A Project On

TEXT-BASED EMOTION DETECTION USING MACHINE LEARNING

Submitted in partial fulfillment of the course named

CS675 - Introduction to Machine Learning

at

Master of Science

In

Department of Computer Science

By

Chaitanya Karnati(ck338)

Abhinesh Goud Panjugula(ap2787)

ABSTRACT

Emotion Detection is an automated process capable of understanding the feelings or opinions that underlie a text. Emotion Detection is a machine learning tool that analyzes texts for polarity, from positive to negative. By training machine learning tools with examples of emotions in text, machines automatically learn how to detect sentiment without human input. Emotion Detection models can be trained to read beyond mere definitions, to understand things like, context, sarcasm, and misapplied words.

“I really like the new design of your website!” → Positive “I’m not sure if I like the new design” → Neutral

“The new design is awful!” → Negative.

INTRODUCTION

In the past few decades, there has been a huge increase in the usage of machines and they are been used in various industries. Along with rise in machine's exposure with human, the interaction has also become simpler and easier. To attain this, we should provide machines with the capability to understand their environment especially, with human beings. Generally, machines whenever referred, its term consists of robots and computers. The difference between both is that robots have more abilities in interaction.

Humans use their senses to understand their surrounding environment. Therefore, Machines are made in such a way that they aim to imitate such human senses to interact with their surroundings. These days, there are many ways in which the machines can record their surrounding environment through sensors and cameras. Using this recorded information withproper algorithms allows to create machine's point of view.

In Machine Learning, computer systems are programmed to learn from data that is input without being continually reprogrammed. Simply, they improve their performance in a task continuously. For example, playing a game; without any extra help from a human. Machine learning is now developing in being used for different kinds of fields such as art, science, finance, healthcare, etc. Machine learning was made possible not just by Arthur Samuel's breakthrough program in 1959—using a relatively simple (by today's standards) search tree as its main driver, his IBM computer continually improved at checkers—but by the Internet as well.

Deep learning is an artificial intelligence (AI) function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. For example, the recommendations on what to watch on Netflix are based on DL. In recent years, the use of Deep Learning algorithms has been very successful in this regard. For instance, Jeremy Howard showed in his Brussels 2014 TEDx talk how computers trained using deep learning techniques were able to achieve some amazing tasks.

Goal:

The main goal of this project is to analyze texts for polarity, from positive to negative and also to know the emotion they are feeling through texts. This helps to know human emotion with respect to the text they send, so that we can predict their condition and react accordingly.

With the help of training machine learning tools to imitate human emotions and feelings with examples in text, machines automatically learn how to detect sentiment and emotions without human input.

Objectives:

The objectives of text-based emotion detection are: -

- Create a dataset with text that are used.
- Then based on their text we give if their sentiment in the dataset.
- Based on that we write a code to implement testing emotions based on text.
- We detect the sentiments of the text in the output whether it is positive, negative,
  or neutral.

Methodology:

The research on Text-based emotion and sentiment detection are classified from many perspectives: the techniques that are used, detailed level of text analysis, point of view of the text, etc.

1. From a technical point of view, we identified machine learning, lexicon-based, statistical and rule-based approaches.
    - ML method uses various algorithms to discover the emotion and sentiment by training on a dataset.
    - Lexicon-based approach involves calculating emotion and sentiment polarity for its analysis using the semantic orientation of sentences and words in review. The “semantic orientation” is an estimation of judgement and individuality of the given text.

- Rule-based method is used to know about the opinion of words in a text and then categorize them based on the number of positive, negative, or neutral words. It examines different rules for classification such as negation words, mixed opinions, booster words, idioms, dictionary polarity, emoticons, mixed opinions, etc.
- The Statistical models presents each and every review as a mixture of latent aspects and ratings. It is assumed that aspects and their ratings can be represented by multinomial distributions and try to cluster head terms into aspects and sentiments into ratings.

