SENTIMENT ANALYSIS ON MOVIE REVIEWS

An Internship Report Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

by

Akash Agarwal

(Univ. no. 2110014325030)

Under the Guidance of

Mr. Ankit Singh

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

FACULTY OF ENGINEERING & TECHNOLOGY,

UNIVERSITY OF LUCKNOW, LUCKNOW.

2022 – 23

SENTIMENT ANALYSIS ON MOVIE REVIEWS

An Internship Report Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

by

Akash Agarwal

(Univ. no. 2110014325030)

Under the Guidance of

Mr. Ankit Singh

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

FACULTY OF ENGINEERING & TECHNOLOGY,

UNIVERSITY OF LUCKNOW, LUCKNOW.

2022 – 23

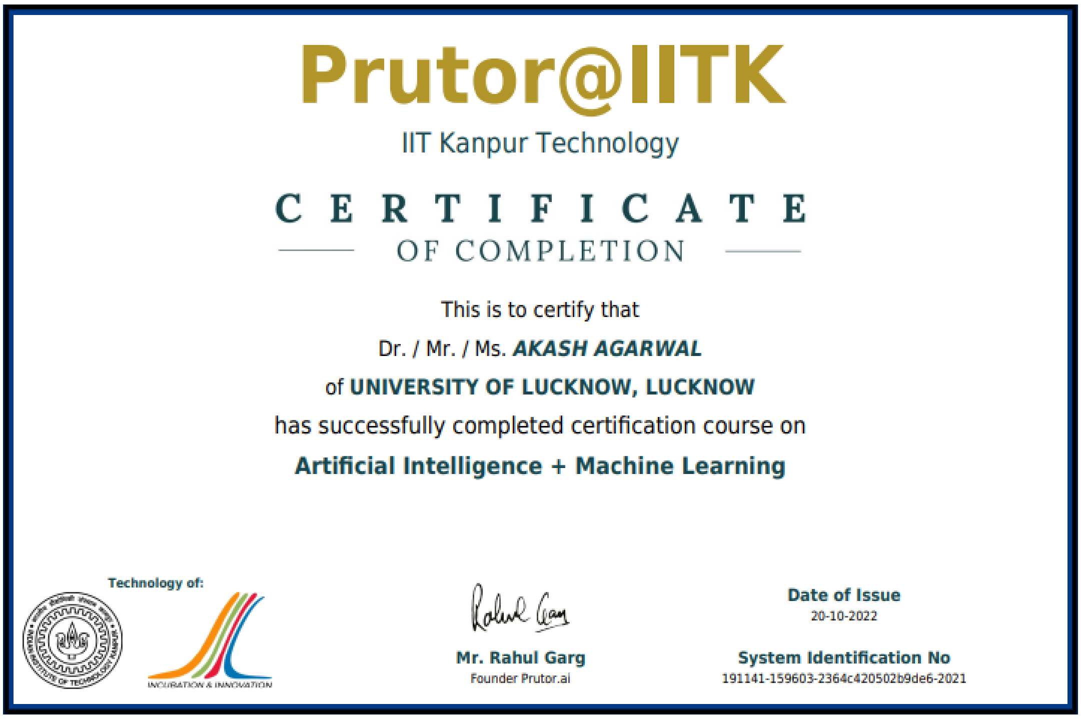
DECLARATION

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief. It contains no material previously published or written by another person or material which is substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher education, except where due acknowledgement has been made in the text.

Akash Agarwal
Univ. no. 2110014325030

Date:

CERTIFICATE



Prutor@IITK

IIT Kanpur Technology

CERTIFICATE

OF COMPLETION

This is to certify that

Dr. / Mr. / Ms. ***AKASH AGARWAL***

of **UNIVERSITY OF LUCKNOW, LUCKNOW**

has successfully completed certification course on

Artificial Intelligence + Machine Learning

Technology of:

Rahul Garg

Mr. Rahul Garg

Founder Prutor.ai

Date of Issue

20-10-2022

System Identification No

191141-159603-2364c420502b9de6-2021

ABSTRACT

Sentiment Analysis or Opinion mining on movie reviews. Sentiment is a thought, view, or attitude, especially one based mainly on emotion instead of reason for this we need to classify each and every review, this can be done through classifiers such as NAÏVE BAYES, SVM, K-NN, we preferred NAÏVE BAYES classifier, since it is a probabilistic approach, we depend upon each word's probability to be positive and negative, through this we will classify the review to be one of those. After classifying each review, the sentiment is drawn based on the count of the classes. Consider 50000 reviews, in those reviews there will be positive as well as negative reviews of that particular movie, the review is purely based on the public opinions being expressed in the portal. After classifying the reviews, we go for the max count method to predict the outcome. This algorithm is proposed in order to avoid the fake reviewing system followed now a days to promote the movie by the producers and crew of the movie. This gives the straight opinion of the public.

Keywords—Sentiment Analysis; Opinion Mining; Naïve Bayes; Probabilistic approach.

ACKNOWLEDGEMENT

I offer sincere thanks and humble regards to, **Prutor@IITK** for imparting us very valuable professional training in MCA. I pay my gratitude and sincere thanks to **Mr. Ankit Singh**, my subject matter expert for giving me the cream of his knowledge, I am grateful to him as he has been a constant source of advice, motivation and inspiration.

I am also thankful to him for giving his suggestions and encouragement throughout the project work. I take the opportunity to express gratitude and thanks to our computer labs staff for providing me opportunity to utilize the available resources for the completion of the project.

Finally, my thanks and appreciation go to my collage's young and dynamic teacher, **Er. Zeeshan Ali Siddiqui**, who irrespective of the situation, always encouraged and supported me to prepare this project report.

Akash Agarwal

Table of Contents

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Context

Sentiment analysis is a series of methods, techniques, and tools used to detect and extract subjective information, such as opinion and attitudes, from language. Traditionally, sentiment analysis has been about opinion polarity, i.e., whether someone has positive, neutral, or negative opinion towards something.

