# MOVIE REVIEWS SENTIMENT ANALYZER

***Mini Project Report***

Submitted in partial fulfillment of the V Semester BE degree as per the requirements of Osmania University, Hyderabad

By


Anudeepa Jetangi 160619733138
B. Lavanya Durga 160619733125
P. Sahithi Reddy 160619733151



Department of Computer Science & Engineering
**Stanley College of Engineering and Technology for Women**
(Affiliated to OU & Approved by AICTE)
Chapel Road, Abids, Hyderabad - 500001


2021-2022

Stanley College of Engineering & Technology for Women
Chapel Road, Hyderabad
(Affiliated to OU & Approved by AICTE)

---

Ref No: SCETW/CSE Dept/Vth Semester 2022/MiniProject Date:22/01/2022

**CERTIFICATE**

This is to certify that the mini project titled “Movie Reviews Sentiment Analyzer” is a bonafied work carried over by Ms. Anudeepa Jetangi (H.T. No 160619733138), Ms. B.Lavanya Durga (H.T. No 160619733125) and Ms. P.Sahithi Reddy (H.T. No.160619733151) in partial fulfillment of the requirements for the award of the degree Bachelor of Engineering in Computer Science and Engineering from Osmania University during the III/IV Semester-I of their B.E. course during the academic year 2021-2022.

| | |
|---|---|
| Dr Y V S S Pragathi | Ms. Nadia Anjum |
| **Head, Department of CSE** | **Project Guide** |

# ACKNOWLEDMENT

First of all, we are grateful to The Almighty God for giving us the strength to complete this Mini Project. We pay our respects and love to our parents and all our Family members and friends for their love and encouragement throughout our career.

We wish to express our thanks to Sri. Kodali Krishna Rao, Correspondent and Secretary, for providing us with all the necessary facilities. We wish our sincere gratitude to Dr.Satya Prasad Lanka, Principal, for his constant encouragement. We deeply express our sincere gratitude to our Head of the Department, Dr Y V S S Pragathi, for encouraging and allowing us to present the Mini Project on the topic "Movie Reviews Sentiment Analyzer" at our department premises for the partial fulfillment of the requirements leading to the award of the B.E degree.

It is our privilege to express sincere regards to our Project Guide Ms. Nadia Anjum for the valuable inputs, guidance, encouragement, whole-hearted cooperation and constructive criticism throughout the duration of our project. We thank our Project Coordinator Mrs. Shivani for her guidance and support throughout the duration of our project.

We take this opportunity to thank all our faculty, who have directly or indirectly helped our project. Last but not the least, we express our thanks to our friends for their cooperation and support.

# TABLE OF CONTENTS

# ABSTRACT

Human’s 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. A movie review is an article reflecting its writer’s opinion about a certain movie and criticizing it positively or negatively, which enables everyone to understand the overall idea of that movie and make the decision whether to watch it or not. A movie review can affect the whole crew who worked on that movie.

Sentiment analysis is the analysis of emotions and opinions from any form of text. It is also termed as opinion mining. This technique is used to find the sentiment of the person with respect to a given source of content. In this project, a dataset consisting of 50,000 IMDB reviews is taken. Python NLTK library is used to perform tokenization to transfer the input string into a word vector, stemming is utilized to extract the root of the words, feature selection is conducted to extract the essential words, TF-IDF is used to convert the text reviews into numbers. In this sentiment analysis, machine learning Multinomial Naive Bayes Classification algorithm is used and the model is been trained. This model is then dumped into a pickle file.
The model is finally deployed using Flask- a micro web frame written in Python and run on a local host.

# LIST OF FIGURES

# 1. INTRODUCTION

Movie reviews are an important way to gauge the performance of a movie. While providing a numerical/stars rating to a movie tells us about the success or failure of a movie quantitatively, a collection of movie reviews is what gives us a deeper qualitative insight on different aspects of the movie. A textual movie review tells us about the strong and weak points of the movie and deeper analysis of a movie review can tell us if the movie in general meets the expectations of the reviewer

Sentiment Analysis is a major subject in machine learning which aims to extract subjective information from the textual reviews. The field of sentiment of analysis is closely tied to natural language processing and text mining. It can be used to determine the attitude of the reviewer with respect to various topics or the overall polarity of review. Using sentiment analysis, we can find the state of mind of the reviewer while providing the review and understand if the person was “happy”, “sad”, “angry” and so on.

The process of sentiment analysis includes tokenization, removing stop words, stemming, and classification. Tokenization is the identification of the basic units by the process of segmenting text into sentences and words. Tokenization is considered a pre-processing step.Stopwords are the words in any language which does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence Stemming is the process of removing prefixes and affixes to convert the word into its stem or root form. One vital data mining function is classification, which builds a model for labeling testing data based on previous training data. Different measures can be used to evaluate this model, such as accuracy, Area under the Curve (AUC), F-score, recall, and precision. Assigning classes (negative or positive) to reviews can be done by such model which predicts the label of new data.

In this project we aim to use Sentiment Analysis on a set of movie reviews given by reviewers and try to understand what their overall reaction to the movie was, i.e. if they liked the movie or they hated it. We aim to utilize the relationships of the words in the review to predict the overall polarity of the review.

## 1.1 Purpose

Humans are subjective creatures and their opinions are important because they reflect their satisfaction with products, services and available technologies. Being able to interact with people on that level has many advantages for information systems; such as enhancing products quality, adjusting marketing and business strategies, improving customer services, managing crisis, and monitoring performances. A study illustrates that in some cases, the success or the failure of a movie depends on its reviews. Therefore, a vital challenge is to be able to classify movies reviews to capture, retrieve, quantify and analyze watchers more effectively.

In last decade there is a rise of social media such as blogs and social networks, which has fueled the interest in sentiment analysis. Online opinion has turned into a kind of virtual currency with the proliferation of reviews, ratings, recommendations and other forms of online expression, for businesses that are looking to market their products, identify new opportunities and manage their reputations. In order to automate the process of filtering out the noise, understanding the conversations, identifying the relevant content and following appropriate actions, many are now looking to the field of sentiment analysis.

