A

Mini Project

On

OPINION MINING ON FEEDBACK MANAGEMENT SYSTEM

(Submitted in partial fulfillment of the requirements for the award of degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

BY

J.KRISHNA MOHAN(187R1A05F1)

Under the Guidance of

K. RANJITH REDDY

(Assistant professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

2018-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**OPINION MINING ON FEEDBACK MANAGEMENT SYSTEM**" being submitted by **J. KRISHNA MOHAN (187R1A05F1)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him under our guidance and supervision during the year 2021-22.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Mr. K.RANJITH REDDY
Assistant Professor
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on _____

ACKNOWLEGDEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. I take this opportunity to express my profound gratitude and deep regard to my guide

Mr. K. RANJITH REDDY, Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which I am about to embark.

I also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Dr. **M. Varaprasad Rao, Mr. J. Narasimha Rao, Dr. T. S. Mastan Rao, Dr. Suwarna Gothane, Mr. A. Uday Kiran, Mr. A. Kiran Kumar, Mrs. G. Latha** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

I am also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

I am obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. I would like to express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. I am grateful for their constant support and help.

Finally, I would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. I sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

J. KRISHNA MOHAN (187R1A05F1)

ABSTRACT

In today's environment where we're suffering from data overload, companies have mountains of customer feedback collected. But for mere humans, it's still impossible to analyze it manually without any sort of error or bias. Often, companies with the best intentions find themselves in an insights vacuum for accurate decision making. Sentiment analysis (a form of text analytics) measures the attitude of the customer towards the aspects of a service or product which they describe in text. It typically involves taking a piece of text, whether it's a sentence, a comment or an entire document, summarizing it into a few actionable insights and returning a "score" that measures how positive or negative the text is. Companies want their brand being perceived positively. When looking to decrease customer churn, we use sentiment analysis to focus on the comments where the sentiment is strongly negative. Likewise, we look at customer comments which have a strong positive sentiment to find out why these customers love us, and try and focus on what we can do as a business to increase the Revenue / Profits / Brand value.

LIST OF FIGURES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1.INTRODUCTION

1.INTRODUCTION

1.1 PROJECT SCOPE

This project is titled as “Opinion Minning On Feedback Management System”. This software provides facility to upload the dataset and get to know about the sentiment of the review. This project uses machine learning methodologies to classify the reviews based on given dataset. We use convolutional neural networks to classify each review from the dataset given.

1.2 PROJECT PURPOSE

This has been developed to facilitate the extracting the features of the customer reviews and their opinions. Opinions of different features from customer feedback are classified and summarized by a feature-based opinion mining technique. This information will be more helpful to make decisions for better improvement of product quality.

1.3 PROJECT FEATURES

The main feature of this project is to perform the aspect-based sentiment analysis on the given reviews using CNN algorithm. The CNN is a deep learning algorithm that is employed for hierarchal textual classification. It uses five different filter sizes to each review(comment) and GlobalMaxPolling1D layers are applied to each consecutive layer. The outputs are all connected to the final dense layer. The output of this project is to classify the given review into three categories namely positive, negative and neutral based on the polarity value generated foe each review by the CNN model.

2.SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, "what must be done to solve the problem?" The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

Motivated by different real-world applications, researchers have considered a wide range of problems over a variety of different types of corpora. We now examine the key concepts involved in these problems. This discussion also serves as a loose grouping of the major problems, where each group consists of problems that are suitable for similar treatment as learning tasks. One set of problems share the following general character: given an opinionated piece of text, wherein it is assumed that the overall opinion in it is about one single issue or item, classify the opinion as falling under one of two opposing sentiment polarities, or locate its position on the continuum between these two Polarities.

2.2 EXISTING SYSTEM

In the existing system the text classification methods and algorithms are used which are not that much accurate in classification. In general, the existing systems classification are KNN, ANN and SVM due to their low accuracy dependency of quality of classifications which are not efficient.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

- More classification.
- Time consuming.

To avoid all these limitations and make the working more accurately the system needs to be implemented efficiently.

2.3 PROPOSED SYSTEM

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides higher accuracy and reduces the classification work. The existing system has several disadvantages and many more difficulties to work well. The proposed system tries to eliminate or reduce these difficulties up to some extent. The proposed system helps the user to work user friendly and he can easily do his jobs without time lagging.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features

- Ensure data accuracy's.
- Minimum time needed for the various processing.
- Greater efficiency.
- Better service.
- User friendliness and interactive.
- Minimum time required.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company.

Three key considerations involved in the feasibility analysis

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.3.2 ECONOMIC FEASIBILITY

Development of this application is highly economically feasible. The organization needed not spend much money for the development of the system already available. The only thing is to be done is making an environment for the development with an effective supervision. If we are doing so, we can attain the maximum usability of the corresponding resources. Even after the development, the organization will not be in condition to invest more in the organization. Therefore, the system is economically feasible.

2.3.3 TECHNICAL FEASIBILITY

We can strongly say that it is technically feasible, since there will not be much difficulty in getting required resources for the development and maintaining the system as well. All the resources needed for the development of the software as well as the maintenance of the same is available in the organization here we are utilizing the resources which are available already.

2.3.4 BEHAVIORAL FEASIBILITY

Whatever we think need not be feasible. It is wise to think about the feasibility of any problem we undertake. Feasibility is the study of impact, which happens in the organization by the development of a system. The impact can be either positive or negative. When the positives nominate the negatives, then the system is considered feasible. Here the feasibility study can be performed in two ways such as technical feasibility and Economical Feasibility.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor : Intel Dual Core@ CPU 2.90GHz.
- Hard disk : 16GB and Above.
- RAM : 4GB and Above.

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements.

- Operating system : Windows 8,10
- Languages : Python
- Backend : Machine Learning
- IDE : PyCharm

3.ARCHITECTURE

3.ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for sentiment analysis of given data using machine learning, starting from input to final prediction.

