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 - generic problem description

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 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.

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

1.3 Brief problem statement

Motivated by the importance of sentiment analysis and its applications, we decided to perform sentiment analysis of IMDb movie reviews dataset by classifying a movie review as positive or negative.

1.4 Literature survey

Seeing a tremendous increase in the number of papers focusing on sentiment analysis during the recent years, it would be correct to say that this topic has grabbed a lot of attention among researchers. According to the research by Mika V. Mäntylä Daniel Graziotin [2], nearly 7,000 papers of this topic have been published. 99% of the papers have appeared after 2004 making sentiment analysis one of the fastest growing research areas. The sudden outburst of modern sentiment analysis happened only in mid-2000's which, focused on product reviews available on the internet, e.g., [3]. Additionally, research overlapping sentiment analysis and natural language processing has addressed many problems that contribute to the applicability of sentiment analysis such as irony detection [4] and multi-lingual support [5]. Furthermore, with respect to emotions, efforts are being made and thus advancing from simple polarity detection to more complex nuances of emotions and differentiating negative emotions such as anger and grief [6]. Referring to the "Sentiment analysis algorithms and applications: A survey" [7], which talks about the overall process of sentiment analysis and various

machine learning approaches. The paper describes algorithms including probabilistic classifiers, Naive Bayes classifier, maximum entropy classifier, SVM, neural networks, decision tree classifier and rule-based classifiers. Apart from these, the paper also discusses about other approaches like semantic approach, lexicon-based approach, etc. This paper gave us a good idea about the various approaches suitable for performing sentiment analysis and helped us give a quick start to the project.

2 Methods

2.1 Approach - 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 7 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.

- *Logistic regression*: Logistic regression is a technique used when the dependent variable is binary in nature. This technique generally produces 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 this project, 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 approach 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 the target variable can take either 0 (negative) or 1 (positive), decision tree algorithm has been used for this project.
- *Random forest*: It is an ensemble learning method 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 [8]. 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 this project, this concept has been used to find the most probable label, which may be positive or negative.
- *Artificial neural networks*: They are computing systems inspired by the biological neural networks. The neural network is a framework for many different machine learning algorithms to work together and process complex data inputs. ANN systems learn to perform tasks by considering examples, generally without being programmed with any task-specific rules.
- *Ensemble learning* Ensemble learning is a machine learning paradigm where multiple learners are trained to solve the same problem. In contrast to ordinary machine learning approaches which try to learn one hypothesis from training data, ensemble methods try to construct a set of hypotheses and combine them to use.
- *Support vector classifier* It is a discriminative classifier defined by a separating hyper-plane. In simple terms, given a labeled training data, this supervised learning algorithm outputs an optimal hyper-plane which categorizes new examples. In two dimensional space, this hyper-plane is a line dividing a plane into two parts wherein each class lay on either side. (We used linear SVC).

2.2 Rationale

In order to classify reviews as positive or negative, we used the simplest and the most interpret-able algorithm, logistic regression. The results were then compared with other classification techniques - support vector classifier and decision trees. Since the decision trees performed poorly, we wanted to

analyze if the results could be improved using an ensemble of those trees i.e. using Random Forest as a classifier. In order to get the best of the results, we also implemented Ensemble learning by assigning different weights to different algorithms according to their performance. Also, we went ahead and tried if a basic neural network could perform better than the algorithms defined. If the data had no linear relationship or differentiating tokens, neural network performance could have been better.

3 Plan and Experimentation

3.1 Data set

The data set was taken from the Stanford online repository. It comprises of 50000 reviews and corresponding labels stating if the review is positive or negative. It contains 25000 positive samples and 25000 negative samples. Data preprocessing is a crucial step in data analysis. Firstly, the textual data was converted to lowercase followed by the removal of special symbols and HTML tags. Then, the words like 'the', 'and' etc. (stop words) which do not contribute significantly to the meaning of review were removed using stop words package from the NLTK library. The reviews were then treated in two ways:

- Stemming - The process of reducing a word to its stem that affixes to suffixes and prefixes. This was done using PorterStemmer.
- Lemmatization - The process of grouping together different forms of the same word. Lemmatization allows end users to query any version of a base word and get relevant results. This was done using WordNetLemmatizer.

Lemmatization performed better as the words were reduced to their 'meaningful' root forms.

3.2 Hypotheses

- Which classifier works best for classifying this data set?
- Which among linear and non-linear classifiers performs better?
- Can we improve the accuracy of classification using an ensemble of different classification algorithms?
- How did different classifiers fare for different metrics?

3.3 Experiment Design

The experiments are conducted in Python 3 and for organized visualization, Jupyter notebook is used. 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.

Exploratory data analysis was performed on the data set to develop a better understanding of the nature of the data. The distribution of the length of reviews and the frequency of different words in the corpus were plotted. The reviews were then converted into vectors using CountVectorizer. The results of PCA of bag-of-words was also plotted. As the components were overlapping, tf-idf model was applied to it. TF-IDF stands for term frequency–inverse document frequency and is a numerical statistic that is intended to provide information on the importance of a word in a document in a collection or corpus.

The tf-idf array, formed using minimum-gram of 1 and maximum n-gram of 2, was used for building the models. The maximum features were limited to 10000 and the terms that occur in more than 70% of the reviews were ignored.

The data was split into training and testing parts using random selection in the ratio of 70:30. We experimented with various parameters of different algorithms. For logistic regression, the better results were obtained when liblinear solver was used. For support vector classification, linear classification gave better results. After experimenting with different values, we found that random forest gave higher accuracy with 100 estimators.

Ensemble learning was applied by assigning different weights to different algorithms and the performance was better with higher weights assigned to logistic regression and support vector classifier.