1.2 Motivation

The interest on other’s opinion is probably almost as old as verbal communication itself. Historically, leaders have been intrigued with the opinions of their subordinates to either prepare for opposition or to increase their popularity. Sentiment analysis is a well-known task in the realm of natural language processing. The objective is to determine the polarity of that text. The sentiments can consist of different classes.

1.3 Objective

We consider two cases:

1) A movie review is positive (+) or negative (-).

2) A movie review is very negative (- -), somewhat negative (-), neutral (o), somewhat positive (+), or very positive (+ +).

Positive Sentiment in subjective sentence: "I really love the movie BAAHUBALI"—This sentence expresses positive sentiment about the movie BAAHUBALI and we can tell that from the sentiment threshold value of word "love". So, threshold value of word "love" has positive numerical threshold value. Negative sentiment in subjective sentences: "MISTER is a disaster" this sentence expresses negative sentiment about the movie named "MISTER" and we can decide that from the sentiment threshold value of word "disaster". So, threshold value of word "disaster" has negative numerical threshold value. Sentiment Analysis is of three different types: Document level, Sentence level and Entity level. However we are studying phrase level sentiment analysis. The traditional text mining concentrates on analysis of facts whereas opinion mining deals with the attitudes. The main fields of research are sentiment classification, feature based sentiment classification and opinion mining.

1.4 Scope

Now, the use of sentiment analysis in a commercial environment is growing. This is evident in the increasing number of brand tracking and marketing companies offering this service. Some services include: - Tracking users and non-users opinions and ratings This is both advantageous to the producers and consumers. How in the sense is the producer get the exact and accurate review from the consumers and the consumers use the other consumers' reviews to use the product similarly in the movie industry it is both use full to the movie goers and the crew of the movie.

CHAPTER 2

PROJECT MODULES

2.1 Dataset Collection

The aggregation of review data set is a critical step in building artificial intelligence (AI). Movie review data sets are used in various ways including training and/or testing algorithms. 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.

2.2 Data Pre-processing

Data pre-processing or data cleansing is a crucial step and most of the ML engineers spend a good amount of time in data pre-processing before building the model. 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. While this produced better models, it also led to over-fitting on training examples and worse performance on the test set.

2.3 Cleansing of Dataset

One necessary pre-processing step prior to feature extraction was removal of HTML tags like “<br>”. 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. 2.4 Feature Extraction

We used N-gram modeling wherein we created unigrams, bigrams and mixture of both. While creating unigrams is more or less similar to the bag of words approach, bigrams provided more contextual information on the review text.. Also, to get more insight on textual information we created a feature set using a mixture of n-gram with n = 5 and using only those grams with minimum count of 10. In case of n-gram modeling, we did not remove the stopwords.

2.5 Model Creation System

Modeling in machine learning is an iterative phase where a data scientist continually train and test machine learning models to discover the best one for the given task.

2.6 Model training and visualization

Training a model simply means learning (determining) good values for all the weights and the bias from label examples. Model Visualization provides reason and logic behind to enable the accountability and transparency on the model.

2.7 Model Persistence System

All the code related to storing the model in a particular format that we can use later and it will handle all the code to load the model.

2.8 View Display Manager

View Display Manager is used to manage frontend of the application.

2.9 Prediction System

In this module we will use stored model to predict the results that is weather the review is positive or negative.

2.10 Timeline Visualization

It will show the data stored in tables in the form of pie chart. Here we can see overall report of the movie Pie chart will display percentage of like and dislikes.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The Sentiment Analysis System for Movie Review which is developed implements the Naive Bayes Classifier method in classifying the movie review documents into two classes: positive sentiment class and negative sentiment class. The flowchart regarding the overview of the Sentiment Analysis System for Movie Review is illustrated in Figure 3.1.

3.2 Algorithm

The explanation on Figure 3.1 is: firstly, the derived from the movie review is submitted by the user one by one. The submitted review may be categorized into training data or test data based on the user's wishes. The system, then, executes the document extraction process in order to generate a collection of terms from each review text and so-called n-gram. The next step is to train the system. At this step, the system constructs a model of the training data by calculating the occurrences of each term in the training data and likelihood value. The result of the system training process is a table of feature sets containing the number of occurrences and likelihood values of each term in the training data. The likelihood value is, then, incorporated into the next step which is the calculation of the system accuracy