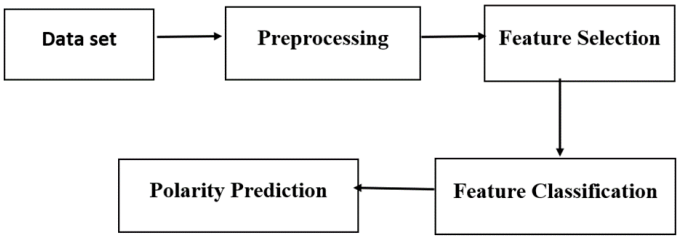


Figure 3.1: Project Architecture of Opinion Mining on Feedback Management System

3.2 DESCRIPTION

The project has been classified into five modules (or stages) in a sequential order that the Input textual review has to go through to remodel itself into a valuable Net numeric Score that proves to be an asset in commercial business. This modular approach of the project is shown below sequentially.

Module 1: DATA ACQUISITION (Gathering and Cleaning of Data)

- Initially the reviews either in the form of textual comments or a multifaceted data set containing such numerous textual reviews is collected and given to the machine using input devices. This could be generalized by saying that the data generated by an organization externally is ingested into the our machine for evaluation and prediction of valuable insights. Legal, privacy and compliance issues must be taken into consideration, particularly with respect to permitted use of data sets from Web portals.

- In our project we are using .csv dataset file from Amazon-an online E-commerce platform which holds around 5,000 unbiased reviews from all its customers. This data set is uploaded to the application for further evaluation.

Module 2: DATA PRE-PROCESSING (Transformation, Normalization and Elimination)

Transforming text into another form, which an algorithm can perceive involves Pre-processing steps:

- **Noisy Entity Elimination:** Sanitizing the raw data directly obtained from the users is a crucial and an initial pre-processing technique to be followed by removing unnecessary and irrelevant entities for further Text Mining. NLP libraries have the capability to remove 200Kb of noise from given data.
- **Tokenization:** Breaking a character string into pieces, called as Tokens, and throwing away the certain characters at the same time, including punctuation and special characters.

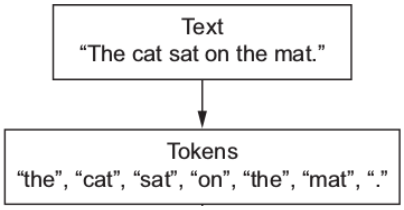


Figure 3.2: Tokenization of a phrases

- **Stemming** (Rule Based Approach): **Stemming** algorithms typically cut off the end or the beginning of the word (prefix or suffix) into its Stem word which is usually doesn't have a meaning in the dictionary.
- **Lemmatization** (Dictionary-based approach): Lemmatization algorithms cut off the word into its base form ie; Lemma with the use of vocabulary and morphological analysis of the words. These lemmas have a defined meaning in the dictionary.

Module 3: TRAINING AND TESTING DATA

To estimate the accuracy of the model, we have split the categorical data into Train and Test sets. A training set is to accommodate and tune our model whereas a testing set is to create predictions on and to evaluate the model precisely.

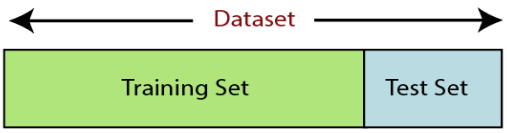


Figure3.2: Splitting of Data Set

Here's a snippet of us the code which shows that 12% of the data accorded to Testing data set and the remaining 88% to the Training data set.

In the training process, the model learns to equate a particular input (i.e. textual review) to the corresponding output based on the test samples used for the process of Training. The feature extractor transfers the text input into a feature vector. Pairs of feature vectors and tags (e.g. positive, negative, or neutral) are fed into the machine learning algorithm to generate a model.

Module 4: CLASSIFICATION ALGORITHM

Popular Deep learning class that is employed for Hierarchical Textual classification is Convolutional Neural Networks (CNN) .

Defining CNN

- Text as a sequence is passed to a **CNN LSTM Classifier - Long Short Term Memory,** special type of Recursive Neural Network RNN which is capable of using long memory as the input of activation functions in the hidden layer.
- Embedding Matrix is given to embedding layer.
- Five different filter sizes are applied to each comment and GlobalMaxPooling1D layers are applied to each consecutive layer.
- The outputs are all concatenated together.

- A Dropout layer then dense then Dropout followed by the final Dense layer is applied.

Training CNN

The number of epochs is the amount to which our model will loop around and learn. The batch size is the amount of data which the model will see at given period of time. As we are training on small data set in just a few epochs out model will over fit.

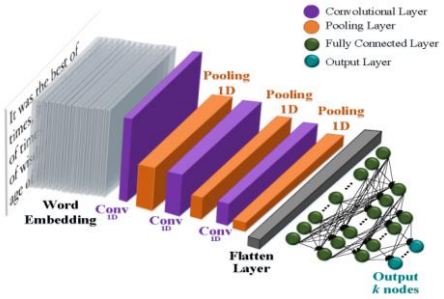


Figure 3.2: Layers in Convolutional Neural Networks

Here is a code snippet on building a CNN classifier using Keras.

- The fforemost layer is the embedded layer that uses 100 length vectors to represent each word.
- SpatialDropout1D performs variational dropout in Natural Language Processing.
- The next layer is the LSTM layer with 100 memory units.
- **Softmax** is the Activation function is applied to the fully connected layer for Multi-class classification. [8]It “squashes” the output from each neuron to be equally a number between 0 and 1. For instance, a number of 6.77 could be “**squashed**” to 0.32.It also divides each “squashed” value from the vector to sum up to a total of 1. The new vector contains the probabilities that the data belongs to a specific class.

Module 5: SENTIMENT PREDICTION (DATA VIEW):

USER GRAPHICAL INTERFACE:

For User Interface, we have made use of Tkinter,which is a standard Graphical User Interface library in Python. User friendly GUI components including Buttons, tkMessage box and Frames have been used as shown below:

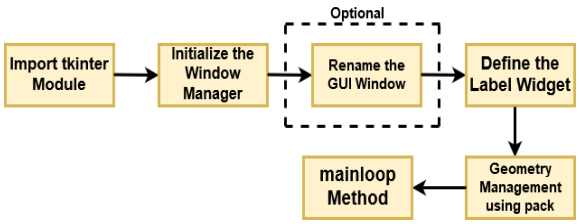


Figure 3.2: Initiation of Tkinter module

- **Button**: Button has a property for switching on/off. When a user clicks on the button, an event is triggered in the Tkinter GUI.

button_widget = tk.Button(widget, option)

- **Canvas**: Canvas is used to draw various shapes in GUI which makes the look and feel of the application more appealing.

canvas_widget = tk.Canvas(widget)

- **Label**: Label is used to create a single line component for entering textual input and images.

label_widget = tk.Label(widget, option)

3.3 USE CASE DIAGRAM

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.

