

IMDB Reviews

Text Mining and Search

Agazzi Ruben 844736, Cominetti Fabrizio 882737

Abstract

In this project, user reviews from the IMDB platform were analyzed through the use of text mining techniques. After carrying out an initial phase of text processing and text representation, the project continued with the classification of the reviews, through some text classification techniques - such as Support Vector Machines (SVM), Multilayer Perceptron (MLP), and Logistic Regression. Next, a text clustering phase was carried out through the use of two algorithms: DBSCAN and k-means.

Keywords

Text Mining — Text Classification — Text Clustering

University: University of Milan-Bicocca

Contents

1	Introduction	1
2	Data	1
3	Text Processing	2
4	Text Representation	2
5	Text Classification	2
5.1	SVM	2
5.2	Multilayer Perceptron	3
5.3	Logistic Regression	3
5.4	Evaluation	4
6	Text Clustering	4
6.1	DBSCAN	4
6.2	K-means	4
6.3	Evaluation	4
7	Summary	4

1. Introduction

The project aims to analyze the dataset "IMDB reviews" through text mining techniques, specifically through *Text Classification* and *Text Clustering*. The dataset contains a total amount of 50000 user-released reviews on the IMDB platform, divided in half between training and testing. The dataset is ideal for performing a binary sentiment classification task, the first objective of our analysis. Next, we decided to exploit text clustering techniques with the goal of identifying different clusters within the text.

Text Classification is the activity of predicting which data items belongs to a predefined finite set of classes. There are many types of classification, in our case it is *Binary Classification*, where each item belongs to exactly one class in a set of two (positive or negative).

In addition, text classification may be performed according to several dimensions ('axes') orthogonal to each other. For example, by topic (the most frequent case), by sentiment - our case -, by language, by type, by author, by native language, by gender, and more.

Text clustering, on the other hand, is the task of grouping a set of unlabeled texts in such a way that texts in the same cluster are more similar to each other than to those in other clusters.

2. Data

As stated before, the dataset used contains a total of 50000 user reviews on the IMDB platform, a platform that describes itself in the following manner : "IMDb is the world's most popular and authoritative source for movie, TV and celebrity content. Find ratings and reviews for the newest movie and TV shows" [IMDB].

The dataset is also defined as a "Large Movie Review Dataset". From an initial exploration of the data, we can observe that the dataset does not provide information about the date and reference film of the review, or any other indication, but contains only the text of the review and the extracted sentiment - positive or negative.

The data, initially divided into training and testing, but also between positive and negative sentiment, were merged, so that there would be a single dataset for the 25000 reviews to be used in the training phase and the 25000 reviews to be used in the testing phase.

Finally, the dataset contains precisely 12500 reviews labeled as positive and as many labeled as negative, both training and testing.

Having obtained the starting dataset, a series of *Text Processing* operations were performed:

- *Remove Numbers*, all numbers within the text have been removed;
- *Remove StopWords*, all words in the stopwords list have been removed;
- *Remove Punctuation*, all punctuation has been removed;
- *Remove Extra Space*, all extra spaces within the text have been removed;
- *Tokenization*, the process of breaking down a text into units called tokens;
- *Lower Case*, all words were converted to lower case;
- *Lemmaization*, the process of grouping together the inflected forms of a word.

Below we can observe some basic statistics for the train and the test set. Precisely, the number of words in the corpus, and the average review length.

	Train	Test
Number of words	24902	24798
Average review length	685.01	668.02

Table 1. Statistics for train and test data

In addition, we built two WordClouds, one for training set texts, the other for test set texts.

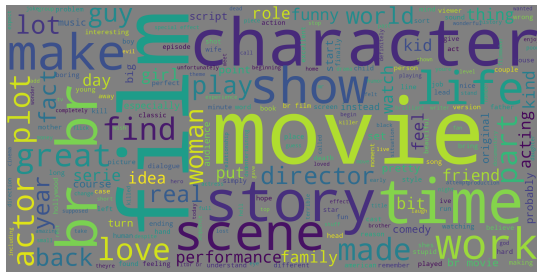


Figure 1. WordCloud - Training data

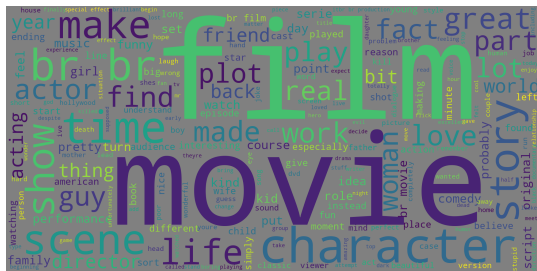


Figure 2. WordCloud - Test data

4. Text Representation

Once the corpus of texts had been properly processed, we moved on to the next stage of text representation.

Text Representation is the process to represent text with graphical methods. Considering the purposes of the project, the reviews were represented in structured form according to two methods: *Bag of Words* and *Tf-Idf*.

The Bag of Words representation identifies each document by a vector in which contains the number of occurrences of each word. This model doesn't consider grammar and order of words.

In the Tf-Idf representation, the number of occurrences of each word is weighted against the inverse of the word’s presence in the corpus.