2. Another classification is mostly based on the text and its structure: feature level or word/ sentence level classification, document level. Sentence-level or Word-level classification are used to convey sentimental polarity for every sentence and each word in a review, whereas Document-level classification is mainly used to know sentiment polarity for the. Researches prove that most of the methods focus mainly on Document- level classification.
3. We can differentiate the methods which calculate emotion and sentiment strength for various kinds of methods and products which rate a review. Maximum number of outputs which focus on global review classification consider only the polarity of the review and depend on the ML techniques. Solutions which demand a better and clear classification of reviews use more linguistic features that includes intensification, negation, modality and discourse structure.

ROLES AND RESPONSIBILITIES

Roles:

- Decision making support:
  It is very hard to build a website that performs decision making. Few of the advantages of analysis are: it gives us various ideas that help us in decision making in our day-to-day life for example, choosing a good novel, selecting a good movie, or buying a new gadget, etc.
- Business related application:
  As there is a change in market regularly, we can see that, the competition in co-operative world is rising constantly. Also, the need to invent an innovative and new product that can satisfy the customers has been increasing steadily. To gain

more importance for their products, organizations are assembling all the requirements of their users to improve the efficiency of their product from the feedback they receive from their customers.

• Predictions and trend analysis:

Tracing the feedback of people by sentiment and emotion analysis enables any person to predict the context of the market that helps a person in trading and

polling. By using this all-opinions user can predict the market trends.

Responsibilities:

Emotion and Sentiment analysis is very important as it helps businesses to quickly gain knowledge of the overall views given by their customers. By solving the emotions and sentiments that lies in the reviews, social media conversations, etc, we can take decisions faster and also make accurate decisions.

It is told that approximately 90% of the date in the world is unstructured and is unorganized. Large volumes of unstructured data are been created everyday through chats, social media, articles, surveys, emails, documents, etc. But it is tough to analyse emotion and sentiment in a timely and efficient manner.

The responsibilities of emotions and sentiment analysis are endless and can be applied to any industry, from finance and retail to hospitality and technology. Below, we've listed some of the most popular ways that sentiment analysis is being used in business:

- Social Media Monitoring
- Brand Monitoring
- Voice of customer (VoC)
- Customer Service
- Market Research

REQUIREMENT ENGINEERING

A Software Requirements Specification (SRS) – a requirements specification for a software may be a complete description of the behavior of a system to be developed. It

includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Nonfunctional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System Requirements Specification: A structured assortment of data that embodies the necessities of a system. A business analyst, typically titled system analyst, is answerable for analyzing the business wants of their shoppers and stakeholders to assist determine business issues and propose solutions. inside the systems development lifecycle domain, the BA generally performs a liaison perform between the business facet of AN enterprise and therefore the data technology department or external service suppliers.

Functional Requirements

Functional Requirements are the features that should be included in most of the systems to satisfy the customers and maintain their business needs accordingly.

In functional requirements we focus on the documenting the activities and operations that our project is supposed to perform and they include the following:

- The system can manage (add, read, update, and delete) movie review datasets, and can also manage (add, read, update, and delete) the dictionary of stop words.
- It can perform the classifier training process and display the model within the sort of feature sets of the term data from the training data.
- The system can able to display the test data result and display confusion matrix generated from the classifier testing.
- The system can display a set of movie review dataset terms derived from tokenizing, filtering, and stemming processes.
- The system can display analysis result derived from reviews submitted by users.

Interface Requirements:

- User Interface: - The user interface of this system is a user friendly Java Graphical User Interface.

- Hardware Interfaces: - The interaction between the user and the console is achieved through Java capabilities.
- Software Interfaces: - The required software is PYTHON
- Operating Environment: - Windows XP, Linux.

Non-Functional Requirements:

Non-Functional requirements are the explanation of characteristics, qualities, and features of the system as well as any constraint that limit the boundaries of the proposed system.
These are mostly based on economy, performance, information, security and controlled services that are in the proposed system.
In terms of non-functional requirements, we should be mainly focusing on the performance requirements of our system

- The system uses an authentication through login process so as to differentiate user level.
- The system can run in various web browsers which support the system environment,
- The system gives a fast response, and
- The system has a user-friendly interface design.