An actor is represents a user or another system that will interact with the system you are modeling. A use case is an external view of the system that represents some action the user might perform in order to complete a task.

Contents:

- Use cases
- Actors
- Dependency, Generalization, and association relationships
- System boundary

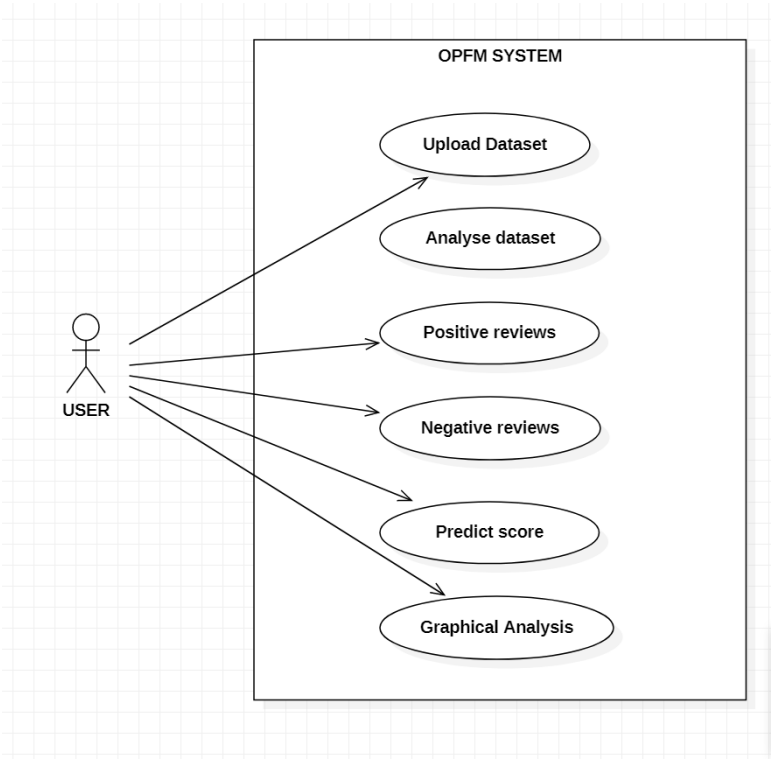


Figure 3.3 : Use case Diagram for Opinion Mining on Feedback Management System

3.4 CLASS DIAGRAM

Class diagrams are widely used to describe the types of objects in a system and their relationships. Class diagrams model class structure and contents using design elements such as classes, packages and objects. Class diagrams describe three different perspectives when designing a system, conceptual, specification, and implementation. These perspectives become evident as the diagram is created and help solidify the design. Class diagrams are arguably the most used UML diagram type. It is the main building block of any object oriented solution. It shows the classes in a system, attributes and operations of each class and the relationship between each class.

In most modeling tools a class has three parts, name at the top, attributes in the middle and operations or methods at the bottom. In large systems with many classes related classes are grouped together to create class diagrams. Different relationships between diagrams are show by different types of Arrows. Below is a image of a class diagram. Follow the link for more class diagram examples.

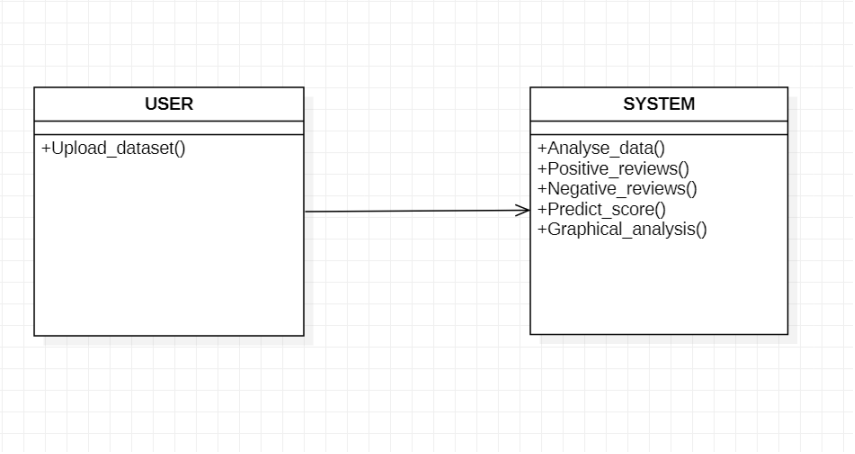


Figure 3.4: Class Diagram for Opinion Mining on Feedback Management System

3.5 SEQUENCE DIAGRAM

Sequence diagrams in UML shows how object interact with each other and the order those interactions occur. It's important to note that they show the interactions for a particular scenario. The processes are represented vertically and interactions are show as arrows. This article explains the purpose and the basics of Sequence diagrams.