Neural network was implemented using dropout_rate of 0.4 and 'relu' function in hidden layer and 'sigmoid' function for the output layer. The validation set for the neural network was 10% of the training data and the optimizer used for the neural network was 'adam', with a learning rate of 0.3.

A more sophisticated neural network could be designed, but since it would effectively increase the time and space without providing much improvement in accuracy.

These algorithms were used to create models and predictions were made based on the classifiers. Once the model was built, predictions on the test data were made. Confusion matrix for each model predictions was created and the following metrics were then calculated.

- Accuracy: Accuracy refers to the closeness of a measured value to a standard or known value.
- Precision: Precision refers to how close estimates from different samples are to each other.
- Recall: Also known as sensitivity, it is the fraction of relevant instances that have been retrieved over the total amount of relevant instances.
- F1 score: F1 score conveys the balance between the precision and the recall and is calculated using the harmonic mean of the two.

4 Results

Confusion Matrices

Logistic Regression

6646(TP)	911(FN)
736(FP)	6707(TN)

Decision Tree

5460(TP)	2097(FN)
2137(FP)	5306(TN)

Random Forest

6444(TP)	1113(FN)
1143(FP)	6300(TN)

Multinomial Naive Bayes

6452(TP)	1105(FN)
913(FP)	6530(TN)

Neural Network

6424(TP)	1133(FN)
844(FP)	6599(TN)

Support Vector Classifier

6682(TP)	875(FN)
830(FP)	6613(TN)

Ensemble Learning

6607(TP)	950(FN)
674(FP)	6769(TN)

Evaluation Metric Comparison table

Model	Accuracy	F1	Precision	Recall
Logistic Regression	0.89	0.89	0.87	0.9
Support Vector Classifier	0.89	0.88	0.88	0.89
Ensemble Learning	0.89	0.89	0.87	0.9
Neural Networks	0.86	0.86	0.85	0.88
Multinomial NB	0.86	0.86	0.85	0.87
Random Forest	0.84	0.85	0.85	0.84
Decision Tree	0.71	0.72	0.72	0.71

After obtaining the accuracy, F1 score, precision and recall for all the algorithms used, we were able to identify that Logistic Regression gave the best overall results.This is primarily due to the binary nature of the output.

The order of performance of algorithms based on accuracy was: Logistic Regression > Ensemble Learning > Support Vector Classifier > Neural Networks > Multinomial Naive Bayes > Random Forest > Decision Tree.

For achieving optimal results with Ensemble Learning, Logistic Regression and Support Vector Classifier were given higher weights as compared to Multinomial Naive Bayes and Random Forest with 0.4 (Logistic), 0.4 (SVC), 0.1 (MNB) and 0.1 (RF).

Decision trees performed the worst due to the lack of properly differentiable tokens, leading to rules which don't give very clear classification results.

If there is a linear relationship in the feature space and output, linear classifiers perform better than non-linear classifiers, as can be verified by the results.

5 Conclusion

The degree of complexity of sentiment analysis depends on the number of possible output class labels. For instance, we found data sets with 5 output labels: positive, slightly positive, neutral, slightly negative and negative. In such a case, an approach like logistic regression cannot be used since the output is not binary.

The key takeaway from the project is that when the output has a linear relationship with feature space, a simple and fast algorithm like Logistic Regression outperforms other techniques. Also, since decision trees did not perform well, we can say similar tokens were present in reviews of both positive and negative classes.

We also expected ensemble learning methods to cause a significant increase in the accuracy as compared to individual classifiers. However, the accuracy obtained from ensemble of the classifiers did not improve significantly as compared to individual classifiers.

A possible future direction is to determine the tone of the review. For example, a negative review can have a straight forward negative tone, a sarcastic tone or a mild humorous tone. This requires more sophisticated algorithms.

The latest developments in sentiment analysis include using various novel deep learning architectures like auto encoders. Our sentiment analysis problem is a relatively simple version. New approaches are being developed in the field of artificial intelligence to acquire the sentiment of a speaker in real time. Such approaches require the application of complex deep learning architectures like Convolutional neural networks (to read the eye movements and expressions) and a multitude of other algorithms to understand the sentiment of the speaker.

GitHub link: https://github.ncsu.edu/pmgandh2/imdb_movie_reviews_sentiment_analysis

6 References

1. M S Neethu, "Sentiment analysis in twitter using machine learning techniques", https://ieeexplore.ieee.org/document/6726818
2. The Evolution of Sentiment Analysis - A Review of Research Topics, Venues, and Top Cited Papers, https://arxiv.org/ftp/arxiv/papers/1612/1612.01556.pdf
3. K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in Proceedings of the 12th international conference on World Wide Web, 2003, pp. 519–528.
4. A. Reyes and P. Rosso, "On the difficulty of automatically detecting irony: beyond a simple case of negation," Knowledge and Information Systems, vol. 40, no. 3, pp. 595–614, 2014
5. A. Hogenboom, B. Heerschop, F. Frasincar, U. Kaymak, and F. de Jong, "Multi-lingual support for lexicon-based sentiment analysis guided by semantics," Decision support systems, vol. 62, pp. 43–53, 2014.
6. E. Cambria, P. Gastaldo, F. Bisio, and R. Zunino, "An ELM-based model for affective analogical reasoning," Neurocomputing, vol. 149, pp. 443–455, 2015.
7. Walaa Medhat, Ahmed Hassan, Hoda Korashy, "Sentiment analysis algorithms and applications: A survey", https://www.sciencedirect.com/science/article/pii/S2090447914000550
8. Sida Wang, "Baselines and bigrams: Simple, good sentiment and topic classification", https://dl.acm.org/citation.cfm?id=2390688