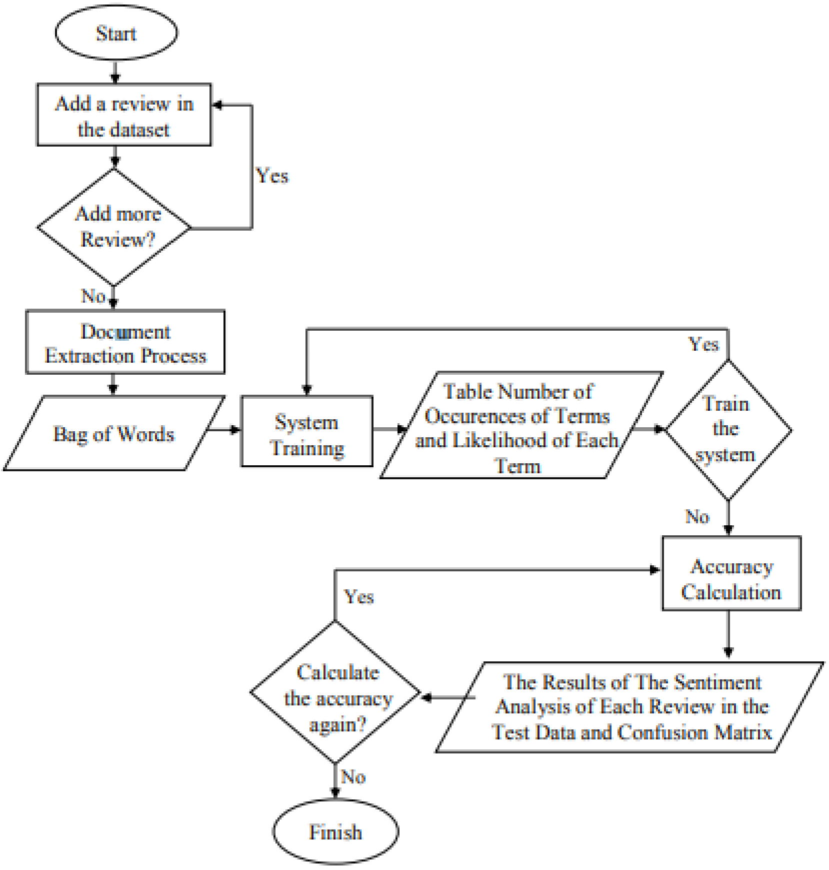

Start
Add a review in the dataset
Add more Review?
Yes
No
Document Extraction Process
Bag of Words
System Training
Table Number of Occurences of Terms and Likelihood of Each Term
Train the system
Yes
No
Accuracy Calculation
The Results of The Sentiment Analysis of Each Review in the Test Data and Confusion Matrix
Calculate the accuracy again?
Yes
No
Finish

Fig. 3.1 Overview of Sentiment Analysis

The system performs the classification process using the Naive Bayes Classifier method in order to obtain the best class of sentiment of each review in the category of data train. The results of the accuracy calculation are the sentiments of the classification of each test data review and the confusion matrix table obtained from the original sentiment comparison of the test data review and the sentiment of the analysis result. System accuracy can be concluded from the confusion matrix. The planning steps which are implemented based on the waterfall method.

Naïve Bayes classifier partitions the text composed of the documents with highest probabilities. It is the optimized probabilistic technique here we drew the comparison with two another classification techniques. By comparing we conclude that Naïve Bayes out performs the other two techniques.

CHAPTER 4

REQUIREMENT ANALYSIS

System requirement is the ability of the system to meet the condition desired by the users. System requirement analysis is performed by grouping the needs into functional requirements and non-functional requirements.

4.1 Functional Requirements

The functional requirements of the Sentiment Analysis System for Movie Review are as follows:

4.1.1 Hosting Of Web Application

Being a web application this is necessary to host the app from server.

4.1.2 Upload Review

To predict and produce result it is required to submit movie review.

4.1.3 Prediction

This is most crucial step as whole project move around analysing review.

4.1.4 Installing Libraries

It is necessary to install all the required libraries.

4.1.5 Output Shown On Screen

We will see a Pie chart representing likes with pink colour and dislikes with blue colour.

4.2 Non-Functional Requirements

Meanwhile, the non-functional requirements of the Sentiment Analysis System for Movie Review are as follows:

4.2.1 Performance Requirements

Performance depends on network connection and its speed. High congestion on internet network will automatically decrease server response capability. In this case our website can take much to open. Otherwise there are no internal complexities in this website.

4.2.2 Scalability Requirements:

- Can be enhanced for availability in different expert.
- Videos and tips can be uploaded by experts for End Users.

4.2.3 Software Quality Attribute:

4.2.3.1 Maintainability

Software can be maintained according to future scope with a minimal loss.

4.2.3.2 Portability

System is portable and is easily compatible with operating system and Internet Explorer, Opera , Mozilla Firefox .

- The system shall be extremely portable via the USB drive.
- The system shall be easy to migrate or backed up via another USB drive.

4.2.3.3 Reliability

Since we use SQLite3 as back end which after uninstallation maintains records. It also adds up a feature of reliability in it.

4.2.3.4 Reusability

It can be reused in different application and added as a sub module.

4.2.3.5 Robustness

Usage of Python as a Advance programming language ensures itself it's robustness. Also as a developer site admin is responsible for ensuring it's robustness and security.

4.2.3.6 Usability

It has user friendly interface makes it's usage very easy going process.

- The system shall be easy to use
- The system shall be easy to learn
- The system shall user easy to locate buttons
- The system shall prompt customer with friend easy to read error messages.

4.2.3.7 Flexibility

It's future scope adds flexibility in it. More than one admin can be added.

4.2.3.8 Interoperability

Among different modules there is well compatibility and communication. Data flow between through different interface is also maintained.

4.2.4 Software Requirements

4.2.4.1 Client

Browser : Chrome, Mozilla Firefox, IE9 or above

4.2.4.2 Developer

Browser : Chrome, Mozilla Firefox, IE9 or above
Operating System : Windows 7 or above, Linux
IDE : Visual Code
Database : sqlite3
Interpreter : Python 3.7 or higher

4.2.4.3 Server Side Requirements

Interpreter : Python 3.6 or higher

Libraries : streamlit, pandas, numpy, plotly.express, matplotlib, fastai==1.0.61, PIL, sqlalchemy, wheel,base64,fpdf, datetime

Operating System : Windows 7 or above or Linux

4.2.5 Hardware Requirements

4.2.5.1 Client

Processor : Dual Core or above
RAM : 1 GB
Disk space : 2 GB
Monitor : 15”
Others : Keyboard, mouse, monitor

4.2.5.2 Developer

Processor : Quad Core or above
RAM : 12 GB
Disk space : 4 GB
Monitor : 15”
Others : Keyboard, mouse, monitor

4.2.5.3 Tools and Platform

Programming Language : Python 3.6

Python Libraries

streamlit, pandas, numpy, plotly.express, matplotlib, pillow, sqlalchemy == 1.4, FPDF, fastai==1.0.61, datetime, Image, flask_sqlalchemy

IDE

Visual Studio Code

CHAPTER 5

SYSTEM DESIGN

Spiral model is a combination of sequential and prototype model. This model is best used for large projects which involves continuous enhancements. There are specific activities which are done in one iteration (spiral) where the output is a small prototype of the large software. The same activities are then repeated for all the spirals till the entire software is build.