The weights, called *Tf-Idf* weights, are the product of the two indices *Tf* and *Idf*:

$$w_{t,d} = \frac{tf_{t,d}}{\max(tf_{t_i,d})} \times \log(\frac{N}{df_t})$$

Where the Term Frequency $tf_{t,d}$ represent the frequency of the term t in the document d , divided by the frequency of the most occurring word in the document to prevent bias towards longer documents; and the the Inverse Document Frequency idf_t represents the inverse of the informativeness of the document for a term t .

The two representations were implemented through two features of the sklearn package in python: CountVectorizer for Bag of Words, TfidfVectorizer for Tf-Idf. The range of n-grams chosen for both was 1-2, that is, Uni-grams and Bi-grams.

5. Text Classification

Text Classification is the activity of predicting which data items belongs to a predefined finite set of classes. There are many types of classification, in our case it is *Binary Classification*, where each item belongs to exactly one class in a set of two (positive or negative).

To pursue the goals of our project, we chose to use three classification algorithms, and for each of them we used the two types of text representation developed earlier.

The selected algorithms are as follows: Support Vector Machine (SVM), Multilayer Perceptron (MLP), and Linear Regression (LR).

5.1 SVM

Support vector machines (SVMs) are a set of supervised learning methods used for classification and other tasks [SVM]. An SVM classifies data by finding the best hyperplane that separates all data points of one class from those of the other class. The best hyperplane for an SVM means the one with the largest margin between the two classes [SVM2]. The advantages of support vector machines are its effectiveness in high dimensional spaces, its effectiveness in cases where the

Text Mining and Search — 3/4

number of dimensions is greater than the number of samples, its memory efficiency, and its versatility. On the other side, the disadvantages of support vector machines include the fact that if the number of features is much greater than the number of samples, avoiding over-fitting in choosing Kernel functions and regularization term is crucial, and also SVMs do not directly provide probability estimates.

In our case, we used the LinearSVC package built on scikit-learn.

	precision	recall	f1-score	support
0	0.87	0.86	0.87	12500
1	0.86	0.87	0.87	12500
accuracy			0.87	25000
macro avg	0.87	0.87	0.87	25000
weighted avg	0.87	0.87	0.87	25000

Figure 3. Classification report - SVM with Bag-of-Words

	precision	recall	f1-score	support
0	0.82	0.78	0.80	12500
1	0.79	0.83	0.81	12500
accuracy			0.81	25000
macro avg	0.81	0.81	0.81	25000
weighted avg	0.81	0.81	0.81	25000

Figure 4. Classification report - SVM with Tf-Idf

As we can observe above, accuracy is better with Bag-of-Words, with a value of 0.87.

5.2 Multilayer Perceptron

Multilayer perceptron (MLP) is a supplement of feed forward neural network. It consists of three types of layers: the input layer, output layer and hidden layer [MLP].

The input layer receives the input signal to be processed. The required task such as prediction and classification is performed by the output layer. An arbitrary number of hidden layers that are placed in between the input and output layer are the true computational engine of the MLP.

In our case we exploited the potential of Keras. Keras is an open source library for machine learning and neural networks in Python.

	precision	recall	f1-score	support
0	0.82	0.87	0.85	12500
1	0.86	0.81	0.83	12500
accuracy			0.84	25000
macro avg	0.84	0.84	0.84	25000
weighted avg	0.84	0.84	0.84	25000

Figure 5. Classification report - MLP with Bag-of-Words

	precision	recall	f1-score	support
0	0.81	0.86	0.83	12500
1	0.85	0.79	0.82	12500
accuracy			0.83	25000
macro avg	0.83	0.83	0.83	25000
weighted avg	0.83	0.83	0.83	25000

Figure 6. Classification report - MLP with Tf-Idf

As we can observe above, accuracy is also better with Bag-of-Words, with a value of 0.83.

5.3 Logistic Regression

Logistic Regression is a statistical approach and a Machine Learning algorithm that is used for classification problems and is based on the concept of probability [LR]. It is widely used when the classification problem at hand is binary. Logistics regression uses the sigmoid function to return the probability of a label.

In our case, we used the LogisticRegression package built on scikit-learn.

	precision	recall	f1-score	support
Positive	0.85	0.87	0.86	12500
Negative	0.86	0.84	0.85	12500
accuracy			0.85	25000
macro avg	0.85	0.85	0.85	25000
weighted avg	0.85	0.85	0.85	25000

Figure 7. Classification report - LR with Bag-of-Words

	precision	recall	f1-score	support
Positive	0.87	0.87	0.87	12500
Negative	0.87	0.87	0.87	12500
accuracy			0.87	25000
macro avg	0.87	0.87	0.87	25000
weighted avg	0.87	0.87	0.87	25000

Figure 8. Classification report - LR with Tf-Idf

As we can observe above, in this case instead the accuracy is better with Tf-Idf, with a value of 0.87.

5.4 Evaluation

The highest accuracy value is obtained with the Logistic Regression algorithm with Tf-Idf weights. With the Support Vector Machine and Multilayer Perceptron algorithms, on the other hand, higher accuracy values are obtained with Bag-of-Words text representations.

6. Text Clustering

...

6.1 DBSCAN

...

6.2 K-means

...

6.3 Evaluation

...

7. Summary

...