## 1.2 Scope

In this project, we aim to use sentimental analysis on a set of movie reviews given by viewers and try to understand what their overall reaction to the movies was, i.e., if they liked the movie or dislike it. We aim to utilize the relationship of the words in the review to predict the overall sentiment of it. If we focus, what is Sentiment analysis? Sentiment analysis is the interpretation and classification of emotions (positive negative and neutral) within the text data using text analysis techniques. Sentiment analysis tools allow the business to identify customers' sentiment towards products, brands, or services in online feedback.

Sentiment analysis of natural language texts is a vast and growing field. By using Natural Language Processing, we will make the computer truly understand more than just the objective definitions of the words. This analysis will help us segregate the data that has good as well as bad movie reviews. It includes using TF-IDF model which is a way of extracting features from the text for use in modelling. Not only that but also using classifier module to identify whether a given piece of text is positive or negative. We use ML algorithms (Multinomial Naïve Bayes, NLTK, TF-IDF) which suits best and feasible to our dataset and help the viewers with analysis models in the liberty of decision making.

## 1.3 Study of Existing System

The main aim of this project is to identify the underlying sentiment of a movie review on the basis of its textual information. Many classification algorithms have been used to identify the main sentiment from the given reviews such as random forest and SVM. But these algorithms have almost failed when it comes to dealing with huge datasets. SVM's are based on kernel function, due to which it stores the training examples as an NxN matrix of distances between the training points to avoid computing entries over and over again. This matrix consumes a lot so space and slows down the computation. Similarly, random forest classifier too consumes a lot of time in constructions trees for many features. Both, of these algorithms are not efficient in terms of time and space complexity. Thus, we explore more models in this project and finally choose the model that performs very well on the evaluation metrics and also takes the least computation time and space.

## 1.4 Proposed System

The field of Natural language processing converts text to signals which are understandable by the machine. It allows machines to understand how human speaks. Real-world applications include automatic text summarization, sentiment analysis, topic extraction, named entity recognition, parts-of-speech tagging, relationship extraction and stemming. We use NLP for text mining, machine translation, and automated question answering. In this project, we focus on Sentiment Analysis that helps us analyze people's sentiments, emotions, and evaluations from written language. The implementation of NLP models to identify a given set of words as positive sentiment or negative sentiment.

Any software engineer knows, there is a major distinction between the way people speak with each other and the way we "talk" with computers. When composing programs, we need to utilize precisely grammar and structure, yet when chatting with other individuals, we take a great deal of freedoms. We make short sentences. We make longer sentences, we layer in additional importance. We make numerous sentences with a similar importance. We discover numerous approaches to state a similar thing. You get the thought. It's convoluted! As counterfeit consciousness discovers its way into more of our gadgets and assignments, it turns out to be basically essential for us to have the capacity to speak with Computers in the dialect we're comfortable with. We can simply request that software engineers compose more projects, however we can't request that customers figure out how to compose code just to approach Siri for the climate. Customers must have the capacity to address computers in their "natural" dialect.

Analysis of the sentiments on Movie reviews from a platform like IMDB. By using Natural Language Processing, we will make the computer truly understand more than just the objective definitions of the words. This analysis will help us segregate the data that has good as well as bad movie reviews. It includes using Bag of words model which is a way of extracting features from the text for use in modeling. Not only that but also using Classifier module to identify whether a given piece of text is positive or negative. In this case, we are using Random Forest as our classifier using decision trees. The scikit library will help the algorithm learn with a faster curve, helper class to clean our data, Pandas will help us read our CSV files and NTLK removes unnecessary data from the dataset.

In the proposed system, we try to solve the below challenges: out of vocabulary words, errors in the user generated content, language ambiguity.

One fundamental purpose in sentiment analysis is categorization of sentiment polarity. Given a piece of written text, the problem is to categorize the text into one specific sentiment polarity, positive, negative or neutrality. With the existing system challenges, we now propose the model which performed very well on the evaluation metrics and also takes the least computation time and space are discussed below.

In this project, TF-IDF and Multinomial Naïve Bayes models have been used. Finally, NLTK package from Python is used to perform various data pre-processing steps like removing non-alphabetic characters, tokenization, removing stop words and stemming.

Below diagram defines the process for Sentiment Analysis process:

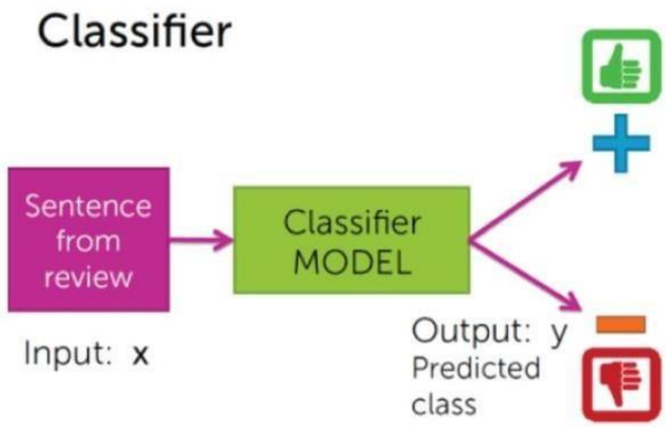


Fig 1.1 Sentiment Analysis Process

# 2. REQUIREMENTS

## 2.1 Software Requirements

We have used Anaconda for Python and Flask for creating a webapp.

**Anaconda**

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and mac OS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.

**Flask**

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, and upload handling, various open authentication technologies and several common framework related tools.

**Operating System**: Microsoft Windows 10

**Languages**: HTML, CSS, Python

## 2.2 Hardware Requirements

1. **Processor**: Intel CORE i5 processor with minimum speed 2.9 GHz speed.