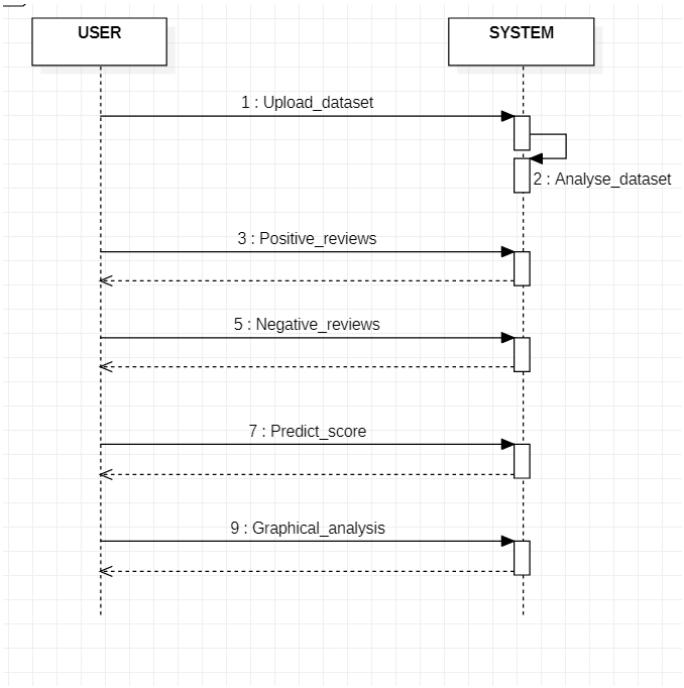


Figure 3.5: Sequence Diagram for Opinion Mining on Feedback Management System

3.6 ACTIVITY DIAGRAM

It describes about flow of activity states.

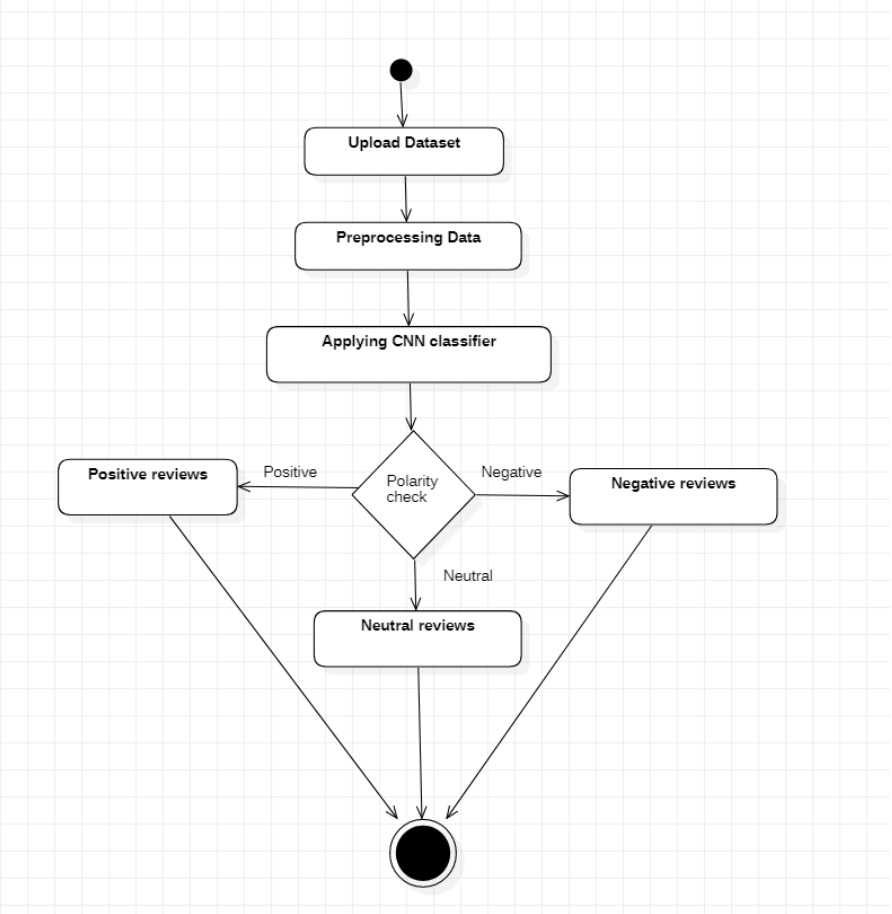


Figure 3.6: Activity Diagram for Opinion Mining on Feedback Management System

4.IMPLEMENTATION

4. IMPLEMENTATION

4.1 SAMPLE CODE

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import os
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
import base64
from emoji import UNICODE_EMOJI
from sklearn.feature_extraction.text import CountVectorizer
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
import re
from keras.models import Sequential
from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D
from sklearn.model_selection import train_test_split
os.environ["PYTHONIOENCODING"] = "utf-8"

main = Tk()
main.call('encoding', 'system', 'utf-8')
main.title("OPINION MINING FOR FEEDBACK MANAGEMENT SYSTEM")
main.geometry("1300x1200")
sid = SentimentIntensityAnalyzer()

#Preprocessing
global X
global Y
X = []
Y = []
```

```
text.delete("1.0",END)
train = pd.read_csv(filename,encoding="utf8")
count = 0
for i in range(len(train)):
    sentiment = train._get_value(i,0,takeable = True)
    tweet = train.get_value(i, 1, takeable=True)
    tweet = tweet.lower()
    icon = train.get_value(i, 2, takeable=True)
    if str(icon) != 'nan':
        icon = UNICODE_EMOJI[icon]
        icon = ''.join(re.sub('[^A-Za-z\s]+', '', icon))
        icon = icon.lower()
    else:
        icon = ''
        arr = tweet.split(" ")
        msg = ''
        for k in range(len(arr)):
            word = arr[k].strip()
            if len(word) > 2 and word not in stop_words:
     msg += word + " "
     textdata = msg.strip() + " " + icon
    X.append(textdata)
    count = count + len(arr)
    X = np.asarray(X)
    Y = pd.get_dummies(train['sentiment']).values

#generateModel
    global XX
    global tokenizer
    global X_train, X_test, Y_train, Y_test
    max_fatures = 2000
    tokenizer = Tokenizer(num_words=max_fatures, split=' ')
    tokenizer.fit_on_texts(X)
    XX = tokenizer.texts_to_sequences(X)
    XX = pad_sequences(XX)
```

```
    X_train, X_test, Y_train, Y_test = train_test_split(XX, Y, test_size=0.13, random_state=42)

