SYNOPSIS

Sentiment Analysis of IMDB Movie Review Data

Using Big Data

By

Shivanee Panchal

Roll No. **DBSOM19BD04029**

M.Sc. (Big data and Business Analytics)

Under the guidance of

Prof. Srinatha D.K.

Batch 2019 -2021

ABSTRACT

Sentiment analysis also term as refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. In recent years, Opinion mining is a hotspot in the field of natural language processing, and it is also a challenging problem Sentiment analysis is widely applied to reviews and social media for a variety of applications, ranging from marketing to customer service. Movie reviews are an important way to gauge the performance of a movie. The objective of this paper is to extract features from the product reviews and classify reviews into positive, negative. . In this project we aim to use Sentiment Analysis on a set of movie reviews which is given by reviewers and then try to understand what the overall reaction to the movie was according to them, i.e. if they liked the movie or they hated it. We aim to use the relationships of the words in the review to predict the overall polarity of the review.

Keywords: Movie Review Mining, Pre-processing, Sentiment Analysis, Stemming

1. INTRODUCTION

Movie reviews are an important way to gauge the performance of a movie. While providing a numerical/stars rating to a movie tells us about the success or failure of a movie quantitatively, a collection of movie reviews is what gives us a deeper qualitative insight on different aspects of the movie. A textual movie review tells us about the the strong and weak points of the movie and deeper analysis of a movie review can tell us if the movie in general meets the expectations of the reviewer. Sentiment Analysis is a major subject in machine learning which aims to extract subjective information from the textual reviews. The field of sentiment of analysis is closely tied to natural language processing and text mining. It can be used to determine the attitude of the reviewer with respect to various topics or the overall polarity of review. Using sentiment analysis, we can find the state of mind of the reviewer while providing the review and understand if the person was “happy”, “sad”, “angry” and so on. In this project we aim to use Sentiment Analysis on a set of movie reviews given by reviewers and try to understand what their overall reaction to the movie was, i.e. if they liked the movie or they hated it. We aim to utilize the relationships of the words in the review to predict the overall polarity of the review.

2. OBJECTIVE

Internet is rich source of reviews on reviews, ecommerce products or online services. Customer always prefers to read reviews before going out for dinner or watching movie or paying money to service provider. But it is hardly possible to read all reviews in today's fast life. Also every review may provide new information of product or feature of product. So there is probability of missing any important review given by customer.

We need to identify polarity of review i.e. whether it is positive, negative or neutral. Sentiment analysis will assist us to find out polarity of reviews. Due to some limitations here I am taking secondary data from IMDB movie review website. All the customers are not good in technologies so if we visualize all the reviews, it will make easier decision making process for customer. Customer will be able to see all reviews at a glance and he/she will take decision faster.

Thus, our main objectives are:

a. Dealing with neutral reviews: Output must consider reviews (Positive/Negative) as they make impact on decision making.
b. Improved Efficiency: Many reviews are given for single movie. Using machine learning algorithms we can improve efficiency of sentiment analysis.
c. Sentiment Analysis: To determine attitude of mass people towards particular movie/product or service.
d. User oriented Data Visualization: Customers are mainly nontechnical persons. So we aim to visualize results in user readable format.

3. SYSTEM REQUIREMENTS

- Hardware Requirements:

⇒ Core i5/i7 processor
⇒ At least 8 GB RAM
⇒ At least 60 GB of Usable Hard Disk Space

- Software Requirements:

⇒ Python 3.x
⇒ Anaconda Distribution
⇒ NLTK Toolkit
⇒ UNIX/LINUX/MAC Operating System.

- **Dataset: IMDB review dataset**:

⇒ I am using a publicly available dataset consisting of 50,000 reviews from IMDB, allowing no more than 30 reviews per movie. The constructed dataset contains an equal number of positive and negative reviews, so random guessing yields 50% accuracy.

⇒ You can download the dataset from the following link:

⇒ https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews
⇒ The dataset has three columns. The first column specifies a sequential number of the record. The second column has a text which is the review and the third column is a class denoting the sentiment of the reviewer. The class has value 1 if the sentiment is positive and 0 if the sentiment is negative.
⇒ 'Here Objective is to build a custom sentiment analyser that can classify sentiment (positive or negative) of reviewers out of their movie reviews.'

