Sentiment Analysis on IMDB Movie Review

IME673A: Applied Machine Learning
Indian Institute of Technology Kanpur
Supervised by: Dr. Veena Bansal

21114014 Pushkar Awasthi
21114018 Reeshabh Anand

Introduction:

The act of computationally recognizing and categorizing opinions contained in a piece of text, especially in order to discern whether the writer has a good, negative, or neutral attitude toward a given topic, product , etc. Sentiment Analysis is a technique for analyzing a piece of text to determine the sentiment contained within it. It accomplishes this by combining machine learning and natural language processing (NLP).So Movie Review analysis is type of customer feedback analysis here, we'll walk through the steps of creating a model that can perform sentiment analysis on a big movie database. The information was gathered from the Internet Movie Database (IMDB).

Problem Statement

The main goal is to estimate the sentiment many movie reviews from the Internet Movie Database (IMDB).

Dataset

Based on the content of the reviews, this dataset contains 50,000 movie reviews that have been pre-labeled with "good" and "negative" sentiment class labels.

Now we will walk you down with the steps we have performed in our dataset:

Setting Up Libraries

Importing necessary libraries. We'll be using a number of Python tools and frameworks dedicated to text analytics, .natural language processing, and machine learning such as numpy, pandas, seaborn, matplotlib, sklearn (CountVectorizer, Tfidf Vectorizer) nltk (stopwords,PorterStemmer) etc.

Text Preprocessing

Cleaning, pre-processing, and normalising text to bring text components like phrases and words to some standard format is one of the key steps before going into the process of feature engineering and modelling.

Text normalization

Words are tokenized. To separate a statement into words, we utilise the word tokenize () method.

Removing html strips and noise text

Here in data head we can see some html code so first we need to clean that html strips. Also removing some noisy texts along with square brackets

Removing special characters

Because we're working with English-language evaluations in our dataset, we need to make sure that any special characters must be deleted.

Text stemming

Stemming is a technique for eliminating affixes from words in order to retrieve the base form. It's the same as pruning a tree's branches down to the trunk. The stem of the terms eating, eats, and eaten, for example, is eat.

Removing stop words and normalization

Stop words are words that have little or no meaning, especially when synthesizing meaningful aspects from the text. Stop words are words that are filtered out of natural language data (text) before or after it is processed in computers. While "stop words" usually refers to a language's most common terms, all-natural language processing algorithms don't employ a single universal list. Stop words include words such as a, an, the, and others. Text normalisation is the process of converting previously uncanonical text into a single canonical form. Because input is guaranteed to be consistent before operations are done on it, normalising text before storing or processing it allows for separation of concerns.

Here we are going forward with two approaches: (Bag of Words & tfidf)

Bag of words Model

Below, we will call the fit transform method on Count Vectorizer. This will construct the vocabulary of the bag-of-words model and transform the sample sentence below into a sparse feature vector.

The Bag of Words (BoW) model is the most basic type of numerical text representation. A phrase can be represented as a bag of words vector, just like the term itself (a string of numbers).

Term Frequency-Inverse Document Frequency model (TFIDF)

It is used to convert text documents to matrix of tfidf features.

The term frequency-inverse document frequency statistic is a numerical measure of how essential a word is to a document in a collection.

Vectorizer used Above: Word Embeddings, also known as Word Vectorization, is an NLP technique for mapping words or phrases from a lexicon to a corresponding vector of real numbers, which can then be used to derive word predictions and semantics. Vectorization is the process of translating words into numbers.

Labeling the sentiment text and splitting sentiment data

Label Binarizer is a SciKit Learn class that takes Categorical data and outputs a Numpy array. Unlike Label Encoder, it encodes data into dummy variables that indicate whether a specific label is present or not. Label Binarizer is used to encode column data.

Modelling the dataset

Let us build Multinomial Naive Bayes model for both bag of words and tfidf features

The Multinomial Nave Bayes algorithm considers a feature vector in which each term reflects the number of times it appears or how frequently it appears, i.e. frequency.

Model Training

Training the model for both the approach both bag of words and tfidf features.

Model performance on test data

Model performance by predicting the model for both the approach both bag of words and tfidf features.

Accuracy of the model

Accuracy for both the approach using bag of words and tfidf features for different algorithms

a. Logistic Regression Model
```
lr_bow_score : 0.7512
lr_tfidf_score : 0.75
```
b. Linear support vector machines
```
svm_bow_score : 0.5829
svm_tfidf_score : 0.5112
```
c. Multinomial Naive Bayes
```
mnb_bow_score : 0.751
mnb_tfidf_score : 0.7509
```

We can observed that both logistic regression and multinomial naive bayes model performing well compared to linear support vector machines as the advantages of applying a Logistic function in this case is that model handles sparse matrices really well, and the weights can be interpreted as a probability for the sentiments.

notebook link: https://colab.research.google.com/drive/1ronvlNKFg-3L5mmozkiO2e-UNtf-i39v?usp=sharing