Software Requirements:

User Interface- The User Interface of this framework is an easy-to-understand Java Graphical User Interface. **Software Interfaces -** The required software is PYTHON

Packages: Some packages that are to be installed for this project are: -

1. **Pip install NumPy -** Def: NumPy intends to give an array or cluster object that is up to 50x quicker than customary Python records. The cluster or array object in NumPy is called ndarray, it gives a ton of supporting capacities that make working with ndarray simple. Arrays are habitually utilized in information science, where speed and assets are vital.
2. **Pip install keras** - Def: Keras is an incredible and simple to-utilize free open-source Python library for creating and assessing profound learning models. It wraps the effective mathematical calculation libraries Theano and TensorFlow

and permits you to characterize and prepare neural organization models in only a couple lines of code.

3. **Pip install sklearn** - Def: Scikit-learn is presumably the most valuable library for AI in Python. The sklearn library contains a ton of productive apparatuses or tools for AI and statistical demonstrating including characterization, regression, grouping and dimensionality reduction.
4. **Pip install Matplotlib** - Def: Matplotlib, it is a plotting library for the Python programming language and its mathematical math extension NumPy. It gives an object- oriented Programming interface to implanting plots into applications utilizing universally useful GUI tool stash like Tkinter, wxPython, Qt, or GTK.
5. **Pip install nltk** - Def: The Natural Language Toolbox (NLTK) is a stage utilized for building Python programs that work with human language information for applying in statistical natural language processing (NLP). It contains text handling libraries for tokenization, parsing, characterization, stemming, labeling and semantic thinking. NLTK is a main stage for building Python projects to work with human language information. Composed by the makers of NLTK, it manages the peruser through the basics or fundamentals of composing Python programs, working with corpora, sorting text, examining linguistic structure and more.

Hardware Requirements

1. Processor **-** Pentium –IV

In computers, a processor or preparing unit is an advanced circuit which performs a procedure on some outer information source, typically memory or some other information stream. It commonly appears as a microchip, which can be carried out on a solitary metal– oxide–semiconductor incorporated circuit chip.

- Speed - 1.1 Ghz
  A higher clock speed implies a quicker central processor. Nonetheless, numerous different variables become an integral factor. Your computer processor measures numerous instructions (low-level estimations like arithmetic) from various

projects each second. The clock speed estimates the quantity (number) of cycles your computer chip executes each second, estimated in GHz (gigahertz).

- RAM - 256 MB (min)
  RAM represents Random access memory, yet what's the use? Your PC ram is basically momentary memory where information is put away as the processor needs it. This isn't to be mistaken for long term information that is put away on your hard drive, which stays there in any event even when your PC is off.
- Hard Disk - 20 GB
  Hard disk is as secondary storage device, which is intended to store information for all time. The secondary storage device incorporates an enormous stockpiling limit when contrasted with the essential or primary storage device. The information put away in a hard disk is held when our PC framework closes down.
- Key Board - Standard Windows Keyboard
  A PC console is an info gadget that permits an individual to enter letters, numbers, and different images (together, these are called characters) into a PC. It is perhaps the most

utilized info gadgets for PCs. Utilizing a console is frequently called typing or

composing.