4. ARCHITECTURE

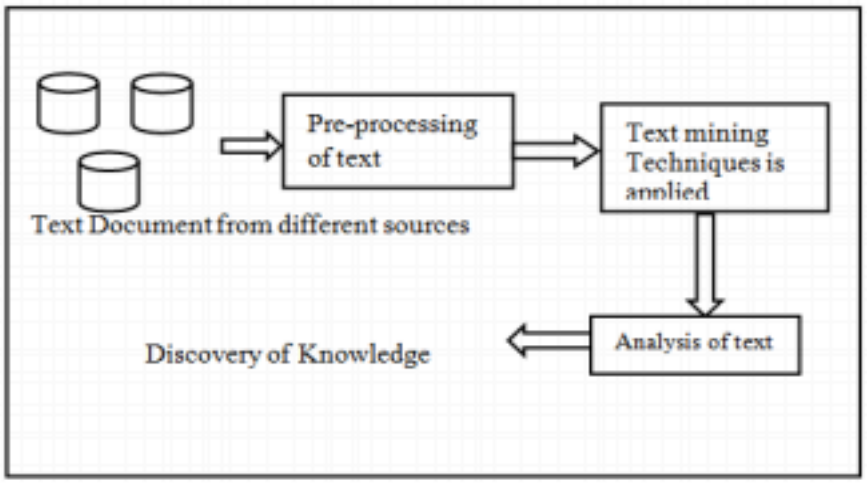


Figure 1. Text Mining Process

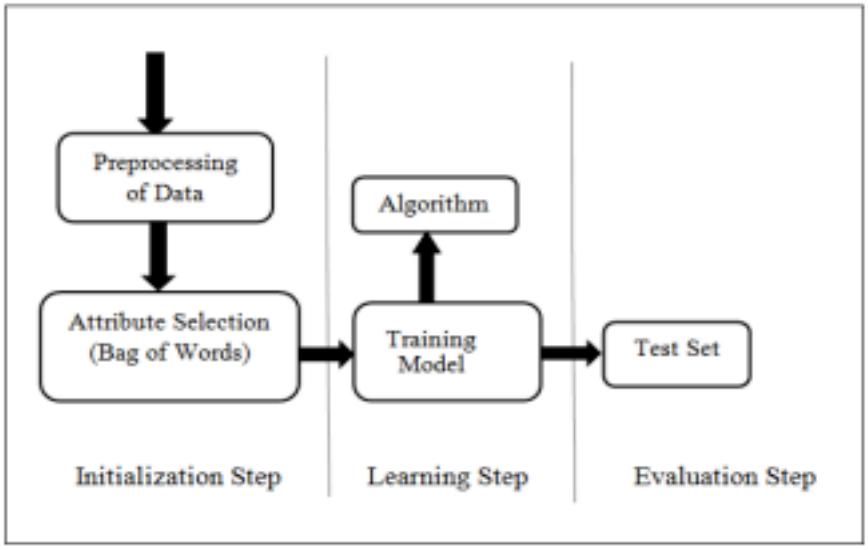


Fig. 2. Steps for training a classifier for sentiment analysis

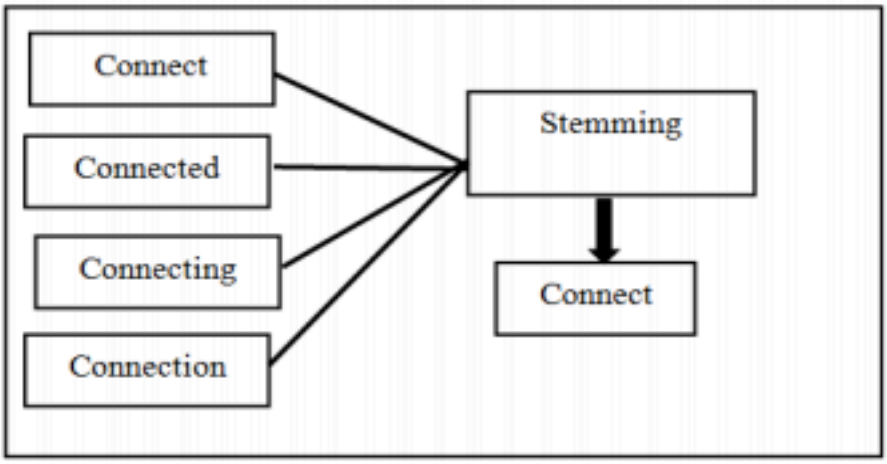
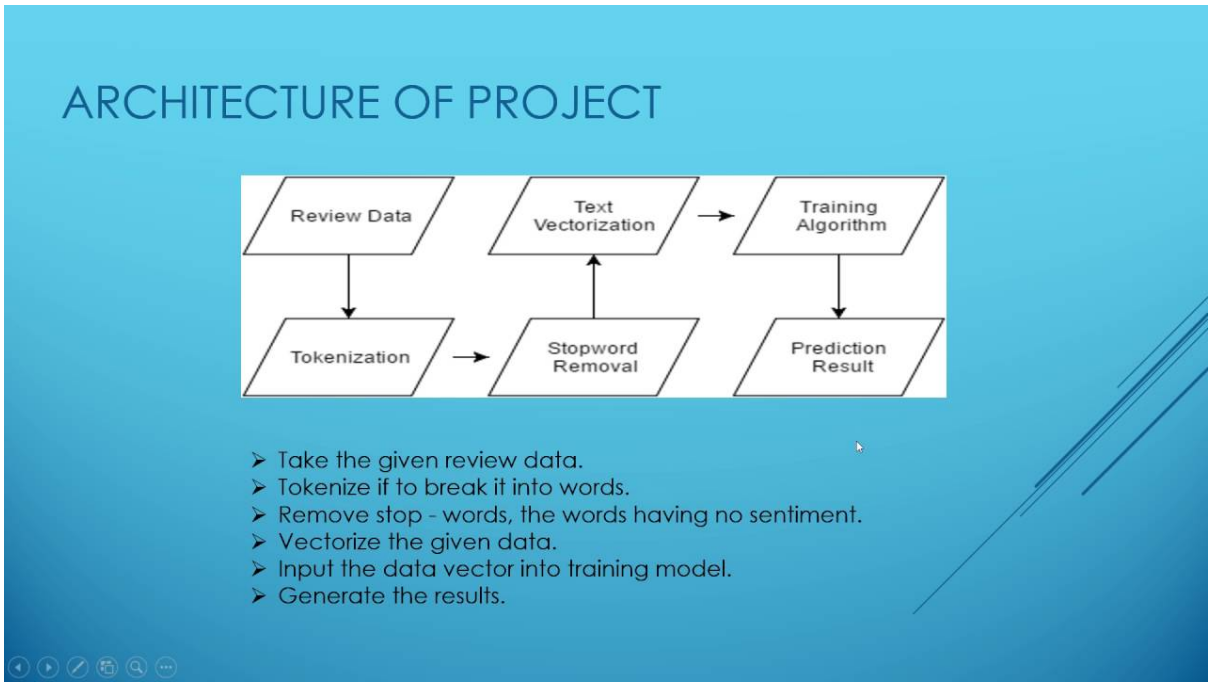


