# INTERNSHIP PROJECT REPORT

# ON

# SENTIMENT ANALYSIS

## Presented By

## Pradnya Shende

## Executive Summary:

In this big-data era, machine learning is a trending research field. Machine learning enables data analytics to study massive data in an effective way. This technique is very helpful to classify and predict the content of language, which is also called natural language processing (NLP). One of the most prominent areas in NLP is sentimental analysis. Sentimental analysis in machine learning is usually applied on three levels, sentence level, document level, and aspect level. Sentence level analyses the sentiment on each sentence. Document level classifies the entire document as binary class or multi-class. Aspect level is a more complicated level, which is identifying the different aspects of a corpus first, and then classifying each document with respect to the observed aspects of each document.

The sentiment analysis is the process of using natural language processing and computational linguistics to extract, identify and categorize different opinions expressed in text format. It has attracted researchers from different disciplines, especially from computer science as it falls under interactive computation or human computer interaction (HCI). Sentiment analysis is mainly a classification problem that merges both domains natural language processing (NLP) and machine learning (ML). It has a wide range of applications such as opinion mining and business analytics. In addition, it has a potential to be implemented for governmental purposes to prevent suicide incidents.

In machine learning approach a more complex yet add sophistication to the system, depends on training different classifiers with a data set referred to as training set. Followed by an evaluation step by testing the classification performance by using the testing data set.

The report aims to classify the sentimental representations of Internet Movie Database (IMDb) reviews via machine learning based classification on document level.

## Dataset:

The dataset consists of 50,000 movie reviews taken from IMDb. Half of the data is used for training while the other half is used for testing. Moreover, both the training and testing dataset have 50% of positive reviews and 50% of negative reviews. The main goal is to estimate the sentiment many movie reviews from the Internet Movie Database (IMDb).

First, dataset is loaded and also, we loaded a range of libraries for general visualization to understand data and deal with unbalanced data.

The following table 1 shows top 5 entries of dataset:

**Table 1. Dataset Overview**

| | review | sentiment |
|---|---|---|
| **0** | One of the other reviewers has mentioned that ... | positive |
| **1** | A wonderful little production. <br /><br />The... | positive |
| **2** | I thought this was a wonderful way to spend ti... | positive |
| **3** | Basically there's a family where a little boy ... | negative |
| **4** | Petter Mattei's "Love in the Time of Money" is... | positive |

- This are top 5 records consist of 2 variables
- One of them is Predictor and another one is Target variable
- Sentiment is target variable and review is predictor

Let's check information about the dataframe.

| Column | Non-Null | Count | Dtype |
|---|---|---|---|
| 0 | review | 43525 | object |
| 1 | sentiment | 43525 | object |

- Data dimension: There is total 43525 observations and 2 variables
- Data types used in dataset is object
- There is no missing value in whole data

## Data Pre-processing:

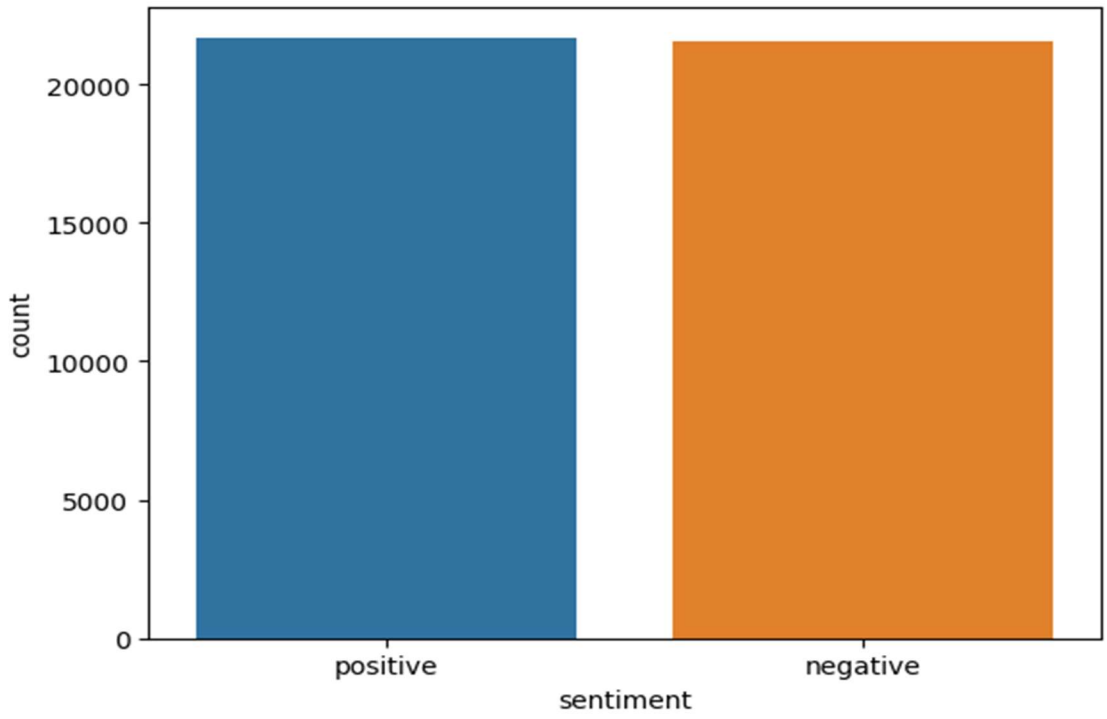
**Missing value treatment:**

As we don't have null values in our dataset. So, there is no need of missing value treatment.

**Removing Duplicates:**

There are 304 duplicated rows in our dataset. So, we dropped all the duplicated rows.

**Count Plot of Sentiment:**




The plot is equally distributed.

## **Stemming**:

Now coming to the stemming part, it basically is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words.

- First, we use the re package and remove everything that is not a letter (lower or uppercase letters).
- We then convert every uppercase letter to a lower one.
- We then split each sentence into a list of words.
- Then we use the stemmer and stem each word which exists in the column and remove every English stopword present in the list.
- We then join all these words which were present in the form of a list and convert them back into a sentence.
- Finally, we return the stemmed_content which has been pre-processed.

Then We will apply stemming function to our dataset. And next step is to name our input and output features.

Our last preprocessing step would be to transform our textual X to numerical so that our ML model can understand it and can work with it. This is where **TfidfVectorizer** comes into play.

TF-IDF is a measure of originality of a word by comparing the number of times a word appears in a doc with the number of docs the word appears in.

## Splitting the Dataset

This means that we have divided our dataset into 80% as training set and 20% as test set. *Stratify = y* implies that we have made sure that the division into train-test sets have around equal distribution of either classes (0 and 1 or Negative and Positive). random_state = 1 will guarantee that the split will always be the same.

Now, we will train the model. Let's check the accuracy of our training set predictions.

Accuracy score of the training data: 0.92
Accuracy score of the test data: 0.89
So, test accuracy is also pretty good.

## Building a System

Finally, to make this model useful we need to make a system. Taking a sample out of the test-set (I took the second sample) and checking our prediction for this sample.

With this we have built a system as well.