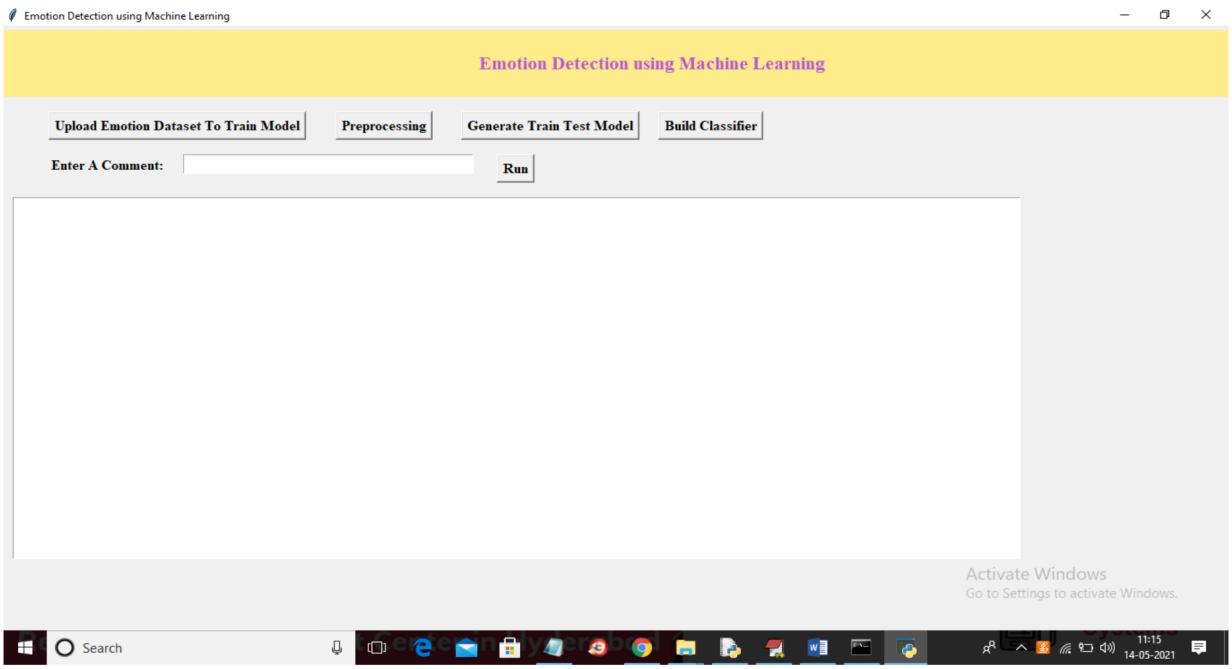
• Mouse - Two or Three Button Mouse • Monitor - SVGA

Dataset:

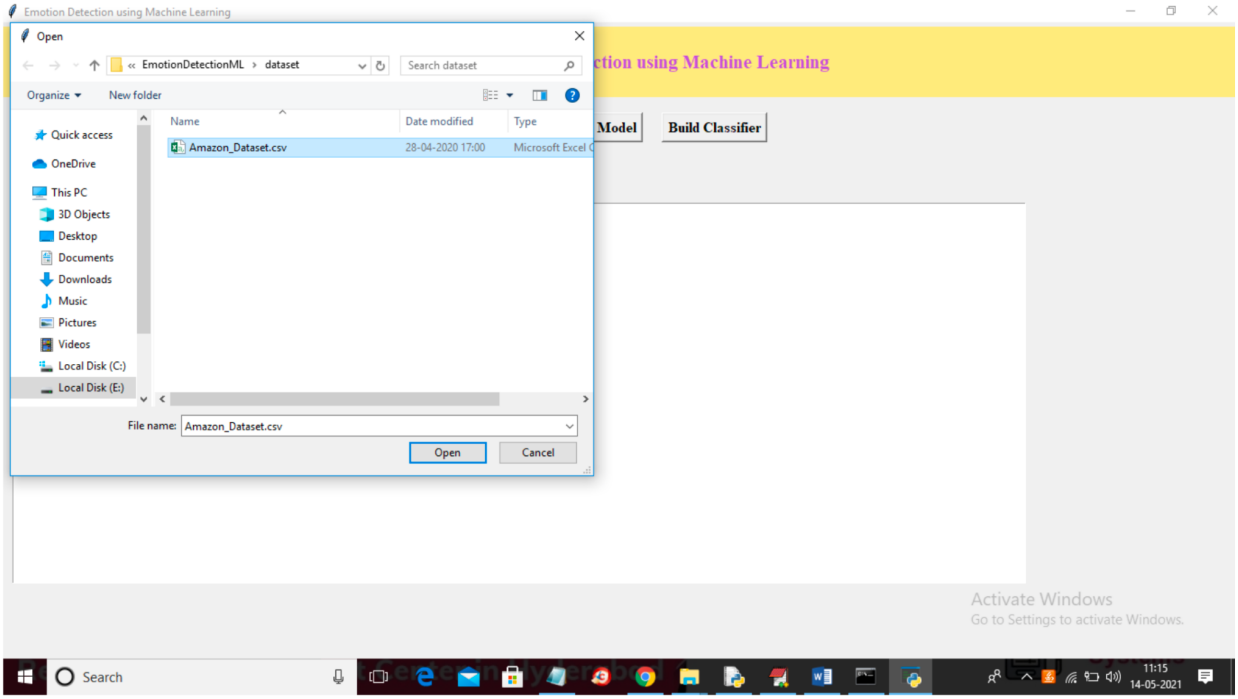
- Dataset is about Amazon reviews taken from kaggle to train the model.
- Here is the link: *https://www.kaggle.com/datasets/yasserh/amazon-product-reviews-dataset*