2. **RAM** : Minimum 4 GB

# 3. ARCHITECTURE

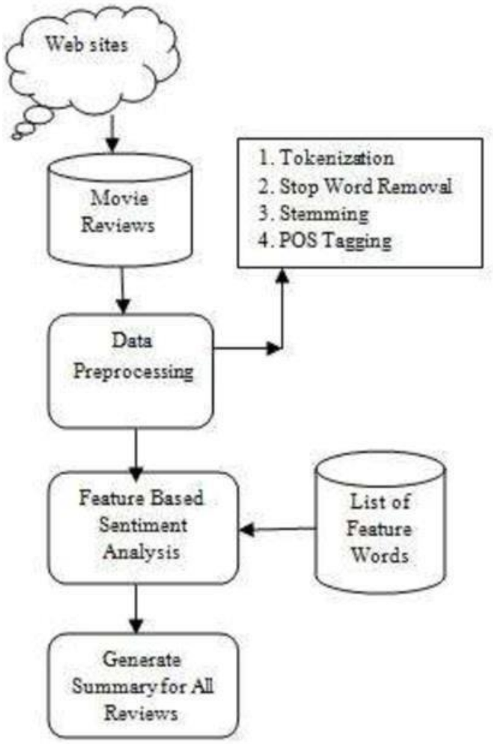


Fig3.1: Architecture of Movie Reviews Sentiment Analyzer.

The above architecture diagram represents the process that takes place throughout the Sentiment Analysis process.

Steps are as follows:

1) Input files are fed into the algorithm.
2) Data preprocessing is then performed, this involves tokenization, stop words removal, stemming. This cleaned data is feed into TF-IDF.
3) TF-IDF involves feature extraction. It gives more importance to less frequent words and less importance to more frequent words.
4) Use the Multinomial Naïve Bayes to train the dataset.
5) The classifier will divide the data into either positive or negative reviews.
6) This trained model is dumped into a pickle file, the data which is received in the pickle gets analyzed with the text review and finally, sentiments captured as positive or negative is viewed.

**Data Preprocessing:**

Data preprocessing is an important tool for Data Mining (DM) algorithm. Movie data is an unstructured dataset, it is a collection of information from people entered his/her feelings, opinion, attitudes etc. This type of information is growing day by day in the internet. This data consists of lots of spelling mistakes, unnecessary punctuations, break statements, hyperlinks etc. All this makes the data more complex and such complexity reduces the speed of training the machine learning model. Thus, it’s important to preprocess the data and remove all such elements at the beginning itself.
The methods used for preprocessing is tokenization, stop words removal, tokenization

**Feature Extraction**:

We used 2 methods for extraction of meaningful features from the review text which could be used for training purposes. These features were then used for training several classifiers.

1) Bag of Words: This is a typical way for word representation in any text mining process. We first calculated the total word counts for each word across all the reviews and then used this data to create different feature representations. . This ensured that we remove most of the misspelled words. Also, words which occurred only once in the dataset would contribute nothing to the classifier. Another feature representation was created along the same lines but with words occurring at least 5 times.
2) TF-IDF : While the above method of feature extraction descried concentrated more on higher frequency parts of the review they completely ignored the portions which might be less frequent but have more significance for the overall polarity of the review. To account for this, we created feature representations of words using TFIDF. The feature representation for this model is similar to the Bag of Words model except that we used TF-IDF values for each word instead of their frequency counts.

**Generate a Summary for the review**:

The review that is given as input in the web app, goes through the preprocessing steps, the essential words/features are extracted from the text. This is the given to the machine learning model. The model predicts the sentiment and shows it on the output screen.

# 4. SOFTWARE DESIGN

## 4.1 UML Diagrams

To model a system, the most important aspect is to capture the dynamic behaviour. Dynamic behaviour means the behaviour of the system when it is running/operating. Only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction. These internal and external agents are known as actors. Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

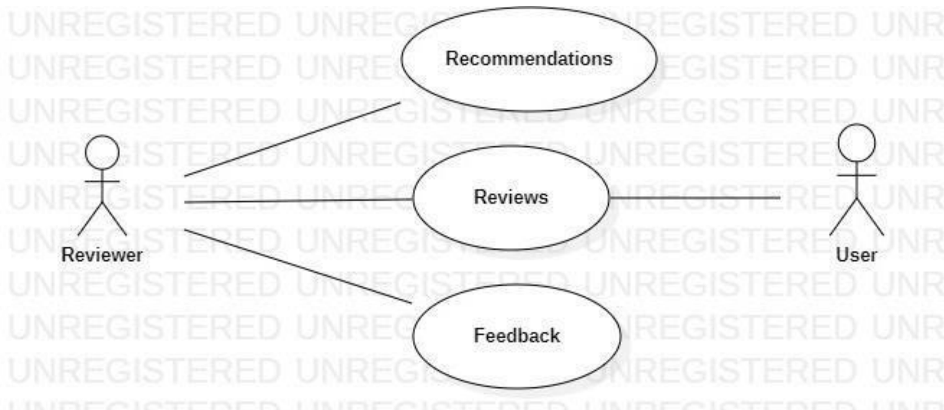


Fig4.1: Use Case Diagram of Movie Reviews Sentiment Analyzer

## 4.2 E-R Diagram

An Entity–relationship model (ER model**)** describes the structure of a database with the help of a diagram, which is known **as** Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.
An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Lets have a look at a simple ER diagram to understand this concept.

In the following diagram we have 3 entities feeback, Member,Movie and 2 relationships Givenby,Rate.

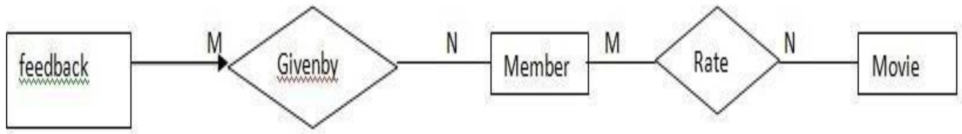


Fig4.2: E-R Diagram of Movie Reviews Sentiment Analyzer

## 4.3 Modules

The Movie Reviews Sentiment Analyzer Web App consists of the following modules: Data Preprocessing, Training Data, Evaluation, and Deployment using Flask.
In this project, TF-IDF and Multinomial Naïve Bayes models have been used for training data. Finally, NLTK package from Python is used to perform various data pre-processing steps like removing non-alphabetic characters, tokenization, removing stop words and stemming.

### 4.3.1 TF-IDF

This is short for Term-frequency-Inverse-Document-Frequency and gives us a measure of how important a word is in the document.