#buildClassifier
    global model
    embed_dim = 128
    lstm_out = 196
    max_fatures = 2000
    model = Sequential()
    model.add(Embedding(max_fatures, embed_dim, input_length=XX.shape[1]))
    model.add(SpatialDropout1D(0.4))
    model.add(LSTM(lstm_out, dropout=0.2, recurrent_dropout=0.2))
    model.add(Dense(3, activation='softmax'))
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    batch_size = 32
    model.fit(X_train, Y_train, epochs=7, batch_size=batch_size, verbose=2)

def module1():
    text.delete('1.0', END)
    # print(base64.b64encode(tf1.get().encode("utf-8")))
    # name = bytes(tf1.get(),"unicode_escape")
    # print(name)
    sentence = tf1.get()
    sentiment_dict = model.predict(sentence)[0]
    negative = sentiment_dict['neg']
    positive = sentiment_dict['pos']
    neutral = sentiment_dict['neu']
    compound = sentiment_dict['compound']
    text.insert(END, "Input Sentence : " + sentence + "\n\n")
    text.insert(END, "Positive : " + str(positive) + "\n")
    text.insert(END, "Negative : " + str(negative) + "\n")
    text.insert(END, "Neutral : " + str(neutral) + "\n")
    text.insert(END, "Compound : " + str(compound) + "\n")
    result = ''
    if compound >= 0.05:
        result = 'Positive'
    elif compound <= - 0.05:
```

```
        result = 'Negative'
    else:
        result = 'Neutral'
    pos, neg, neu = getCount(sentence)
    text.insert(END, sentence + ' CLASSIFIED AS ' + result + "\n\n")
    text.insert(END, 'Positive Words : ' + str(pos) + "\n")
    text.insert(END, 'Negative Words : ' + str(neg) + "\n")
    text.insert(END, 'Neutral Words  : ' + str(neu) + "\n")
    tf1.delete(0, 'end')
    height = [len(pos), len(neg), len(neu)]
    bars = ('Positive Words', 'Negative Words', 'Neutral Words')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, height)
    plt.xticks(y_pos, bars)
    plt.show()
def getCount(sentence):
    pos = []
    neg = []
    neu = []
    arr = sentence.split(' ')
    for i in range(len(arr)):
        word = arr[i].strip()
        if (sid.polarity_scores(word)['compound']) >= 0.3:
            pos.append(word)
        elif (sid.polarity_scores(word)['compound']) <= -0.5:
            neg.append(word)
        else:
            neu.append(word)
    return pos, neg, neu

global filename
global sentence
global positive_mean
global negative_mean
global positive_reviews
global negative_reviews
```

```
def positiveReviews():
    global positive_reviews
    global positive_mean
    positive_reviews = 0
    text.delete('1.0', END)
    query = tf2.get()
    train = pd.read_csv(filename)
    print(train.head)
    count = 0
    total = 0
    cols = train.shape[1]
    print(cols)
    m = 0
    n = 0
    if cols == 2:
        m = 0
        n = 1
    else:
        m = 1
        n = 16
    print(m,n)
    for i in range(len(train)):
        name = str(train._get_value(i, m, takeable=True))
        sentence = str(train._get_value(i, n, takeable=True))
        #print(name,sentence)
        if query in name:
            total = total + 1
            sentiment_dict = model.predict(sentence)[0]
            compound = sentiment_dict['compound']
            positive = sentiment_dict['pos']
            if compound >= 0.05:
                text.insert(END, str(sentence.encode('UTF-8')) + " == " + str(positive) + "\n")
                count = count + 1
    positive_reviews = count
    positive_mean = count / total
    text.insert(END, "\n\nTotal Reviews    : " + str(total) + "\n")
```

```
    text.insert(END, "Positive Reviews : " + str(count) + "\n\n\n")
    text.see(END)
def negativeReviews():
    global negative_reviews
    global negative_mean
    negative_reviews = 0
    text.delete('1.0', END)
    query = tf2.get()
    train = pd.read_csv(filename, encoding='utf8')
    count = 0
    total = 0
    cols = train.shape[1]
    m = 0
    n = 0
    if cols == 2:
        m = 0
        n = 1
    else:
        m = 1
        n = 16
    print(str(m) + " " + str(n))
    for i in range(len(train)):
        name = str(train._get_value(i, m, takeable=True))
        sentence = str(train._get_value(i, n, takeable=True))
        if query in name:
            total = total + 1
            sentiment_dict = model.predict(sentence)[0]
            compound = sentiment_dict['compound']
            negative = sentiment_dict['neg']
            if compound <= - 0.05:
                text.insert(END, str(sentence.encode('UTF-8')) + " == " + str(negative) + "\n")
                count = count + 1
    negative_reviews = count
    negative_mean = count / total
    text.insert(END, "\n\nTotal Reviews    : " + str(total) + "\n")
```