CONSTRUCTION

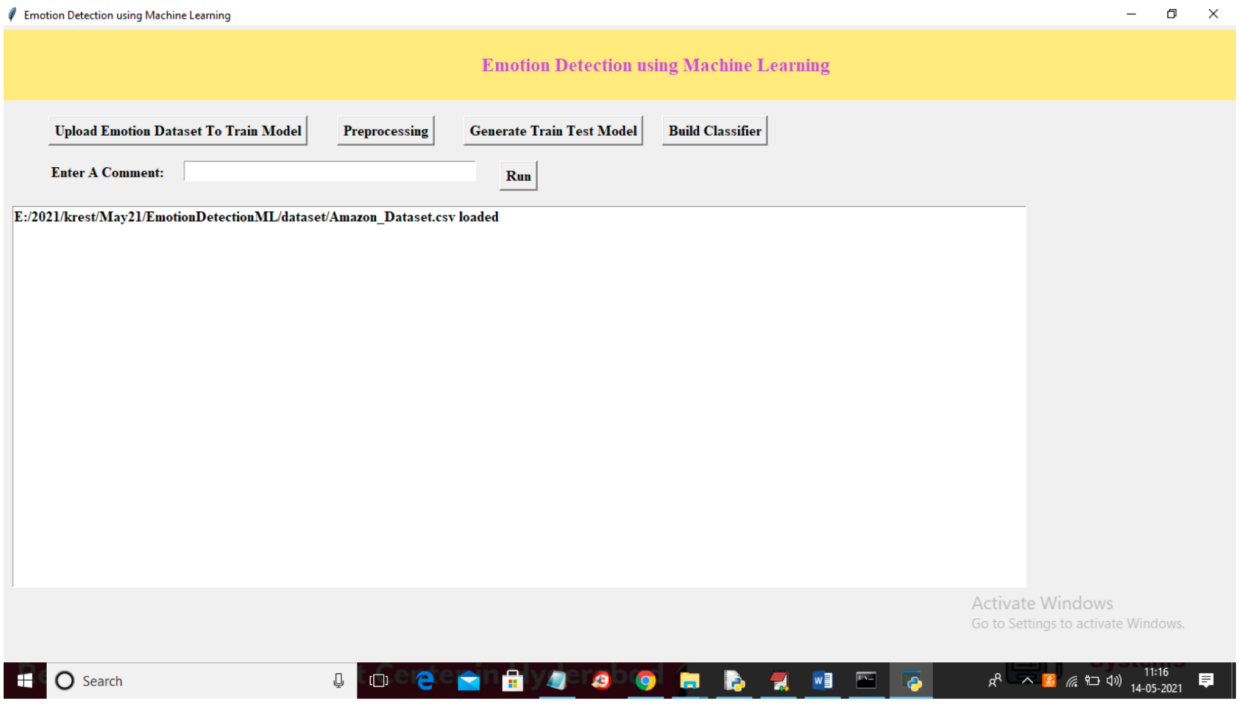
Implementation: To run project double, click on ‘run.bat’ file to get below screen