Term Frequency (Tf):

$$Tf = \frac{number\ of\ times\ term\ appears\ in\ document}{total\ number\ of\ words\ in\ document}$$

Inverse Document Frequency (IDF):

$$Idf = \ln\left(\frac{total\ number\ of\ documents}{number\ of\ documents\ with\ term\ in\ them}\right)$$

We combine both to get Tf-Idf

### 4.3.2 Multinomial Naïve Bayes

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem.

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

Framing it in terms of this problem we have y representing a positive or negative review, X represents a vector of the features from the reviews. The equation we end up with for finding the class is:

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

Naive Bayes algorithms are mostly used in sentiment analysis, spam filtering, and recommendation systems etc. They are mostly used when features are independent and predictors are frequency of words present in the document.

### 4.3.3 NLTK

We need to decide how to deal with frequently occurring words that don't carry much meaning. Such words are called "stop words"; in English, they include words such as "a", "and", "is", and "the". Conveniently, there are Python packages that come with stop word lists built in. Let's import a stop word list from the Python Natural Language Toolkit (NLTK). The following commands to use the library:

```
Import    nltk
nltk.download()
```

## 4.3.4 Dataset

The dataset used for this task was collected from Large Movie Review Dataset which was used by the AI department of Stanford University for the associated publication. The dataset contains 50,000 training examples collected from IMDB where each review is labelled with the rating of the movie on scale of 1-10. As sentiments are usually bipolar like good/bad or happy/sad or like/dislike, we categorized these ratings as either 1 (like) or 0 (dislike) based on the ratings. If the rating was above 5, we deduced that the person liked the movie otherwise he did not. Initially the dataset was divided into two subsets containing 25,000 examples each for training and testing. We found this division to be sub-optimal as the number of training examples was very small and leading to under-fitting. We then tried to redistribute the examples as 40,000 for training and 10,000 for testing.

A typical review text looks like this:

*"I'm a fan of TV movies in general and this was one of the good ones. The cast performances throughout were pretty solid and there were twists I didn't see coming before each commercial. To me it was kind of like Medium meets CSI. Did anyone else think that in certain lights, the daughter looked like a young Nicole Kidman? Are they related in any way? I'd definitely watch it agin or rent it if it ever comes to video.Dedee was great. Haven't seen in her in a lot of things and she did her job very convincingly.If you're into to TV mystery movies, check this one out if you have a chance."*

As seen above, one necessary pre-processing step prior to feature extraction was removal of HTML tags like "". We used simple regular expressions matching to remove these HTML tags from the text. Another important step was to make the text case-insensitive as that would help us count the word occurrences across all reviews and prune unimportant words. We also removed all the punctuation marks like '!', '?', etc as they do not provide any substantial information and are used by different people with varying connotations. This was achieved using standard python libraries for text and string manipulation. We also removed stopwords from the text for some of our feature extraction tasks, which is described in greater detail in later sections. One important point to note is that we did not use stemming of words as some information is lost while stemming a word to its root form.

### 4.3.5 Predictive Task

The main aim of this project is to identify the underlying sentiment of a movie review on the basis of its textual information. In this project, we try to classify whether a person liked the movie or not based on the review they give for the movie. This is particularly useful in cases when the creator of a movie wants to measure its overall performance using reviews that critics and viewers are providing for the movie. The outcome of this project can also be used to create a recommender by providing recommendation of movies to viewers on the basis of their previous reviews. Another application of this project would be to find a group of viewers with similar movie tastes (likes or dislikes).

As a part of this project, we aim to study several feature extraction techniques used in text mining e.g. keyword spotting, lexical affinity and statistical methods, and understand their relevance to our problem. In addition to feature extraction, we also look into different classification techniques and explore how well they perform for different kinds of feature representations. We finally draw a conclusion regarding which combination of feature representations and classification techniques are most accurate for the current predictive task.

# 5. CODE IMPLEMENTATION

**Static/style.css –** This is a style sheet that is used for styling the front end of the web app.

```
html{
   height: 100%;
   margin: 0;
}

body{
   font-family: Arial,Helvetica,sans-serif;
   text-align: center;
   margin: 0;
   padding: 0;
   width: 100%;
   height: 100%;
   display: flex;
   flex-direction: column;
}

/*Website Title*/

.container{
   padding: 30px;
   position: relative;
   background: linear-gradient(45deg, #ffffff, #ffffff, #f9f9f9, #eeeeee, #e0e4e1,
#d7e1ec);
   background-size: 500% 500%;
   animation: change-gradient 10s ease-in-out infinite;
}

@keyframes change-gradient {
        0%{
                background-position: 0 50%;
        }

        50%{
                background-position: 100% 50%;
        }
        100%{
                background-position: 0 50%;
        }
}
```

```
.container-heading{
   margin:0;
}

.heading_font{
 color:#5530B4;
 font-family: 'Yatra One',cursive;
 font-size: 27px;
}

.description{
 font-style: italic;
 font-size:  14px;
 margin: 3px 0 0;
}

/*Text Area */
.ml-container{
 margin: 30px 0;
 flex: 1 0 auto;
}
.message-box{
 margin-bottom: 20px;
}

/*Predict Button */
.my-cta-button{
 background: #f9f9f9;
 border: 2px solid #000000;
 border-radius: 1000px;
 box-shadow: 3px 3px #8c8c8c;
 padding: 10px 36px;
 color: #000000;
 display: inline-block;
 font: italic bold 20px/1 "Calibri", sans-serif;
 text-align: center;
}
.my-cta-button:hover{
 color: #4d089a;
 border: 2px solid #4d089a;
}

.my-cta-button:active{
 box-shadow: 0 0;
}
```

```
/*Footer*/
.footer{
  font-size: 14px;
  padding:20px;
  flex-shrink: 0;
  position: relative;
  background: linear-gradient(45deg, #ffffff, #ffffff, #f9f9f9, #eeeeee, #e0e4e1,
#d7e1ec);
	background-size: 500% 500%;
	animation: change-gradient 10s ease-in-out infinite;
}

.footer-description{
  margin: 0;
  font-size: 12px;
}

/*Result*/
.results{
  padding: 30px 0 0;
  flex: 1 0 auto;
}

.safe{
  color: green;
}

.danger{
  color: #ff0000;
}
```

**Templates/index.html -** This is a html file that consists of the code needed for the home page where the reviews/input is given.