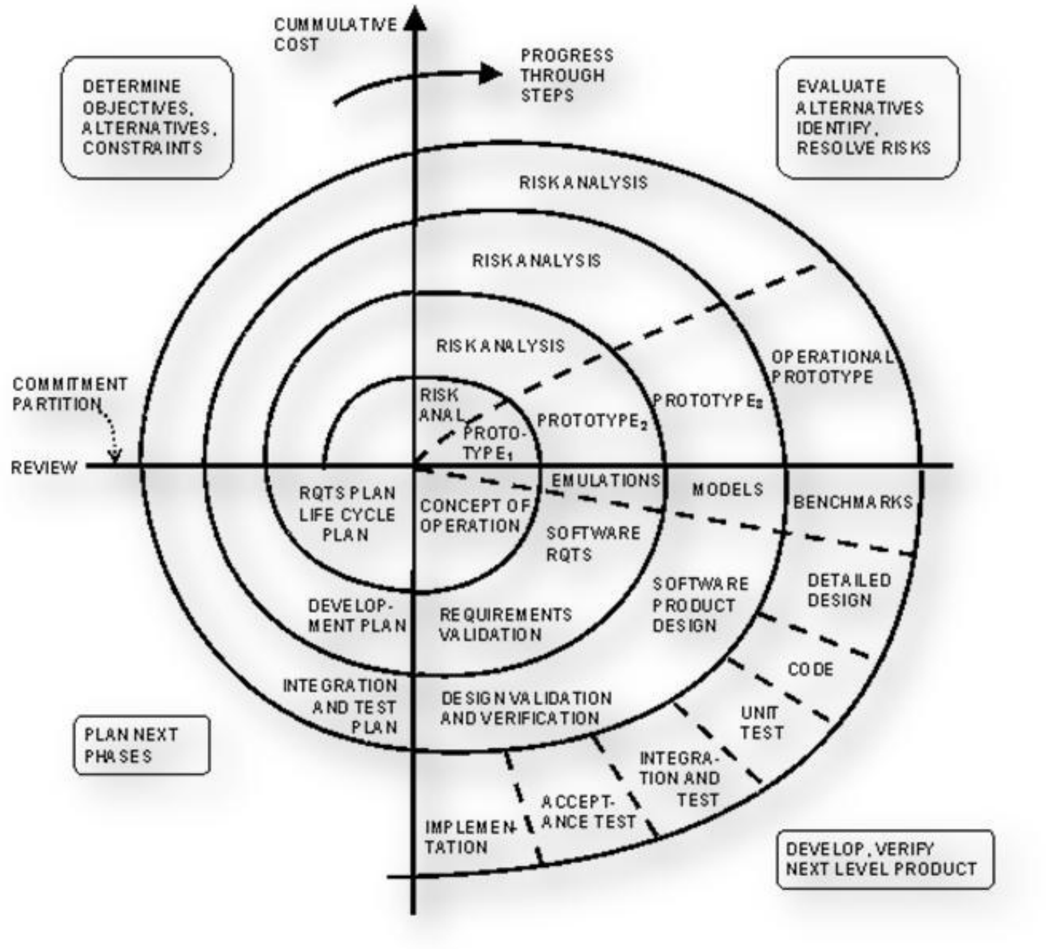


Fig. 5.1 Spiral Model

5.1 System Designing approaches

5.1.1 Top – Down designing

The top - down designing approach started with major components of the system. It is a stepwise refinement which starts from an abstract design, in each steps the design is refined two or more concrete levels until we reach a level where no – more refinement is possible or not needed.

5.1.2 Bottom – Up designing

In bottom – up designing the most basic and primitive components are designed first, and we proceed to higher level components. We work with layers of abstractions and abstraction are implemented until the stage is reached where the operations supported by the layer is complete.

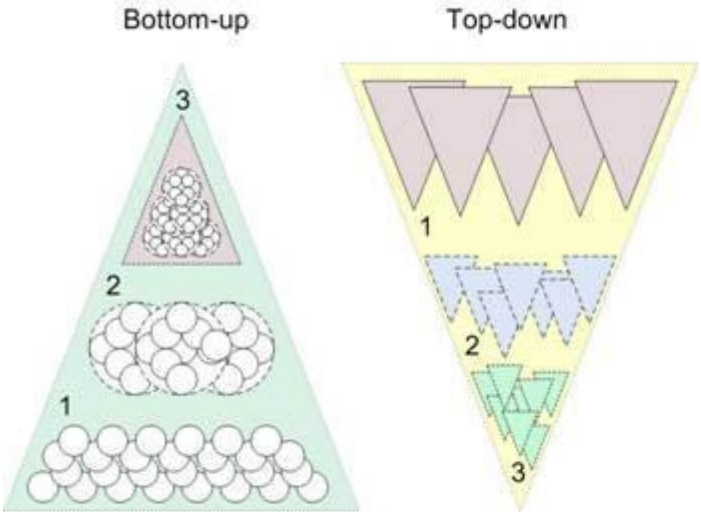


Fig. 5.2 Design Approach

5.2 Design Implementation and Constraints

Constraint 1: Minimum 1 GB of hard disk on user is required to setup the software.

Constraint 2: System with proper internet connection is required

Constraint 3: All libraries are required to be installed on user system.