```
    text.insert(END, "Negative Reviews     : " + str(count) + "\n\n\n")
    text.see(END)
def productScore():
    text.delete('1.0', END)
    text.insert(END, "Mean Positivity : " + str(positive_mean) + "\n\n")
    text.insert(END, "Mean Negativity : " + str(negative_mean) + "\n\n")
def upload():
    global filename
    filename = askopenfilename(initialdir="dataset")
    text.insert('1.0', filename)
    #text.delete('1.0', END)
def graph():
    height = [positive_reviews, negative_reviews]
    bars = ('Total Positive Reviews', 'Total Negative Reviews')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, height)
    plt.xticks(y_pos, bars)
    plt.show()
font = ('times', 15, 'bold')
title = Label(main, text='OPINION MINING FOR FEEDBACK MANAGEMENT SYSTEM')
title.config(bg='mint cream', fg='olive drab')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0, y=5)
font1 = ('times', 14, 'bold')
ff = ('times', 12, 'bold')
l1 = Label(main, text='Enter A Comment:')
l1.config(font=font1)
l1.place(x=50, y=100)
tf1 = Entry(main, width=40)
tf1.config(font=font1)
tf1.place(x=230, y=100)
runButton = Button(main, text="Run", command=module1)
runButton.place(x=330, y=150)
runButton.config(font=ff)
```

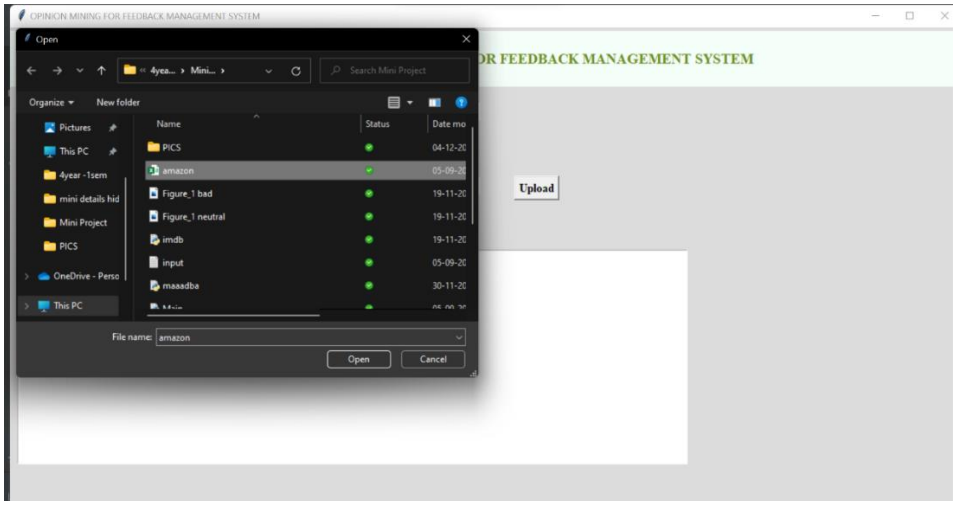
```
l2 = Label(main, text='Upload Dataset:')
l2.config(font=font1)
l2.place(x=50, y=200)
tf2 = Entry(main, width=40)
tf2.config(font=font1)
tf2.place(x=230, y=200)
uploadButton = Button(main, text="Upload", command=upload)
uploadButton.place(x=680, y=200)
uploadButton.config(font=ff)
positiveButton = Button(main, text="Positive Reviews", command=positiveReviews)
positiveButton.place(x=10, y=250)
positiveButton.config(font=ff)
negativeButton = Button(main, text="Negative Reviews", command=negativeReviews)
negativeButton.place(x=150, y=250)
negativeButton.config(font=ff)
productButton = Button(main, text="Product Score", command=productScore)
productButton.place(x=310, y=250)
productButton.config(font=ff)
graphButton = Button(main, text="Graphical Analysis", command=graph)
graphButton.place(x=450, y=250)
graphButton.config(font=ff)
font1 = ('times', 13, 'bold')
text = Text(main, height=15, width=100)
scroll = Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=10, y=300)
text.config(font=font1)
main.config(bg='gainsboro')
main.mainloop()
```

5.SCREENSHOTS

To run project double click on ‘Mini_senti.py’ file to get below screen

Screenshot 5.1 Application GUI

In above screen click on ‘Upload ’ button to upload sentiment dataset.

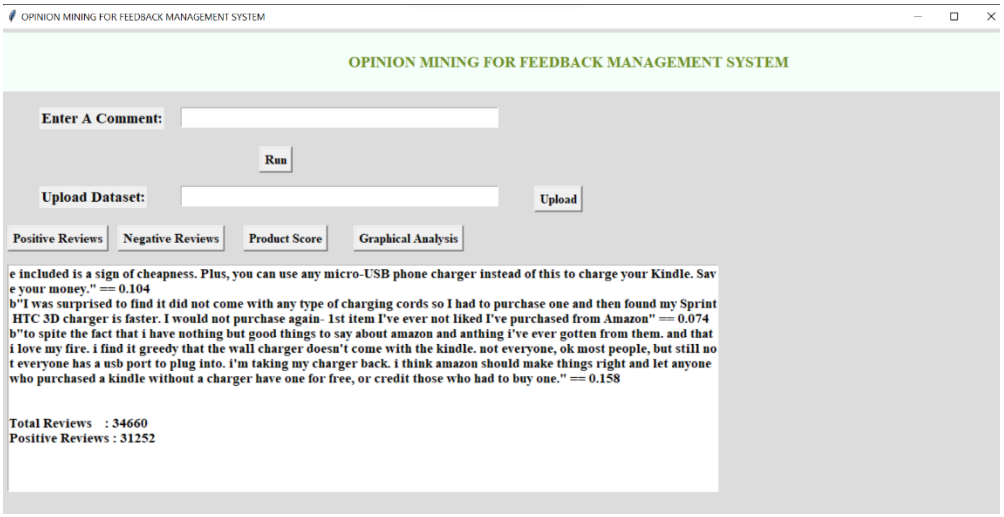


Screenshot 5.2 Upload dataset

In above screen I am uploading ‘amazon.csv’ file and after upload will get below screen

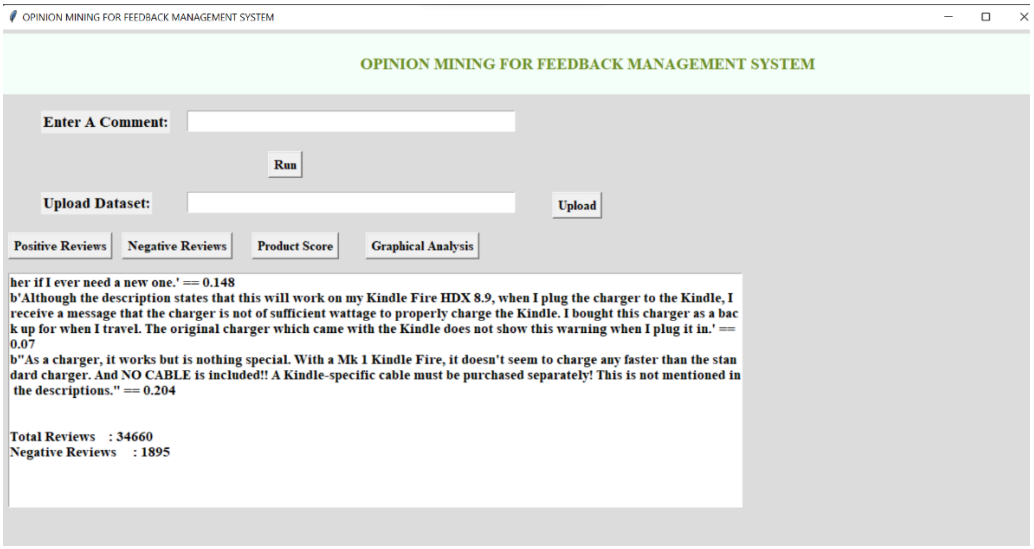
Screenshot 5.3 Dataset uploaded

Now click on “positive Reviews” button to generate the summary of positive reviews present in the dataset.



Screenshot 5.4 Positive reviews

In the above screen total 34,660 reviews are generated upon 31,252 reviews are classified as positive. Click on “Negative Reviews” button to generate the summary of negative reviews present in the dataset.

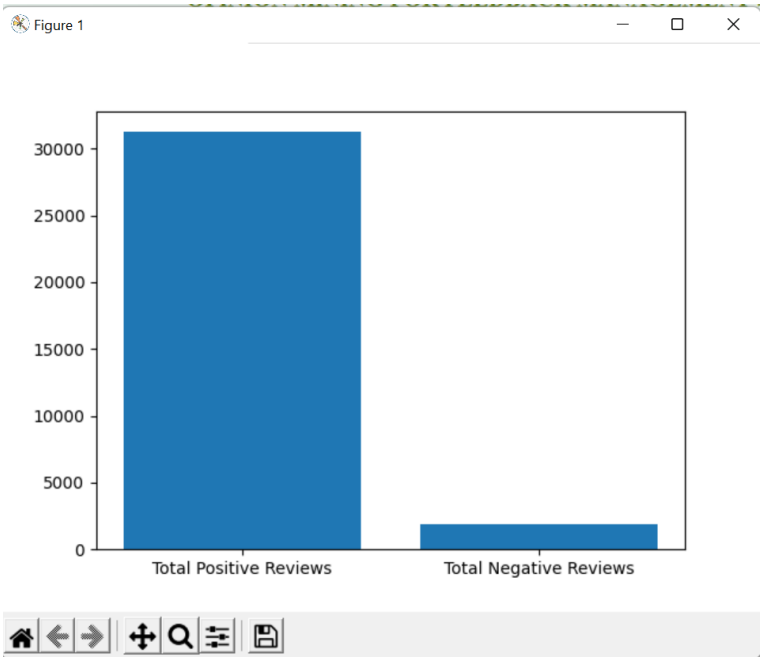


Screenshot 5.5 Negative reviews

In the above screen total 34,660 reviews are generated upon 1,895 reviews are classified as negative. Click on the “Product score” button to generate the summary of mean positive and negative score of dataset.