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
    <meta charset="utf-8">
    <title>Movie Reviews Sentiment Analyser</title>
    <link rel="shortcut icon" href="{{ url_for('static',filename='Movies.ico') }}">
    <link rel="stylesheet" type='text/css' href="{{ url_for('static',filename='style.css')
}}">
    <script src="https://kit.fontawesome.com/5f3f547070.js"
crossorigin="anonymous"></script>
    <link href="https://fonts.googleapis.com/css2?family=Yatra+One&display=swap"
rel="stylesheet">
```

```
</head>
  <body>
    <!--Website Title -->
    <div class="container">
      <h2 class='container-heading'><span class="heading_font">Movie Review's
</span>Sentiment Analyser</h2>
      <div class="description">
        <p2> A Machine Learning Web App, Built in Flask </p2>
      </div>
    </div>

    <!--Text Area -->
    <div class="ml-container">
      <form action="{{ url_for('predict') }}" method="POST">
        <textarea class="message-box" name="message" rows="15" cols="75"
placeholder="Enter Your Review Here ..."></textarea><br/>
        <input type="submit" class="my-cta-button" value="Predict">
      </form>
    </div>

    <!--Footer-->
    <div class="footer">
      <p class="footer-description">Made with ’ by cse_mythros.</p>
    </div>

  </body>
</html>
```

**Templates/result.html -** This is a html file that is used to show the results obtained after prediction of the review (given as input).

```
<!DOCTYPE html>

<html lang="en" dir="ltr">
<head>
    <meta charset="utf-8">
    <title>Movie Reviews Sentiment Analyser</title>
    <link rel="shortcut icon" href="{{ url_for('static',filename='Movies.ico') }}">
    <link rel="stylesheet" type='text/css' href="{{ url_for('static',filename='style.css')
}}">
    <script src="https://kit.fontawesome.com/5f3f547070.js"
crossorigin="anonymous"></script>
    <link href="https://fonts.googleapis.com/css2?family=Yatra+One&display=swap"
rel="stylesheet">
  </head>
```

```html
<body>
<!--Website Title-->
   <div class="container">
     <h2 class="container-heading"><span class="heading_font">Movie Review's
</span>Sentiment Analyser</h2>
     <div class="description">
       <p>A Machine Learning Web App, Built with Flask </p>
     </div>
   </div>

   <!--Result-->
   <div class="results">
     {% if prediction == 1 %}
<h1>Prediction: <span class="safe">POSITIVE Review </span>
     {% elif prediction == 0 %}
       <h1>Prediction: <span class="danger">NEGATIVE Review </span>
     {% endif %}
   </div>

   <!--Footer-->
   <div class="footer">
     <p>Made with  by cse_mythros.</p>
   </div>

  </body>
</html>
```

**app.py -** This is the python file that is first executed in flask. Using this file all the other files are rendered/called and executed.

```python
#Importing Essential Libraries
from flask import Flask,render_template,request
import pickle

#Load Multinomial Navie Bayes and Tf-IDF Vectorizer from disk
filename = 'movie-reviews-sentiment-model.pkl'
classifer = pickle.load(open(filename,'rb'))
tfid = pickle.load(open('tfidf-transform.pkl','rb'))

app = Flask(_name_)

@app.route('/')
def home():
   return render_template('index.html')
```

```
@app.route('/predict',methods=['POST'])
def predict():
    if request.method == 'POST':
        message = request.form['message']
        data = [message]
        vect = tfid.transform(data)
        my_prediction = classifer.predict(vect)
        return render_template('result.html',prediction=my_prediction)

if__name___== '_main_':
    app.run(debug=True)
```

**deployment.py -** This is a python file where the dataset of 50,000 reviews is read. This data is preprocessed and then fed into a multinomial naïve Bayes algorithm

```
import pandas as pd
import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
import nltk.corpus
import re
import string
import pickle

#Reading the data
df = pd.read_csv(r'C:\Users\DELL\Desktop\imbd_movie\IMDB  Dataset.csv')

#Turning sentiment into categorical values
# 1 - positive , 0 - negative
from sklearn.preprocessing import LabelEncoder
l = LabelEncoder()
df['sentiment'] = l.fit_transform(df['sentiment'])

#Remove special characters from text
pattern = re.compile(r'<br\s*/><br\s*/>>*|(\-)|(\\)|(\/)')
def preprocess_reviews(reviews):
  reviews = [pattern.sub('',item) for item in reviews]
  return reviews
clean = preprocess_reviews(df['review'])

#Removing punctuations
pattern = re.compile(r'[^\w\s]')
```

```
def remove_punctuations(reviews):
  reviews = [ pattern.sub('',item) for item in reviews]
  return reviews
clean = remove_punctuations(df['review'])


df['review'] = clean
#Converting the text to lowercase
df['review'] = df['review'].str.lower()

#Removing line breaks
def remove_linebreaks(input):
  pattern = re.compile(r'\n')
  return pattern.sub('',input)
df['review'] = df['review'].apply(remove_linebreaks)

#Tokenization
from nltk.tokenize import word_tokenize
df['review'] = df['review'].apply(word_tokenize)

#Removing stopwords
from nltk.corpus import stopwords
def remove_stopwords(reviews):
 return [w for w in reviews if w not in stopwords.words('english')]
df['review'] = df['review'].apply(lambda x: remove_stopwords(x))

#Lemmatization
from nltk.stem import WordNetLemmatizer
lem = WordNetLemmatizer()
def word_lemmatize(input):
  return [lem.lemmatize(word) for word in input]
df['review'] = df['review'].apply(word_lemmatize)

#Combining all the individual words
def combine_words(input):
  combined = ' '.join(input)
  return combined
df['review'] = df['review'].apply(combine_words)

y = df.iloc[:,-1].values

from sklearn.feature_extraction.text import TfidfVectorizer
tfid = TfidfVectorizer(min_df=2,max_df=0.5,ngram_range=(1,3))
```

