Sentiment Analysis of IMDb movie reviews

Paritosh Gaiwak
Pragam Gandhi
Sanjana Kacholia
Tushar Pahuja

1 Background and Introduction

1.1 Natural Language Processing

Natural language processing (NLP) is a branch of machine learning which uses the concepts of computer science, artificial intelligence and computational linguistics to enable the computers to understand and work on Natural language i.e. the language used by humans to communicate. NLP models derive structure from the unstructured natural language(s) to create applications such as text summarization. For example, chat bots extensively use NLP models to communicate with humans and understand commands given to them.

NLP is challenging because the human language does not follow a particular format or structure. Understanding human language not only involves understanding the words but also the understanding of the concepts and how words are linked together to create meaning. A word can have several meanings which are easy for humans to understand based on the context but challenging for a machine to understand and comprehend. This is one of the main challenges of NLP. Another remarkable thing about human language is that it includes symbols and gestures which are associated with speech or text. So, the main challenges include conserving structure from ambiguity, synonymy and understanding intention of speech.

Some of the important terms in NLP include parsing, stemming, tokenization, Corpus or Corpora, a bag of words, stop words and vectorization. For this project, we have implemented stemming, tokenization, vectorization and stop word removal on the dataset.

NLP has real-world applications like automatic text summarization, tag generation, sentiment analysis, sarcasm detection, topic extraction, named entity recognition and relationship extraction, text mining, machine translation and automated question answering (bots on websites).

1.2 Sentiment Analysis

It's estimated that 80% of the world's data is unstructured and unorganized. Most of this comes from text data like emails, support tickets, chats, social media, surveys, articles and documents. These texts are usually difficult, time-consuming and expensive to analyze, understand and sort through.

Sentiment analysis is the automated process of understanding an opinion about a given subject from written or spoken language[1]. It is a popular sub-field of Natural language processing which helps in building systems to identify and extract opinions from the text. With the help of sentiment analysis, unstructured information (in the context of machine learning) in the form of sentences can be classified into a structured format which is beneficial for commercial applications like marketing analysis, public relations, product reviews, net promoter scoring, product feedback and customer service.

Sentiment analysis systems allow companies to extract information from unstructured text by automating business processes, getting actionable insights and saving hours of manual data processing, in other words, by making teams more efficient.

These sentimental analysis systems are also capable of extracting attributes of the expressions, like:

- Polarity: if the speaker expresses a positive or a negative opinion
- Subject: the thing that is being talked about
- Opinion holder: the person or entity expressing the opinion

We apply sentimental analysis on opinions to classify and extract useful information from them. Text can be broadly categorized into two main types:

- Facts: objective information about something
- Opinions: subjective expressions that describe people's sentiments, appraisals, and feelings toward a subject or topic

There are many methods and algorithms to implement sentiment analysis systems, which can be classified as follows:

- Rule-based systems that perform sentiment analysis based on a set of manually crafted rules
- Automatic systems that rely on machine learning techniques to learn from data
- Hybrid systems that combine both rule-based and automatic approaches

Sentiment analysis can be modeled as a classification problem where two sub-problems can be solved:

- Classifying a sentence as subjective or objective, known as subjectivity classification.
- Classifying a sentence as expressing a positive, negative or neutral opinion, known as polarity classification.

Sentiment analysis can be applied at different levels of scope:

- Document-level sentiment analysis - obtains the sentiment of a complete document or paragraph
- Sentence level sentiment analysis - obtains the sentiment of a single sentence
- Sub-sentence level sentiment analysis - obtains the sentiment of sub-expressions within a sentence

Most of the work in sentiment analysis in recent years has been around developing more accurate sentiment classifiers by dealing with some of the main challenges and limitations in the field like,

- Determining the subjectivity and tone of an argument
- Determining the context and polarity of a sentence accurately
- Detecting sarcasm/appreciation
- Comparison of different texts of comparable sizes

2 Method

2.1 Project description

Motivated by the importance of NLP and its applications, we decided to perform sentiment analysis on IMDb movie reviews data set by finding the sentiment of a movie review. The data set was taken from the Stanford online repository. The model was trained and four different algorithms have been applied to the data set till now.