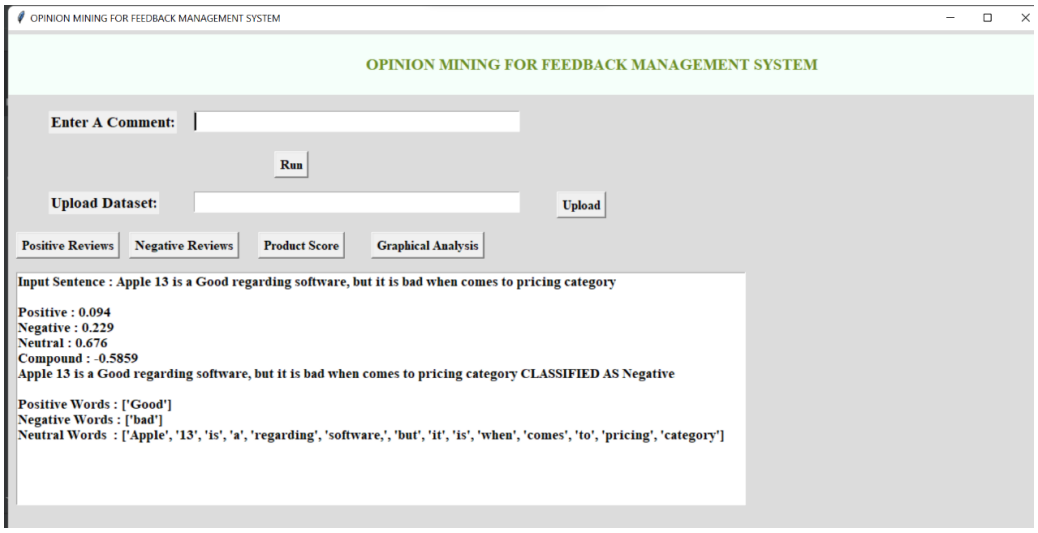
Screenshot 5.6 Product score

In the above screen the mean positivity and mean negativity is shown. Click on the “Graphical Analysis” button to generate the graphical representation of the results obtained.



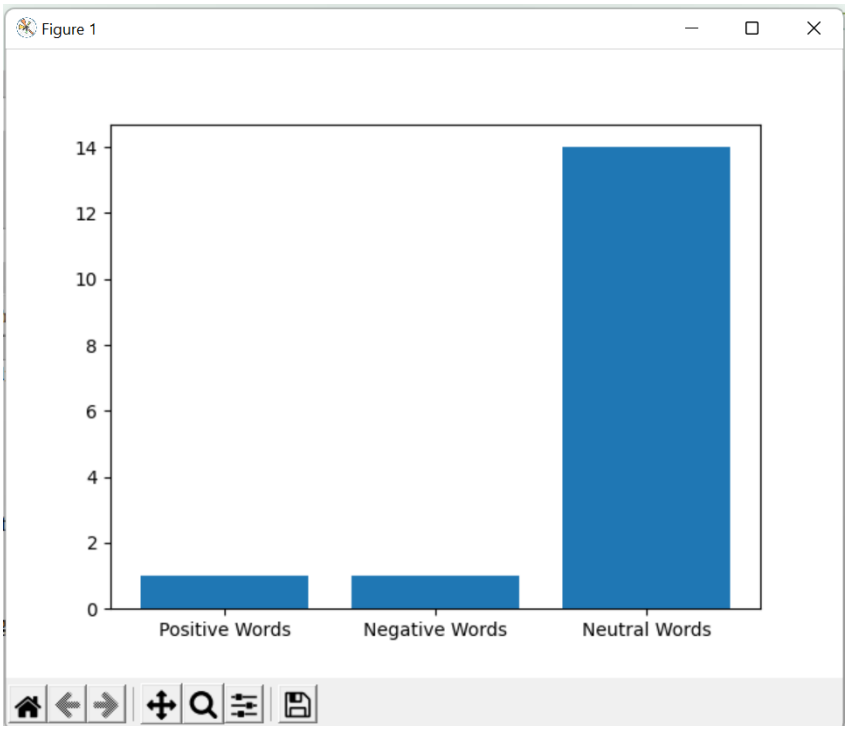
Screenshot 5.7 Graphical Analysis 1

In the above screen we can see the graphical representation of the positive reviews and negative reviews. Now enter a comment in comment text box and press “Run” button.



Screenshot 5.8 Comment Classification

In the above screen the entered comment has been classified into three different categories as positive, negative and neutral.



Screenshot 5.9 Graphical Analysis 2

The graphical representation of classified words in the comment is shown in the above screen.

6.TESTING

6.TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent.

Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONALTESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes.

6.3 TESTCASES

6.3.1 UPLOADING DATASET

Test case ID	Test case name	Purpose	Test Case	Output
1	User uploads dataset	Use it for classification	The user uploads the product review dataset	Uploaded successfully
2	User uploads another dataset	Use it for classification	The user uploads the a non-product review dataset	Uploaded successfully

6.3.2 CLASSIFICATION

Test case ID	Test case name	Purpose	Input	Output
1	Classification test 1	To check if the classifier performs its task	Apple is a good smart phone	Classified as a positive review
2	Classification test 2	To check if the classifier performs its task	Apple battery life is not good	Classified as a negative review
3	Classification test 3	To check if the classifier performs its task	Nokia phone durability is very good	Classified as a positive review.

7.CONCLUSION

7.CONCLUSION & FUTURESCOPE

7.1 PROJECT CONCLUSION

In this project, we presented a machine-assisted literature review, using text classification with Convolutional Neural Networks (CNN). With the rising number of textual comments and reviews, there is almost a near to impossible task to read, analyze, and predict valuable insights. A tool that can go through all the comments given to it, compute their polarities based on their degree of Positivity or Negativity and categorize them into a relevant class, evince to be a powerful weapon in the Recommendation area. We were capable enough to tackle the underlying notion of a conglomerate of firms that rely on Customer Reviews of their products and services to foster computation of Sentiment Polarity and thereby Class-Categorization. We identified that numerous Data Analysis methods could be accompanied in this process namely:

1. Machine learning,
2. Natural language Processing; and
3. Sentiment Computation using specific predictive methodologies.

Multi-domain Product reviews data set from Amazon.com an Online E-commerce Platform was used for this study. Demonstrations for both Sentence-level categorization and Review-level categorization have been evaluated. From the vendor's point of view, a good Recommendation System for assessing speculative data holds a greater marketing asset.

Our project outcome indicates that the CNN-LSTM classifier technique used has yielded about 90.9% accuracy. We believe that further making use of bigger and more diversified datasets of customer reviews for Testing and Training will increase the scope and functionality of our application.

