it with random forest and see how neural network outperforms both of them.

Naive Bayes is a probabilistic supervised machine learning algorithm. naive Bayes uses Bayes theorem to calculate probability (review / positive) and probability (review / negative) both of these probabilities are compared. Review belongs to the sentiment with higher probability.

Bayes Theorem:
P(a / b)=[P (a / b) * P (b)] / [P (a)]
Example:
P(good/positive)= [P (good / positive) * P (positive)] / [P (good)]
where,
P(good/positive) = Number positive documents containing the word "good" / Number of all positive documents .
P(positive)=Number of all positive documents / Total Number of Documents
P(good) = number of documents containing 'good' / number of all documents

For every word in the review probability of that word with the sentiment shall be removed and multiplied together in order to get the actual answer.
P(a1,a2,a3,a4.. / b)
Example: "It was a good movie"
After Cleaning and removing Stop-words:"It good movie"

Probability of the review for positive will be calculated:
P(It/positive)*P(good/positive)*P(movie/positive)

Similarly Probability of the review for negative will be calculated:
P(It/negative)*P(good/negative)*P(movie/negative)

After calculating the probability for both the sentiments, according to the higher probability value the review sentiment is predicted. Naive Bayes created from scratch gave an accuracy of 52 percent whereas naive Bayes with sklearn library gave an accuracy of 78 percent. Smoothing is also implemented in naive Bayes from scratch.

Comparison with **Random Forest:** Random forest is a supervised learning algorithm used for classification. Random forest is an ensemble learning method for classification, regression and other task by constructing multiple decision tree. As it constructs multiple decision tree it provides more chances of error elimination. When model was created with random forest the accuracy was found to be 79 percent which is a minor improvement.

Enhancement is provided with **Artificial neural network**. ANN is a part of deep learning where each node is represented as a neuron. It has an input layer, hidden layer and output layer. Each input will be represented by a neuron in the input layer. We have 10000 words vector which will be provided to our 10000 input neurons. The output required is to classify whether it is positive or negative hence one output neuron will be enough to represent in binary format whether the review is positive "1" or negative "0". Below is the structure of the neural network.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_4 (Dense)              (None, 256)               2560256
_________________________________________________________________
dense_5 (Dense)              (None, 64)                16448
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 65
=================================================================
Total params: 2,576,769
Trainable params: 2,576,769
Non-trainable params: 0
```

It has total of 2.5 million params. Activation function used is "relu" for our hidden layers and activation function used for output layer is "sigmoid " as we want the answer in binary format where 1 presents positive and 0 represents negative. Adam a type of stochastic gradient descent is used as the optimizer. The neural network was created using tensorflow and keras. The accuracy on ANN was achieved to be 85 percent. As this is the best accuracy so far we will save this model and use this for our web user interface.

c. Web User Interface

Once the classification model is created and saved it can be used for creating a web user interface where user can enter the review. On pressing the submit button the review written by the user in the text area will be cleaned as used as an input for our saved model. After providing the input the saved model will predict the sentiment which will we displayed back to the web page.

The web user interface was created with help of flask which is a python library used to integrate python with web-pages. Once the application was ready it was hosted to Heroku at [4]. **Heroku** a cloud Platform as a service provider so that the application can be accessed from anywhere. Below is the example of the interface with positive review.

2. Reference Description:

Kavya Suppala and Narasinga Rao [1] did sentiment analysis using naive Bayes on twitter data-set. They have mentioned in detail how to tokenize every word in the review and represent them as a bag of words with their frequency count. Here naive Bayes approach is used which is a probabilistic supervised machine learning algorithm. Probability is calculated on both positive and negative words. Words with highest probability are selected and the sentiment of the review is predicted accordingly.

Tirath Prasad Sahu and Sanjeev Ahuja [2] did Sentiment Analysis on Movie review. Comparison of different classification model is done and accuracy is checked with 10 fold cross validation.

3. Difference in Approach between Proposed and Existing System:

The original approach using naive Bayes does a good job at identifying the sentiment of the review but as bag of words are created with just frequency count the model also takes into consideration words that occur in majority of the document which add no crucial value to the review. Hence TF-IDF was used to create bag of words which takes frequency count and inverse document frequency so that only crucial words are used for predicting the sentiment of the review.

4. Difference in Accuracy between Proposed and Existing System:

Traditional approach by using naive Bayes gave an accuracy of 78 percent on 10 fold cross validation , It was compared with random forest which gave an accuracy of 79 percent. Classification model created with artificial neural network gave an accuracy of 85 percent which outperforms the rest of the classification models.

5. List of contributions:

1. Implementation of naive Bayes from scratch.
2. Using TF-IDF to create bag of words rather than using normal frequency count.

3. Comparing naive Bayes from scratch , naive Bayes using sklearn and Random forest.
4. Enhancing the classification model with artificial neural network.
5. Implementing Sentiment Analysis on movie review using Web User Interface

Analysis

1. What did I do well:

Generally for sentiment analysis different classification model such as naive Bayes, support vector machine, Random forest are used to identify the sentiment of the text. After trying and experiment with different classification models the decision to experiment with Artificial neural network paid off as it helped to increase the accuracy.

2. What Could i have Done Better:

Currently the accuracy on testing data is not as good as the accuracy on training data which indicates there might be over-fitting happening in the neural network. More experiments could have been conducted to reduce over-fitting change the structure of layers or adding few dropout layers.

3. What is left for Future:

As most of the review in the data set were huge the classification model created does a good job identifying the sentiment of large reviews. It sometimes fails to identify small reviews. This problem can be solved by using another data set for consisting majority of the small reviews and combining the two data set to create a improved model.

Conclusion:

Normally sentiment analysis is done with naive Bayes as mentioned in the base paper [1]. The original approach to classify the sentiment using naive Bayes gave an accuracy of 78 percent. Using TF-IDF rather than normal frequency count helped to identify crucial words. Providing the bag of words from TF-IDF to the artificial neural network increased the accuracy to 85 percent which was an improvement to the traditional approach.

References

[1]. Kavya Suppala, Narasinga Rao 2019 ,Sentiment Analysis using naive Bayes Classifier ,International Journal of Innovative Technology and Exploring Engineering (IJITEE).

[2]. Tirath Prasad Sahu, Sanjeev Ahuja 2016 ,Sentiment Analysis on movie review: A study of feature selection and Classification algorithm, IEEE.

[3]. Dataset- http://ai.stanford.edu/ amaas/data/sentiment/

[4]. Web Demo- http://reviewdetector.herokuapp.com