The design of the Sentiment Analysis System for Movie Review uses Unified-Modeling Language (UML) as the modeling language and object-oriented programming concept. The documentation of the created system includes:

5.2.1 Business Process

Business process diagram describes a series of inputs, needs, and outputs processed by the system in order to produce the desired destination by the user. Description of business process system can be seen in Figure 3

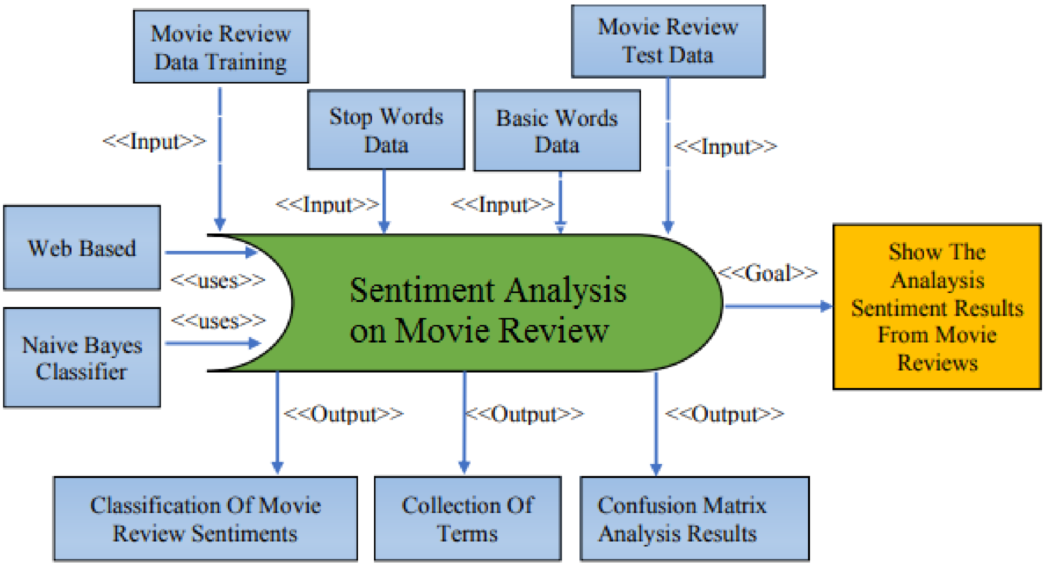


Fig. 5.3 Business process Sentiment Analysis on Movie Review

5.2.2 Use case Diagram

Use case diagram describes the system’s functional needs in the form of diagram. Use case is needed to describe typical interactions between users and system. Use case diagram of the Sentiment Analysis System for Movie Review is illustrated in Figure 5.4.