7.2 PROJECT FUTURESCOPE

The Research scope in Text classification and Sentiment analysis would be boundless with changing times:

Negation Handling: Negation plays a crucial role in Lexical Analysis. Sentences such as: “I like this watch” and “I don’t like this watch” are very similar, but due to the negation term, these phrases are get categorized into the same class instead of separate classes. These make the predictions ambiguous.	**Multi-Lingual Feedback Analysis** The applicability of definite categorization on customer feedback in multiple languages is dubious. Firms working with global clients who take customer feedback needs to translate from diversified languages including minority dialects could be the scope of Sentiment Analysis.
Dealing with Irony Ironical expression of one’s meaning by that signifies the exact contrary to what had been said, typically for conveying a sense of sarcasm. This has become a common facet used by young millennials in contemporary digital terminology.	**Bi-polarized Opinions** As a part of Human tendency, we have binate opinions on Products and Service Providers. Being able to inspect and summarize the twofold sentiments requires precise and varied training of data sets.
signing and Generation of a Corpus database e predicted outcomes of numerous products must be tabulated and stored in Databases for the prospect of Business Analytics and to be able to understand Product Dynamics and Sales.	**Detecting fake and Spam opinions on Online Platforms** With increasing cyber usage, increases the number of spammers who tarnish the reputation of the vendors by deviating from the genuine numeric rating of the customers. This would require User-frequency determination on Web platforms.

8.BIBILOGRAPHY

8. BIBILOGRAPHY

8.1REFERENCES

[1] Nhan Cach Dang, María N. Moreno-García, and Fernando De la Prieta," Sentiment Analysis Based on Deep Learning: A Comparative Study", 14 March 2020, Department of Information Technology, HoChiMinh City University of Transport (UT-HCMC), Ho Chi Minh 70000, Vietnam.

[2] Jenq-Haur Wang Department of Computer Science and Information Engineering, "An LSTM Approach to Short Text Sentiment Classification with Word Embedding's", The 2018 Conference on Computational Linguistics and Speech Processing ROCLING 2018, pp. 214-223

[3] Aina Elisabeth Thunestveit, Supervisor: Herindrasana Ramampiaro, IDI, "Sentiment Analysis on User-Based Reviews: Movie Recommendation Case", June 2016, Master of Science in Informatics, Department of Computer and Information Science, Norwegian University of Science and Technology.

[4] White Paper – Sentiments Analysis By Rajaram Jagannathan

8.2 WEBSITES

[1] https://monkeylearn.com/sentiment-analysis/

[2]https://www.newgenapps.com/blog/artificial-intelligence-vs-machine-learning-vs-data-science/

[3] https://www.edureka.co/blog/python-libraries/

[4]https://medium.com/datadriveninvestor/artificial-intelligence-vs-machine-learning-vs-deep- learning-vs-data-science-2183ac856368

[5] https://www.slideshare.net/anilsth91/tweet-sentiment-analysis-78718965

[6]https://www.newgenapps.com/blog/the-secret-way-of-measuring-customer-emotions-social-media-sentiment-analysis/