Natural Language Processing

|  |
| --- |
| T.A. Mahinda Mahmoud |

**Movie review Sentiment analysis**

|  |  |  |
| --- | --- | --- |
| *Prepared by* | | |
| ID: | | **Name:** |
|  | **20180208** | **Karim Tarek** |
|  | **20182021** | **Abdelrahman Ayman** |
|  | **20173154** | **Ali Sonbol** |

Table of Contents

[**1.** **Abstract**: 3](#_Toc104509756)

[**2.** **Problem definition:** 3](#_Toc104509757)

[**3.** **Objectives:** 3](#_Toc104509758)

[**4.** **Intelligent system methodology** 4](#_Toc104509759)

[**5.** **Description of implementation:** 5](#_Toc104509760)

[**6.** **Test Results** 7](#_Toc104509761)

[**7.** **Discussion and comments on the experience gained** 7](#_Toc104509762)

[**8.** **Reference** 8](#_Toc104509763)

# **Abstract**:

As humans’ opinions help enhance products

efficiency, and since the success or the failure of a movie depends

on its reviews, there is an increase in the demand and need to

build a good sentiment analysis model that classifies movies

reviews.

As humans’ opinions help enhance products

efficiency, and since the success or the failure of a movie depends

on its reviews, there is an increase in the demand and need to

build a good sentiment analysis model that classifies movies

reviews.

As humans’ opinions help enhance products

efficiency, and since the success or the failure of a movie depends

on its reviews, there is an increase in the demand and need to

build a good sentiment analysis model that classifies movies

reviews.

As humans’ opinions help enhance products

efficiency, and since the success or the failure of a movie depends

on its reviews, there is an increase in the demand and need to

build a good sentiment analysis model that classifies movies

reviews.

As humans’ opinions help enhance products efficiency, and since the success or the failure of a movie depends on its reviews, there is an increase in the demand and need to build a good sentiment analysis model that classifies movies reviews.

Sentiment Analysis is concerned with identifying and categorizing opinions it is used for text analysis it combines natural language processing (NLP) and machine learning techniques to assign weighted sentiment scores to the entities. It is the process of determining whether a piece of writing is positive, negative or neutral

# **Problem definition:**

When it comes to sentiment analysis challenges, there are quite a few things that companies struggle with in order to obtain sentiment analysis accuracy. Sentiment or emotion analysis can be difficult in natural language processing simply because machines have to be trained to analyze and understand emotions as a human brain does.

Movie review analysis is the task of automatically detecting the nature of the review written if it is a good review or not. review analysis techniques assume that every review is either positive or negative review by collecting a number of keywords that till what the review summary is and is thus formulated as the task of identifying if it a positive or negative one.

In this problem, we will train a model that can predict the review if it is positive or negative.

# **Objectives:**

* **Dataset understanding**

At first, we must read the data set and understand the columns of the dataset for instance in our dataset there are two columns which are:

* + - **Review:** the comment written by users
    - **Sentiment:** comment about the movie whether it is positive or not
* **Data sampling**

As the data is too massive for our resources to handle so we will take a sample of the data to work with

* **Preprocessing on Dataset**
* **Lowercase**

We lowercased all the reviews to make it easier for the machine learning to later vectorize the data in a form of one hot encoder so there does not be a word with the same meaning but In a time with lower case and another with uppercase.

* **Punctuation removal**

We removed all the punctuations from the data to make it easier for the machine learning model to learn from the data.

* **Training and testing model**

the process where the performance of a fully trained model is evaluated on a testing set

# **Intelligent system methodology**

Sentiment analysis is complicated because of the slang words, misspellings, short forms, repeated characters, and use of regional language.

**Pre-Processing Phase:**

Preprocessing techniques are frequently used in natural language processing to prepare text that is going to be classified. Especially, reviews in e-commerce system, blogs and social media are informal, so they contain so much noisy information, unnecessary in detecting the sentiment. Those will clean text, normalize text and only keep useful information.

They removed URLs, hashtags, user mentions, punctuation, stop words, replaces slang words with actual words and lowercase all the sentences so there does not be a word with the same meaning but In a time with lower case and another with uppercase.

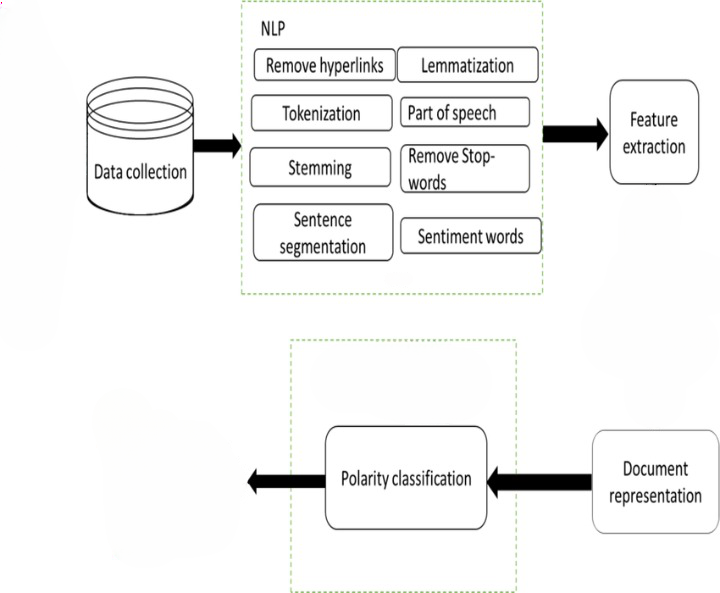
**Feature Extraction:**

Machine Learning algorithms learn from a pre-defined set of features from the training data to produce output for the test data. But the main problem in working with language processing is that machine learning algorithms cannot work on the raw text directly. So, we need some feature extraction techniques to convert text into a matrix (or vector) of features.