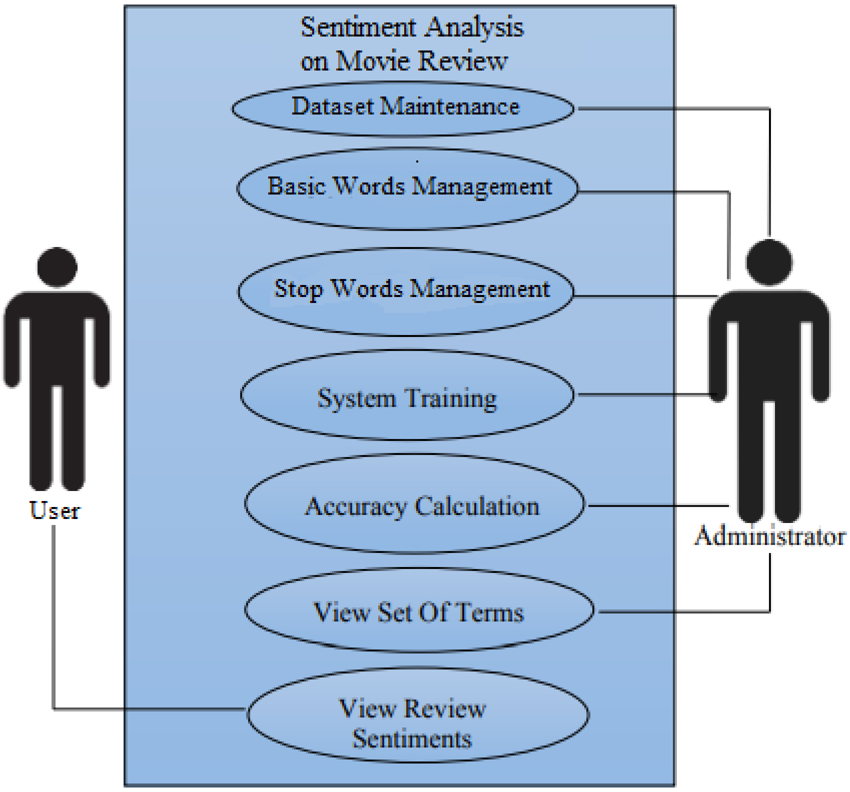


Fig. 5.4 Use Case Diagram Sentiment analysis on Movie reviews

5.2.3 Activity Diagram

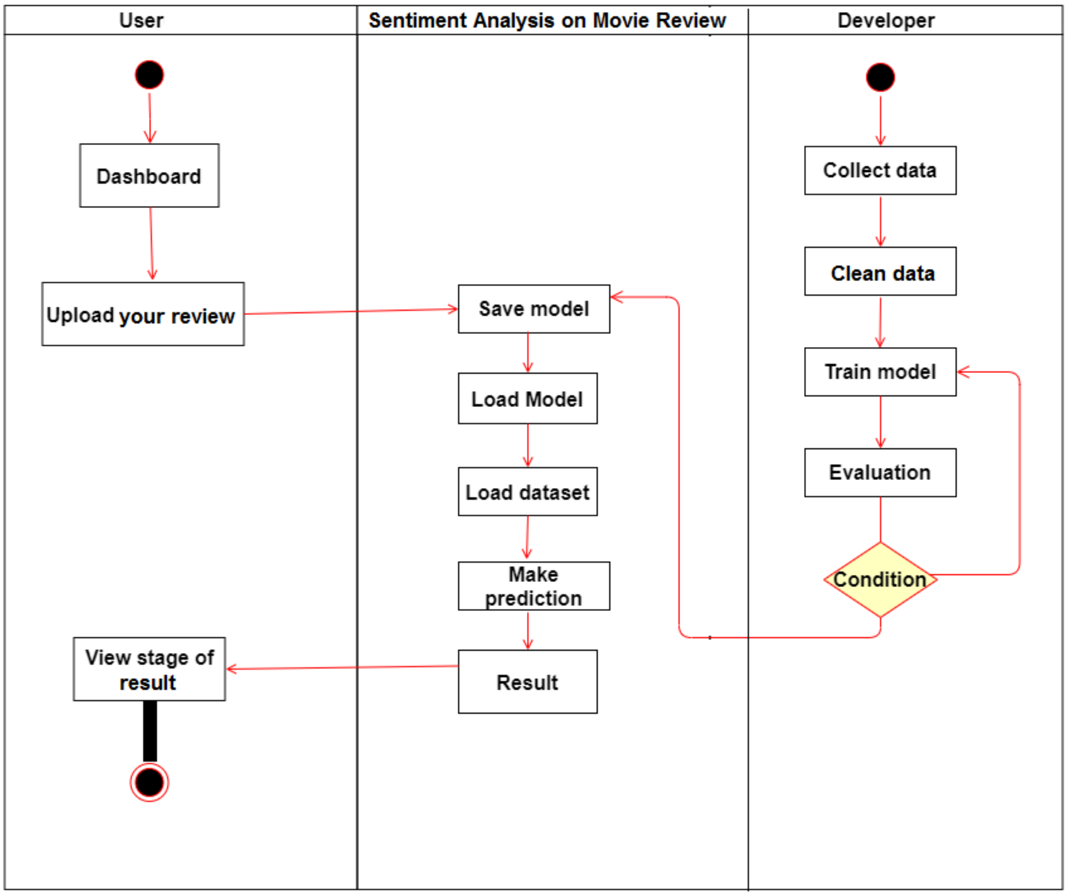


Fig. 5.5 Activity Diagram

5.2.4 Model Training

We have used SciKit Library to train our model, it has following features:

Simple and efficient tools for predictive data analysis

- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

5.2.5 Evaluation Criteria

To evaluate the efficacy of the model, the confusion matrix is estimated and gives an understanding of the proposed methodology and its potential for detailed classification. The classification model's usefulness and productivity were measured using the traditional metrics of accuracy, precision, and recall. Precision is the calculation of the model's correct predictions all over all predictions. The classification of reviews is termed as Accuracy, Sensitivity, Specificity, and F1-score are represented mathematically in terms of confusion matrix classification. The classification model's usefulness and productivity were measured using the traditional metrics of accuracy, precision, and recall. Precision is the calculation of the model's correct predictions all over all predictions. The classification of dislikes and Likes and between Dislikes and Likes is termed as Accuracy, Sensitivity, Specificity, and F1-score are represented mathematically in terms of confusion matrix

Table 5.1 Confusion Matrix

N=10000	Predict Dislike=5192	Predict Like=4808
Actual Dislike=4962	4387	575
Actual Like=5038	805	4233

5.2.6 True Positives (TP)

These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes.

5.2.7 True Negatives (TN)

These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. False positives and false negatives, these values occur when your actual class contradicts with the predicted class.

5.2.8 False Positives (FP)

When actual class is no and predicted class is yes.

5.2.9 False Negatives (FN)

When actual class is yes but predicted class in no.

5.2.10 Accuracy

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. For our model, we have got 0.86 which means our model is approx. 86.2% accurate.

Accuracy = TP+TN/TP+FP+FN+TN

5.2.11 Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. We have got 0.89 precision which is pretty good.

Precision = TP/TP+FP

5.2.12 Recall (Sensitivity)

Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes. We have got recall of 0.84 which is good for this model as it's above 0.5.

Recall = TP/TP+FN

5.2.13 F1 score

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall. In our case, F1 score is 0.86.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

5.2.14 Model Training Stats

Table 5.2 Model Training Stats

	precision	recall	f1-score	support
Dislikes	0.84	0.88	0.86	4962
Likes	0.88	0.84	0.86	5038
accuracy			0.86	10000
macro avg	0.86	0.86	0.86	10000
weighted avg	0.86	0.86	0.86	10000

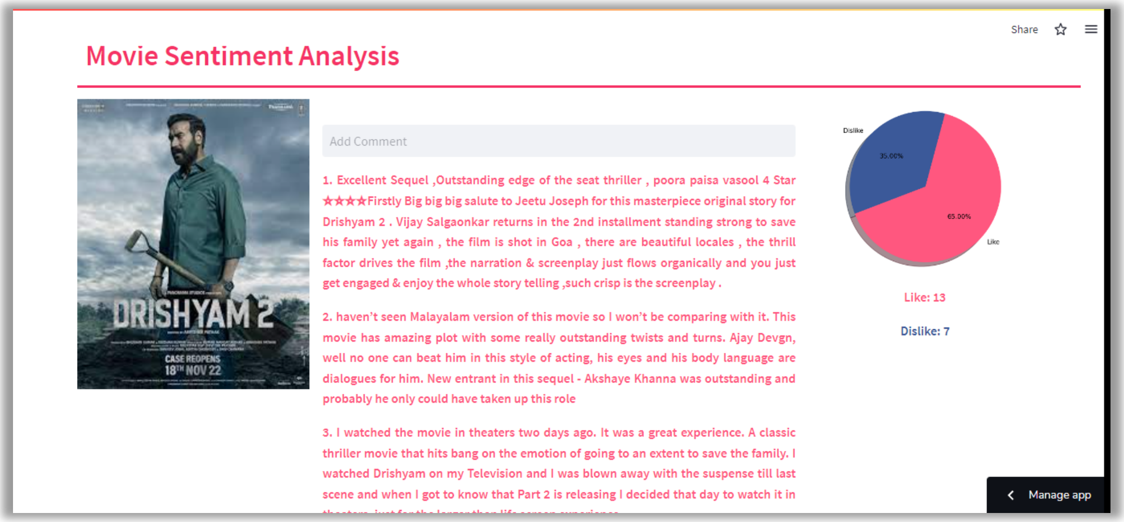
CHAPTER 6

IMPLEMENTATION

System Interface Implementation` The dashboard page is the only page once the administrator has successfully logged in to the system. The dashboard contains brief information on datasets, such as the number of review for each category, the ratio of training and test data as well as the ratio of positive and negative training data. The information is displayed in the form of numbers and graphs of pie charts. System training page contains a table on the occurrences and likelihood values of each term in all train data.