```
X = tfid.fit(df['review'])
X = tfid.transform(df['review'])
X.toarray()

# Creating a pickle file for the TfidfVectorizer
pickle.dump(tfid, open('tfidf-transform.pkl', 'wb'))

#Splitting data into train and test set
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test   =  train_test_split(X,y,test_size=0.2,random_state=0)

#Model Building
from sklearn.naive_bayes import MultinomialNB
classifier = MultinomialNB()
classifier.fit(X_train,y_train)
# Creating a pickle file for the Multinomial Naive Bayes model
filename = 'movie-reviews-sentiment-model.pkl'
pickle.dump(classifier, open(filename, 'wb'))
```

**movie_reviews_sentiment_analyser_ipynb -**This is a ipython notebook where the exploratory data analysis is performed and data is fed into different algorithms, Each of the algorithm is evaluated and the best algorithm is chosen.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
import nltk.corpus
import re
import string
import warnings
warnings.filterwarnings('ignore')

df = pd.read_csv('/content/drive/MyDrive/Datasets/IMDB Dataset.csv')

#EDA

sns.countplot(data=df,x='sentiment')

positive_df = df[df['sentiment'] == 'positive']
positive_df.head(3)
```

```
#Plotting word cloud for positive reviews
from wordcloud import WordCloud
wordcloud = WordCloud(width=500,height=250,max_font_size=80,max_words=150,background_color='white').generate(positive_df.review[0])
f = plt.figure()
f.set_figwidth(15)
f.set_figheight(10)
plt.imshow(wordcloud,interpolation='bilinear')
plt.axis('off')
plt.margins(x=0,y=0)
plt.show()
```

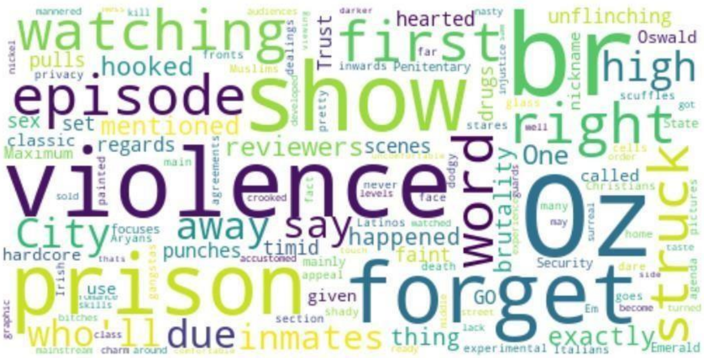


Fig5.1: WordCloud for positive reviews

```
negative_df = df[df['sentiment'] == 'negative']
negative_df.head(3)
#Plotting word cloud of negative sentiment
from wordcloud import WordCloud
wordcloud = WordCloud(width=500,height=250,max_font_size=80,max_words=150,background_color='white').generate(negative_df.review[11])
f = plt.figure()
f.set_figwidth(15)
f.set_figheight(10)
plt.imshow(wordcloud,interpolation='bilinear')
plt.axis('off')
plt.margins(x=0,y=0)
plt.show()
```

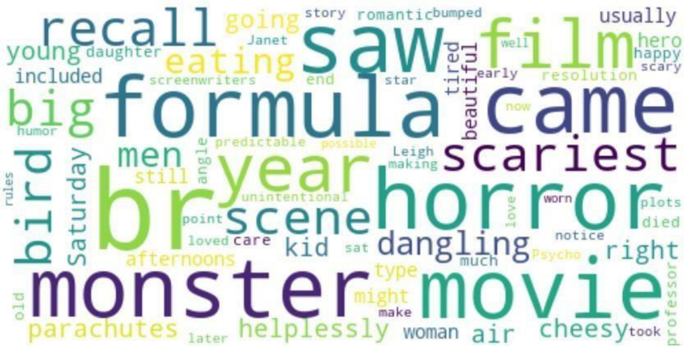

Fig5.2: WordCloud for Negative Reviews

```
#analyzing Machine Learning models

from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import MultinomialNB,GaussianNB,BernoulliNB

rf = RandomForestClassifier(n_jobs=2,random_state=2)
nbm = MultinomialNB()
nbb = BernoulliNB()

rf.fit(X_train,y_train)
nbm.fit(X_train,y_train)
nbb.fit(X_train,y_train)

y_pred1 = rf.predict(X_test)
y_pred2 = nbm.predict(X_test)
y_pred4 = nbb.predict(X_test)
from sklearn.metrics import accuracy_score
acc1 = accuracy_score(y_test,y_pred1)
acc2 = accuracy_score(y_test,y_pred2)
acc4 = accuracy_score(y_test,y_pred4)
#Plotting Models Accuarcy Scores
fig = plt.figure(figsize=(5,3))
ax = fig.add_axes([0,0,1,1])
models = ['MultinomialNB','RandomForest','BernoulliNB']
accuracy = [acc2*100,acc1*100,acc4*100]
ax.bar(models,accuracy,color='bgmc',width=0.8)
plt.xlabel('ML model',size=15)
plt.ylabel("Accuracy",size=15)
```

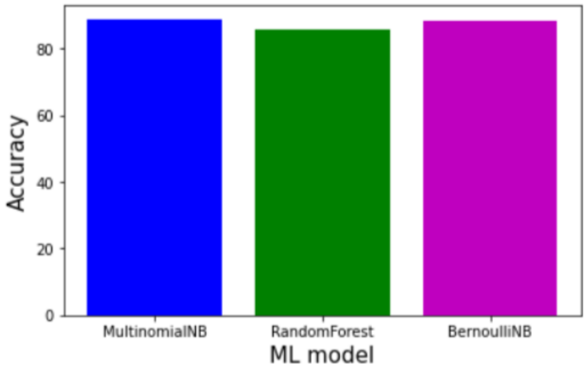


Fig5.3: Bar Graph showing the accuracy of various models

Both BernoulliNB and MultiNomialNB are having high accuracy. Let's find out the classification report and confusion matrix for all the algorithms to find the best as some algorithms perform well on accuracy score but fail to perform well on other metrics.