2.2 Preprocessing

Exploratory data analysis was performed on the data set to develop a better understanding of the nature and structure of the data set. The data set consisted of two attributes, namely review and its

label. As part of EDA, the number of samples - positive and negative- were found to be 25000 each. The distribution of review length and frequency distribution of words were also plotted. Initially, we removed the stop words from the reviews using the stopword corpus present in the nltk.corpus module in Python, as in most cases the stop words are irrelevant and do not affect the contextual meaning of the review. Then, we decoded the data using UTF-8 to maintain uniformity in the data.After this, the HTML tags were removed and the text was converted to lower case. After this, stemming was performed in order to transform the words to their root words. After stemming, the frequency distribution of the review was plotted to understand the words being used.

2.3 Building models

The data was split into training and testing sets using random selection in the ratio of 70:30. The same data was then used for all the 4 methods. Following methods were applied to the data set for training and then testing to find the sentiment of the review. While using the methods described below we had to experiment with some parameters like max_df and min_df where max_df is used to ignore terms that have a document frequency higher than the threshold and min_df is used to ignore terms that have a document frequency lower than the threshold and found the best accuracy was at max_df = 0.7 and min_df = 3.

- *Logistic regression*: Logistic regression is a technique which is used when the dependent variable is binary in nature. This technique is generally known to produce decent results. Logistic regression can be used to describe data and to explain the relationship between one dependent binary variable and one or more independent variables. In our case, presence/absence of various words in the review are considered as independent variables while the label is a binary dependent variable which can be either positive(1) or negative(0). Due to binary nature of the independent variable, we performed logistic regression on the data.
- *Decision tree*: It is a predictive modeling approaches in which a tree is used for classification. The leaves represent class labels and branches represent conjunctions of features that lead to those class labels. As our target variable can take either 0 or 1, that is, our review can either be positive or negative, decision tree algorithm has been used for this classification.
- *Random forest*: It is an ensemble learning method which can be used for classification. It operates by constructing a number of decision trees from the training set and then giving the final result which is the mode of the classes/results of individual trees. Random forests mitigate the problem caused by decision trees' over fitting. We use 100 decision trees as a part of the random forest for the problem after experimenting with different values.
- *Multinomial Naive Bayes*: Naive Bayes uses probability theory and Bayes' Theorem to predict the label of a text[2]. This algorithm is probabilistic in nature, that is, the probability of each label for a given text is calculated, and then the output is the label with the highest probability. In our case, this concept has been used to find the most probable label, which may be positive or negative based on the text, i.e.a movie review.

3 Experimentation setup

The experiments are conducted in Python 3 and for organized visualization, Jupyter notebook is used. The following packages/modules have been used for this project: NLTK (natural language toolkit), re(regular expressions), sklearn, numpy, and matplotlib.

To run the Jupyter notebook, navigate to the location of the .ipynb file on the terminal and type "jupyter notebook". After this step the contents of the folder can be seen on the default browser. Then the required file can be selected. Make sure that the data set is in the same location as the file, or ensure that the location is correct while loading the data set. The use of Jupyter Notebook makes it easier and convenient as we can see the results and visualize the plots there itself.

4 Results

Logistic Regression

6627(TP)	930(FN)
784(FP)	6659(TN)

Decision Tree

5454(TP)	2103(FN)
2126(FP)	5317(TN)

Random Forest

6474(TP)	1083(FN)
784(FP)	6215(TN)

Multinomial Naive Bayes

6413(TP)	1144(FN)
1032(FP)	6411(TN)

Table of comparison for all the algorithms

	Logistic regression	Multinomial Naïve bayes	Decision Tree	Random forest
Precision	0.8769	0.8486	0.7217	0.8566
Recall	0.8942	0.8613	0.7195	0.8405
Accuracy	0.8857	0.8549	0.7180	0.8459
F1	0.8854	0.8549	0.7206	0.8485

5 Conclusion

After calculating the accuracy for the algorithms above, we obtain the following order for accuracy. Accuracy decreases as we go from left to right.

Logistic regression > MultinomialNB > Random forest > Decision tree

6 References

1. Describes both symbolic and machine learning techniques for understanding sentiments from the text [link]
2. Baselines and bigrams: Simple, good sentiment and topic classification [link]