The visitor page is the page which is for the visitors. The visitors can access this page in order to enter the review data they want to know the sentiment of. The system classifies using the Naive Bayes method in order to obtain positive and negative sentiment probabilities as well as the sentiment result obtained based on the probability ratio of the two sentiment classes. The obtained result reflects back in the form of pie chart.

6.1 Screenshots



1. Excellent Sequel ,Outstanding edge of the seat thriller , poora paisa vasool 4 Star ★★★★Firstly Big big big salute to Jeetu Joseph for this masterpiece original story for Drishyam 2 . Vijay Salgaonkar returns in the 2nd installment standing strong to save his family yet again , the film is shot in Goa , there are beautiful locales , the thrill factor drives the film ,the narration & screenplay just flows organically and you just get engaged & enjoy the whole story telling ,such crisp is the screenplay .

2. haven't seen Malayalam version of this movie so I won't be comparing with it. This movie has amazing plot with some really outstanding twists and turns. Ajay Devgn, well no one can beat him in this style of acting, his eyes and his body language are dialogues for him. New entrant in this sequel - Akshaye Khanna was outstanding and probably he only could have taken up this role

3. I watched the movie in theaters two days ago. It was a great experience. A classic thriller movie that hits bang on the emotion of going to an extent to save the family. I watched Drishyam on my Television and I was blown away with the suspense till last scene and when I got to know that Part 2 is releasing I decided that day to watch it in theaters, just for the larger than life screen experience

Fig. 6.1 Full Screen

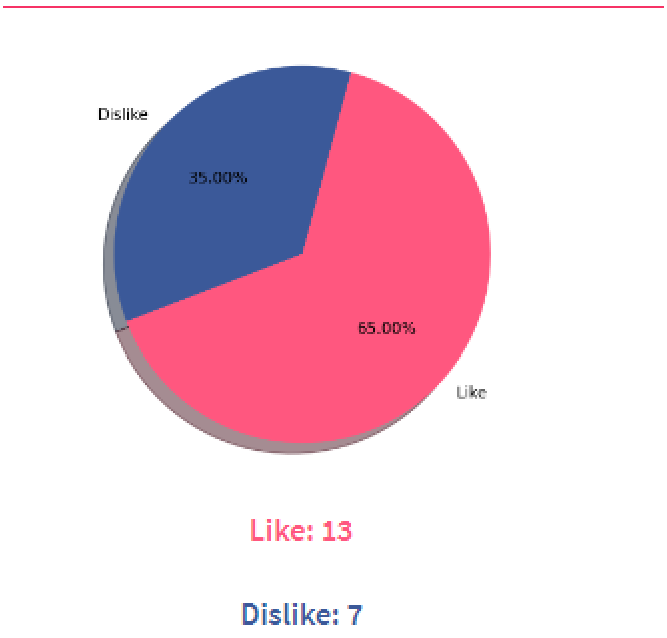


Fig. 6.2 Pie Chart

Fig. 6.3 Movie Poster

6.2 CODING

6.2.1 app.py

This code is used to implement the software. Here we are deploying our application using Streamlit. In this code file we are loading our model file and predicting the polarity of a comment and representing it in the form of pie chart.

```
import streamlit as st
import re
import pickle
from config import*
```

```
import matplotlib.pyplot as plt
def clean(x):
    #x=re.sub(r'\W',' ',x)
    #x = re.sub(r'[^a-zA-Z]',' ',x)
    x = re.sub("wouldn\'t",'would not',x)
    x = re.sub("they \ 've",'they have',x)

    #to remove html tags
    x = re.sub(r'<.*?>', '', x)

    #to remove everything except alpha
    x = re.sub(r'[^a-zA-Z]',' ',x)

    x = re.sub(r'\s+',' ',x)          #remove extra space's
    return x.lower()
st.set_page_config(layout="wide")
choice=st.cache()
#st.title(PROJECT_NAME)
with st.container():
    st.write('<style>body .sticky{ font-family: sans-serif;border-style: } .header{border-bottom-style: solid;padding-left:10px; padding-right: 800px;z-index: 1; background: White; color: #F63366; position:fixed;top:20px;} .sticky { position: fixed;top: 20; } </style><div class="header" id="myHeader"><h2 style="color: #F63366;"><b>'+"Movie Sentiment Analysis"+'</b></h2></div>', unsafe_allow_html=True)
with st.container():
   img_col=st.columns((1,2,1))
   img_col[0].image("tinyurl.1223",use_column_width=True, clamp=True)
   inp=img_col[1].text_input("",placeholder="Add Comment")
   a=0
if inp:
    coment.append(inp)
f=[]
for i in coment:
    d=clean(i)
    f.append(d)

with open("cv1.pkl",'rb') as f1:
    cv1=pickle.load(f1)
t=cv1.transform(f).toarray()
with open("nb_model.pkl",'rb') as f1:
    nb=pickle.load(f1)
pred=nb.predict(t)
ze=0
one=0
for i in pred:
    if i==0:
        ze+=1
    else:
        one+=1

cnt=[ze,one]
review= ['Dislike','Like']
colors=['#3b5999','#FF577F']
for i in coment:
    if pred[a]==0:
        a+=1
        img_col[1].markdown("<p style='text-align: justify; color:
#3b5999;'><b>"+str(a)+". "+i+"</b></p>", unsafe_allow_html=True)
    else:
        a+=1
        img_col[1].markdown("<p style='text-align: justify; color:
```

```
#FF577F;'><b>"+str(a)+". "+i+"</b></p>", unsafe_allow_html=True)

fig1, ax1 = plt.subplots()
ax1.pie(cnt, labels=review,colors=colors, autopct='%2.2f%%',
        shadow=True, startangle=75)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
img_col[2].pyplot(fig1)
img_col[2].markdown("<p style='text-align: center; color: #FF577F;'><b> Like: "+str(one)+"</b></p><p style='text-align: center; color: #3b5999;'><b> Dislike: "+str(ze)+"</b></p></div>", unsafe_allow_html=True)
```