from sklearn.metrics import classification_report,confusion_matrix

*MultiNomial Naive Bayes*

print(classification_report(y_test,y_pred2))

```
precision    recall  f1-score   support

           0       0.89      0.88      0.89      5035
           1       0.88      0.89      0.89      4965

    accuracy                           0.89     10000
   macro avg       0.89      0.89      0.89     10000
weighted avg       0.89      0.89      0.89     10000
```

sns.heatmap(confusion_matrix(y_test,y_pred2),annot=True,fmt='d')

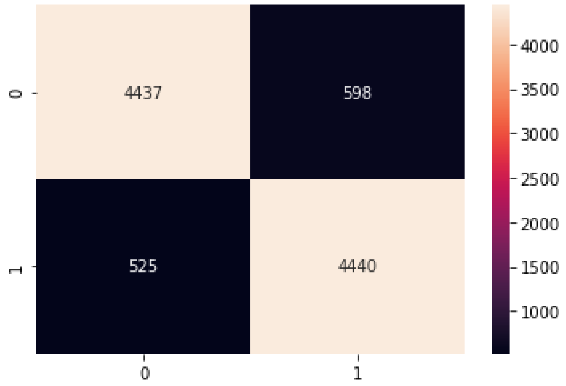


Fig5.4: Confusion matrix of Multinomial Naïve Bayes

*Random Forest Classifier*

```
print(classification_report(y_test,y_pred1))
```

```
precision    recall  f1-score   support

           0       0.86      0.86      0.86      5035
           1       0.86      0.86      0.86      4965

    accuracy                           0.86     10000
   macro avg       0.86      0.86      0.86     10000
weighted avg       0.86      0.86      0.86     10000
```

```
sns.heatmap(confusion_matrix(y_test,y_pred1),annot=True,fmt='d')
```

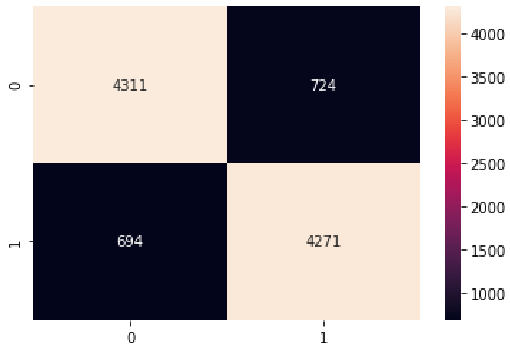


Fig5.5: Confusion Matrix of Random Forest Classifier

*Bernoulli Naïve Bayes*

print(classification_report(y_test,y_pred4))

```
precision     recall   f1-score    support

           0       0.90       0.86       0.88       5035
           1       0.87       0.90       0.89       4965

    accuracy                             0.88      10000
   macro avg       0.88       0.88       0.88      10000
weighted avg       0.88       0.88       0.88      10000
```

sns.heatmap(confusion_matrix(y_test,y_pred4),annot=True,fmt='d')

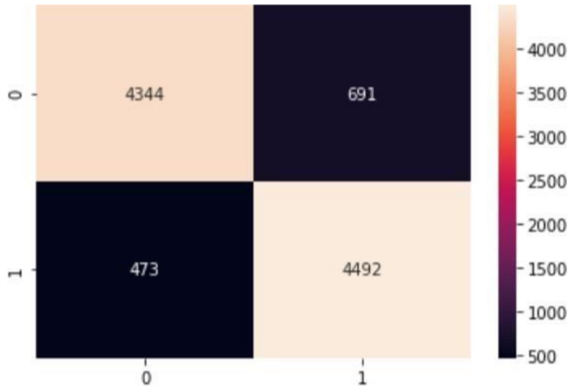


Fig5.6 Confusion matrix of Bernoulli Naïve Bayes

In movie reviews sentiment analysis, False Positive error is considered to be worse. As, if a negative review is predicted as positive, then it might appear in the website and form a bad opinion about the movie.Thus,we must try to minimize FP and choose an algorithm with least FP.

Though, BernoulliNB has high accuracy but low recall i.e. it predicts many reviews False positive. Thus, we choose MultiNomialNB has high accuracy, precision, recall, and f1 score and has low false positives. It’s the best algorithm to be chosen.

Multinomial Naive Bayes is finally chosen among the other algorithms.

# 6. TESTING

Testing is the process of evaluating a system or its component's with the intent to find that whether it satisfies the specified requirements or not. This activity results in the actual, expected and difference between their results i.e testing is executing a system in order to identify and gaps, errors or missing requirements in contrary to the actual desire or requirements.

## 6.1 Types of Software Testing

## 6.1.1 Manual Testing

Manual Testing is a type of software testing in which test cases are executed manually by a tester without using any automated tools. The purpose of Manual Testing is to identify the bugs, issues, and defects in the software application. Manual software testing is the most primitive technique of all testing types and it helps to find critical bugs in the software application.Any new application must be manually tested before its testing can be automated. Manual Software Testing requires more effort but is necessary to check automation feasibility

Manual Testing can be further divided into:

1. White Box Testing
2. Black Box Testing

**White Box Testing**

White Box testing is the detailed investigation of internal logic and structure of the code. To perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code. The tester needs to have a look inside the source code and find out which unit of the code is behaving inappropriately.

**Unit Testing**

Unit Testing is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.In SDLC, STLC, V Model, Unit testing is first level of testing done before integration testing. Unit testing is a WhiteBox testing technique that is usually performed by the developer.

**Black Box Testing**

The technique of testing without having any knowledge of the interior workings of the application. The test is oblivious to the system architecture and does not have access to the source code. Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examine outputs without knowing how and where the inputs are worked upon.

## 6.1.2 Automatic Testing

Automation Testing or Test Automation is a software testing technique that performs using special automated testing software tools to execute a test case suite. On the contrary, Manual Testing is performed by a human sitting in front of a computer carefully executing the test steps.The automation testing software can also enter test data into the System Under Test, compare expected and actual results and generate detailed test reports. Test Automation demands considerable investments of money and resources.

Successive development cycles will require execution of same test suite repeatedly. Using a test automation tool, it's possible to record this test suite and re-play it as required. Once the test suite is automated, no human intervention is required. This improved ROI of Test Automation. The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing altogether.