Fig 3 – Stemming Process



5. ALGORITHM AND FORMULATION

- **Step 1:** Download and Combine Movie Reviews

If you haven't yet, go to IMDb Reviews and click on "Large Movie Review Dataset v1.0". Once that is complete you'll have a file called `aclImdb_v1.tar.gz` in your downloads folder.

- **Step 2**: Read into Python

- **Step 3:** Clean and Pre-process

The raw text is pretty messy for these reviews so before we can do any analytics we need to clean things up.

- **Step 4:** Vectorization

In order for this data to make sense to our machine learning algorithm we'll need to convert each review to a numeric representation, which we call *vectorization*.

1. Sentiment sentence extraction & pos tagging:

Tokenization of reviews after removal of STOP words which mean nothing related to sentiment is the basic requirement for POS tagging. After proper removal of STOP words like "am, is, are, the, but" and so on the remaining sentences are converted in tokens. These tokens take part in POS tagging

In natural language processing, part-of-speech (POS) taggers have been developed to classify words based on their parts of speech. For sentiment analysis, a POS tagger is very useful because of the following two reasons: 1) Words like nouns and pronouns usually do not contain any sentiment. It is able to filter out such words with the help of a POS tagger; 2) A

POS tagger can also be used to distinguish words that can be used in different parts of speech.

2. Process of transforming data into numeric vectors:

1. The first step is to select the data which needs to be modelled for training the analyser.
2. stop words that are present in almost all documents such as a, an, the, is etc, must be removed from training data. Since they occur in almost every sentence they won't be useful at all for extracting sentiment information from the text
3. Further Data Pre-processing is done using regex by removing special characters, digits from our documents.
4. We use term-frequency to represent each term in our vector space. We determine term-frequency which is nothing more than a measure of how many times the terms present in our vocabulary. We can specify many parameters in Tf-Idf such as Max_df (maximum Document frequency), Lowercase(for converting all characters to lowercase), max_features (specifies No of terms used in building vocabulary).
5. we can define the term-frequency as a counting function:

$$\mathrm{tf}(t, d) = \sum_{x \in d} \mathrm{fr}(x, t)$$

- **Step 5:** Build Classifier

About Machine Learning Algorithm Used:

The machine Learning algorithm used is: Stochastic Gradient Descent Classifier from *Scikit-Learn*

Why this Particular Algorithm ? why not other commonly used algorithms such Naive-bayes, Decision tree or Logistic Regression.?

The main reason for using SGD Classifier for training our data model is that our training data is very large hence it is sparse, the classifiers in this module easily scale to problems with more than 10^5 training examples and more than 10^5 features..

The estimator implements regularized linear models with stochastic gradient descent (SGD) learning: the gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule (aka learning rate).

This results in a better accuracy as the data backpropagates to reduce error.

- **Step 6:** Train Final Model

Now we should train a model using the entire training set and evaluate our accuracy on the 25k test reviews.

- **Step 7:** Testing our Data Model:

We use scikit-learn library and calculate accuracy precision recall and f1 results for the testing data which is the other 25000 reviews. This helps us in determining how good our model has trained for Sentimental Analysis.

Also we print the confusion Matrix to determine True Positive, False Positive, True Negative and False Negative.

We can Visualize our Confusion Matrix using Matplotlib Library of Python for better Analytics of our results.