Some of the most popular methods of feature extraction are Bag-of-Words and TF-IDF

**Classification Phase:**

The classification step usually involves a statistical model like Naïve Bayes, Logistic Regression, Support Vector Machines, or Neural Networks



Sentiment Score

Polarity Classification

Document Representation

# **Description of implementation:**

1. **Data Set**

we used a data set from Kaggle The labelled data set consists of 25,000 IMDB movie reviews, specially selected for sentiment analysis. The sentiment of reviews is binary, meaning the IMDB rating < 5 results in a sentiment score of "Negative", and rating >=7 have a sentiment score of "Positive." No individual movie has more than 30 reviews.

1. **Sampling Data**

The second step is sampling data our data contain about 25000 rows. In NLP problems this is too massive data as the step of vectorization will increase the number of rows to maybe more than 16 GB of data so 40 percentage of the data is more than enough

1. **Feature Extraction:**

We cannot work with the text directly when using machine learning algorithms Instead, we need to convert the text to numbers.

We may want to perform a classification of reviews, so each review is an input and a class label is the output for our predictive algorithm. Algorithms take vectors (one hot encoding) of numbers as input, therefore we need to convert review to fixed-length vectors of numbers. A simple and effective model for thinking about text documents in machine learning is called the Bag-of-Words Model.

The model is simple in that it throws away all of the order information in the words and focuses on the occurrence of words in a document and gives each word a weight.

This can be done by assigning each word a weight. Then any document we see can be encoded as a fixed-length vector with the length of the vocabulary of known words. The value in each position in the vector could be filled with a count or frequency of each word in the encoded document.

This is the bag of words model, where we are only concerned with encoding schemes that represent what words are present or the degree to which they are present in encoded documents without any order information.

There are many ways to extend this simple method, both by better clarifying what a “word” is and in defining what to encode about each word in the vector.

The scikit-learn library provides different schemes that we can use, and we will briefly look at them.

1. **Classification:**

For tokenizing we will use The CountVectorizer which provides a simple way to both tokenize a collection of text documents and build a vocabulary of known words, but also to encode new documents using that vocabulary.

The next step is to apply machine learning techniques to our data but first, we must split our data into training data and testing data for this mission we will use train\_test\_split so what is this function?

train\_test\_split is a function in Sklearn model selection for splitting data arrays into two subsets: training data and testing data. With this function, you don't need to divide the dataset manually. By default, Sklearn train\_test\_split will make random partitions for the two subsets.

Now we can apply the StandardScaler to the dataset directly to standardize the input variables.

After that, we can implement machine learning models so we will be implementing 5 models which will be (linear regression, KNN, Decision Tree, Random Forest, and Naive Bayes) and deploy the model with the Highest accuracy.

# **Test Results**

So, the highest accuracy model was Random Forest model with 88,6 % of testing accuracy and after applying hyperparameter tuning using randomized search the accuracy jumped to 89% of testing accuracy.

# **Discussion and comments on the experience gained**

In this research, various techniques were used to identify the polarity of the movies. The algorithms performed were Logistic Regression, K-Nearest Neighbor, Random Forest. The best results were given by Random Forest classifier. The Random Forest classifier achieved 88.6% accuracy, Logistic Regression classifier we achieved 87.8 % accuracy, K-Nearest Neighbor classifier achieved 51.3% accuracy.

Therefore, with the discussion which has been carried out it has been clear that proper preprocessing of data based on natural language processing approaches as well as incorporating already existing models in the domain of sentiment analysis altogether with appropriate classification process can improve the performance of the model for multiclass classification of movie reviews.

This is just the start of sentiment analysis. Further approaches could use bigrams (sequences of two words) to attempt to retain more contextual meaning moreover, As only few algorithms were tested, it is required to test other algorithms or create hybrid methods so that accuracy of the results can be increased.

1. **Code**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# **Reference**

1. <https://www.kaggle.com/datasets/mwallerphunware/imbd-movie-reviews-for-binary-sentiment-analysis>
2. <https://www.researchgate.net/publication/318532057_Sentiment_Analysis_on_Movie_Reviews>
3. <https://www.researchgate.net/publication/332321070_Movies_Reviews_Sentiment_Analysis_and_Classification>
4. <https://www.geeksforgeeks.org/feature-extraction-techniques-nlp/>
5. <https://computationalsocialnetworks.springeropen.com/articles/10.1186/s40649-020-00080-x>
6. <https://www.researchgate.net/figure/The-flow-of-the-opinion-mining-process_fig1_357717019>