## TESTCASES

| Sr.No | Test Case Scenario | Expected Output | Actual Output | Status |
|---|---|---|---|---|
| 1 | Yet while it's not very good at being scary, "The Grudge" excels at being unsettling | Negative Review | Positive Review | Failed |
| 2 | "DilBechara"There is no better way to give tribute to a genuinely amazing actor. The film was pretty emotional and touching. Such natural acting and a brilliant execution from other the cast and crew. | Positive Review | Positive Review | Passed |
| 3 | If the creators are going to make another Godzilla flick, then they should either increase Godzilla's screen time or at the very least bring in better actors and create more memorable characters. That said,it's still a fun movie on its own merits. | Positive Review | Negative Review | Failed |
| 4 | The entire movie was a flashback. That was really annoying for me. Gal Gadot posed a lot. I guess people like that.. Boring watch. was lackluster too. | Negative Review | Negative Review | Passed |

| | | | | |
|---|---|---|---|---|
| 5 | Watched "soorarai potru" ...inspiring tale with lot of life .I don't have much to say about this one as I ain't having words to describe this master class work ..such is the impact...brilliantly scripted with some outstanding performance chained with some remarkable making and cuts. | Positive Review | Positive Review | Passed |
| 6 | Tenet- Its worth a watch either way. See it with subtitles if you can. And definitely don't expect to fully understand whats going on the first time around. | Positive Review | Positive Review | Passed |
| 7. | Avengers End game – This movie is better than joker. Hm.. Totally suckss. Couldn't understand a single thing. | Negative Review | Negative Review | Passed |
| 8 | Parasite - A miracle of a film. It feels like Bong Joon-ho’s already extraordinary career has been building to this: a riotous social satire that’s as gloriously entertaining as it is deeply sardonic. | Positive Review | Positive Review | Passed |

# 7. OUTPUT SCREENS


Fig7.1: Output Screen1

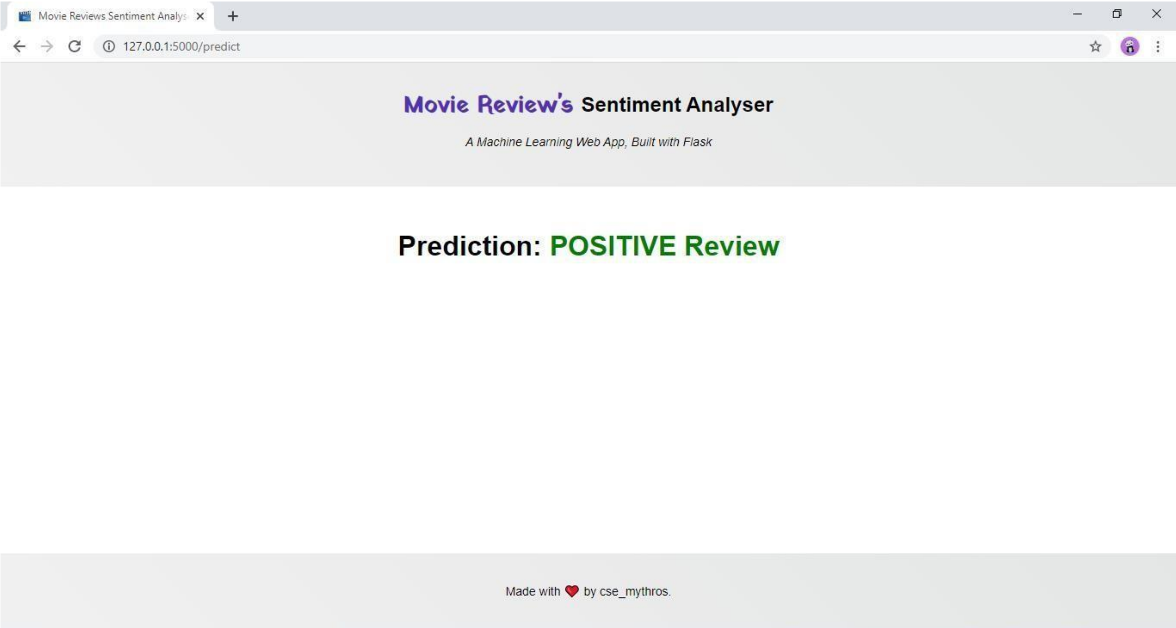


Fig7.2: Output Screen2

# 8. REFERENCES


1) A. Hogenboom, F. Frasincar, F. de Jong, and U. Kaymak,. 2015, Using Rhetorical Structure in Sentiment Analysis, Communications of the ACM, vol. 58, no. 7, pp. 69–77.
2) Kamal, A. 2013 Subjectivity Classification using Machine Learning Techniques for Mining Feature- Opinion Pairs from Web Opinion Sources. International Journal of Computer Science Issues 10(5), 191- 200.
3) Akshay Amolik, Niketan Jivane, Mahavir Bhandari, Dr .M. Venkatesan, Twitter Sentiment Analysis of Movie Reviews using Machine Learning Techniques, School of Computer Science and Engineering, VIT University, Vello.
4) Humera Shaziya, G.Kavitha, Raniah Zaheer, 2015, Text Categorization of Movie Reviews for Sentiment International Journal of Computer Applications (0975 – 8887) Volume 179 – No.7, December 2017.
5) Mais Yasen, Sara Tedmori Movies Reviews Sentiment Analysis and Classification 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)At: Amman, Jordan.
6) ACME, "The Significance of a Film Review", retrieved on: December 22, 2018, from: www.revue-acme.com.
7) Mshne, Gilad and Natalie Glance, (2006), "Predicting Movie Sales from Blogger Sentiment", AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, PP 1-4
8) CraigTrim, "The Art of Tokenization", retrieved on: December 22, 2018, from: www.ibm.com.
9) Lovins, Julie Beth, (1968), "Development of a Stemming Algorithm", Mechanical Translation and Computational Linguistics, Vol. 11, PP 22–31.
10) Palak Baid, Apoorva Gupta, Neelam Chaplot Sentiment Analysis of Movie Reviews using Machine Learning Techniques December 2017International Journal of Computer Applications 179(7) Palak Baid.