6.2.2 Project.ipynb

This code file is concerned with all the work related with model training and saving model. We start with nltk library to download stopwords and removing 'not' from the stopwords. This is followed by loading all the dataset and dividing it in two parts. for training and testing. Data is cleaned and trained. After testing model is saved in pickle file.

```
### Project : Sentiments classification on movies review dataset
import numpy as np
import re
import pickle
#import nltk
#from nltk.corpus import stopwords
from sklearn.datasets import load_files
import pandas as pd
#load_files:Load text files with categories as subfolder names.
### clean data
```

```
def clean(x):
    #x=re.sub(r'\W',' ',x)
    #x = re.sub(r'[^a-zA-Z]',' ',x)
    x = re.sub("wouldn\'t",'would not',x)
    x = re.sub("they \ 've",'they have',x)

    #to remove html tags
    x = re.sub(r'<.*?>', '', x)

    #to remove everything except alpha
    x = re.sub(r'[^a-zA-Z]',' ',x)



    x = re.sub(r'\s+',' ',x)          #remove extra space's
    return x.lower()

#\W:matches any non-alphanumeric character;
#this is equivalent to the set [^a-zA-Z0-9_].
### Convert text into numeric


from nltk.corpus import stopwords
import nltk
words = stopwords.words('english')

if 'not' in words:
    words.remove('not')
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
#cv = CountVectorizer(min_df=10,max_df=.6,stop_words=words)
cv = CountVectorizer(stop_words=words)

#min_df=10:exclude any word that comes in 10 or less than 10 documents
#max_df=.6:excude any word that comes more than 60% of the documents,

from sklearn.naive_bayes import MultinomialNB
nb = MultinomialNB()
### Lets work on 50000 samples
```

```
df = pd.read_csv('movie_reviews.csv')
df.head()
df.shape()
df.sentiment.value_counts()
df['review'] = df.review.apply(clean)
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(df.review.values,df.sentiment.values,
                                test_size=10000,
                                random_state=10)
x_train.shape
np.bincount(y_test)
cv1 = CountVectorizer(stop_words=words)
#x_new=cv1.fit_transform(x_train).toarray()
#x_test_new = cv1.transform(x_test).toarray()

x_new=cv1.fit_transform(x_train)
x_test_new = cv1.transform(x_test)
x_new[0].toarray()
x_new.shape
x_test_new.shape
nb = MultinomialNB()
nb.fit(x_new,y_train)
nb.score(x_test_new,y_test)
test=["I do not like this movie",
      "This movie is adorable",
     "I hate this movie",
      "I love this movie"]
f=[]
for i in test:
    s=clean(i)
    f.append(s)
t=cv1.transform(f)
from sklearn.metrics import confusion_matrix
z=cv1.transform(x_test)
y_pred = nb.predict(z)
confusion_matrix(y_test, y_pred)

### save the model

with open('nb_model.pkl','wb') as f1:
    pickle.dump(nb,f1)
### save the vectorizer
with open('cv1.pkl','wb') as f1:
    pickle.dump(cv1,f1)
### load model
with open('nb_model.pkl','rb') as f1:
    clf=pickle.load(f1)
```

```
clf.predict(t)
```

Output: array([0, 1, 0, 1], dtype=int64)

CHAPTER 7

TESTING METHODOLOGY

This module is tested by the following testing methods
Unit Testing
Integration Testing

7.1 Unit Testing

Unit testing is a procedure used to validate that individual units of source code are working properly. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual program, function, procedure, etc., while in object-oriented programming, the smallest unit is a method, which may belong to a base/super class, abstract class or derived/child class.

Ideally, each test case is independent from the others; mock or fake objects as well as test harnesses can be used to assist testing a module in isolation. Unit testing is typically done by software developers to ensure that the code they have written meets software requirements and behaves as the developer intended.

Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit and how to use it can look at the unit tests to gain a basic understanding of the unit API.

Unit test cases embody characteristics that are critical to the success of the unit. These characteristics can indicate appropriate/inappropriate use of a unit as well as negative behaviors that are to be trapped by the unit. A unit test case, in and of itself, documents these critical characteristics, although many software development environments do not rely solely upon code to document the product in development.

On the other hand, ordinary narrative documentation is more susceptible to drifting from the implementation of the program and will thus become outdated (e.g. design changes, feature creep, relaxed practices to keep documents up to date).

7.2 Integration Testing:

It is sometimes called I&T i.e. Integration and testing, it is the phase of software testing in which individual software modules are combined and tested as a group. It follows unit testing and precedes system testing.

Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items. These "design items", i.e. assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test that all components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e. unit testing

CONCLUSION

The Sentiment Analysis System for Movie Reviews implements the method of Naive Bayes Classifier in order to obtain the highest posterior probability value of the two review classes of sentiment. The posterior probability value is obtained from the total sum of log prior probability and log likelihood of each term in each review of the training data for each sentiment class. The average accuracy produced by the Sentiment Analysis System for Movie Review is 86.20% out of testing processes.

FUTURE SCOPE

This study tried to cover most aspects of sentiment analysis, but there is still work to be done. Most importantly, there is a need for more data for reviews, which could help improve the accuracy of the model. At present, there is a significant difference in the reviews for the first level of classification.

This model can help in the first level of classification to determine whether the person has positive or negative view for movie. Although sentiment sensitivity was not within the scope of this study, future work in detecting the sensitivity of the review can also be an important improvement in the already-existing model.

REFERENCE

1. Mohana Pranadeep potti, ManneDineshKmar, Nagabhyrava Saswanth Ram,P.V.R.Sandeep P.R.Krishna Prasad, 2018, Sentiment Analysis On Movie Reviews Using NAÏVE BAYES Classifier.

2. Rizky S 2011 Konsep Dasar Rekayasa Perangkat Lunak (Software Engineering) Jakarta: Prestasi Pustaka