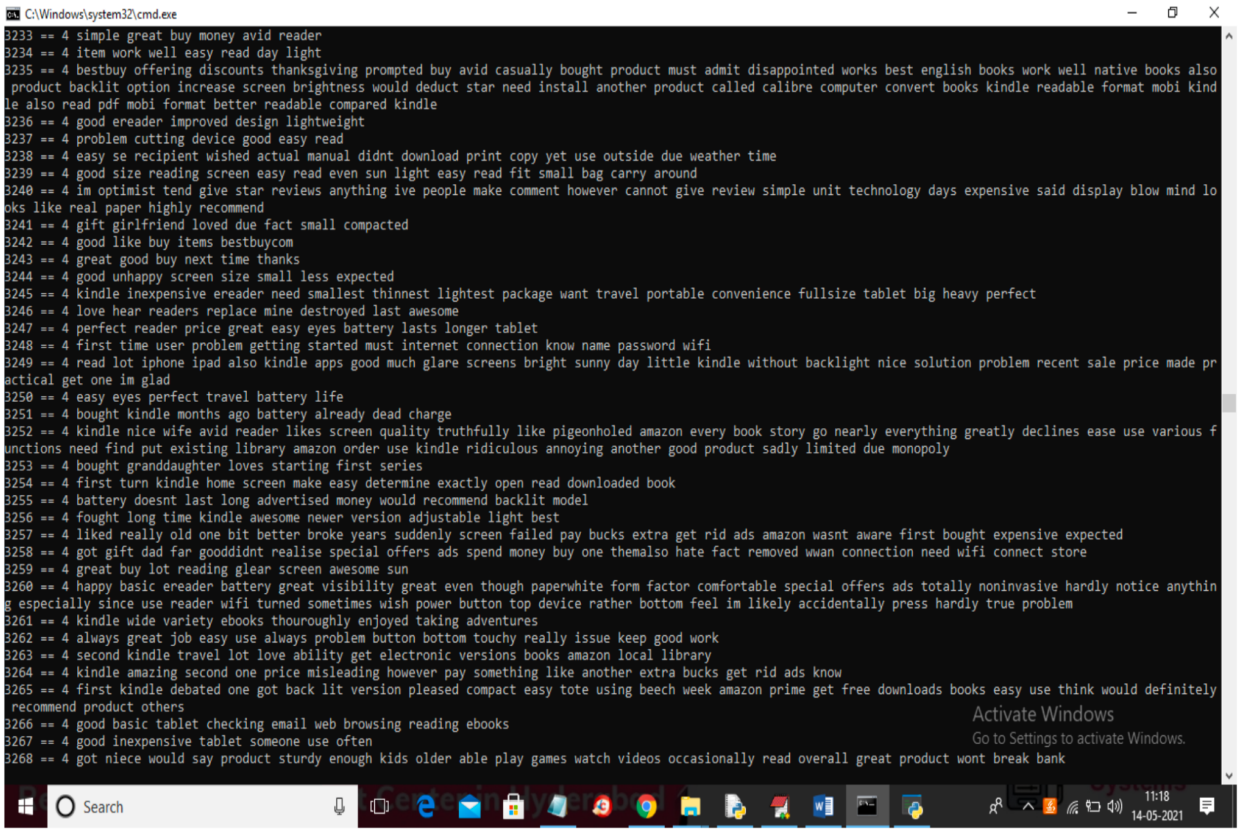
In above screen click on 'Upload Emotion Dataset to Train Model' button to upload Amazon reviews dataset



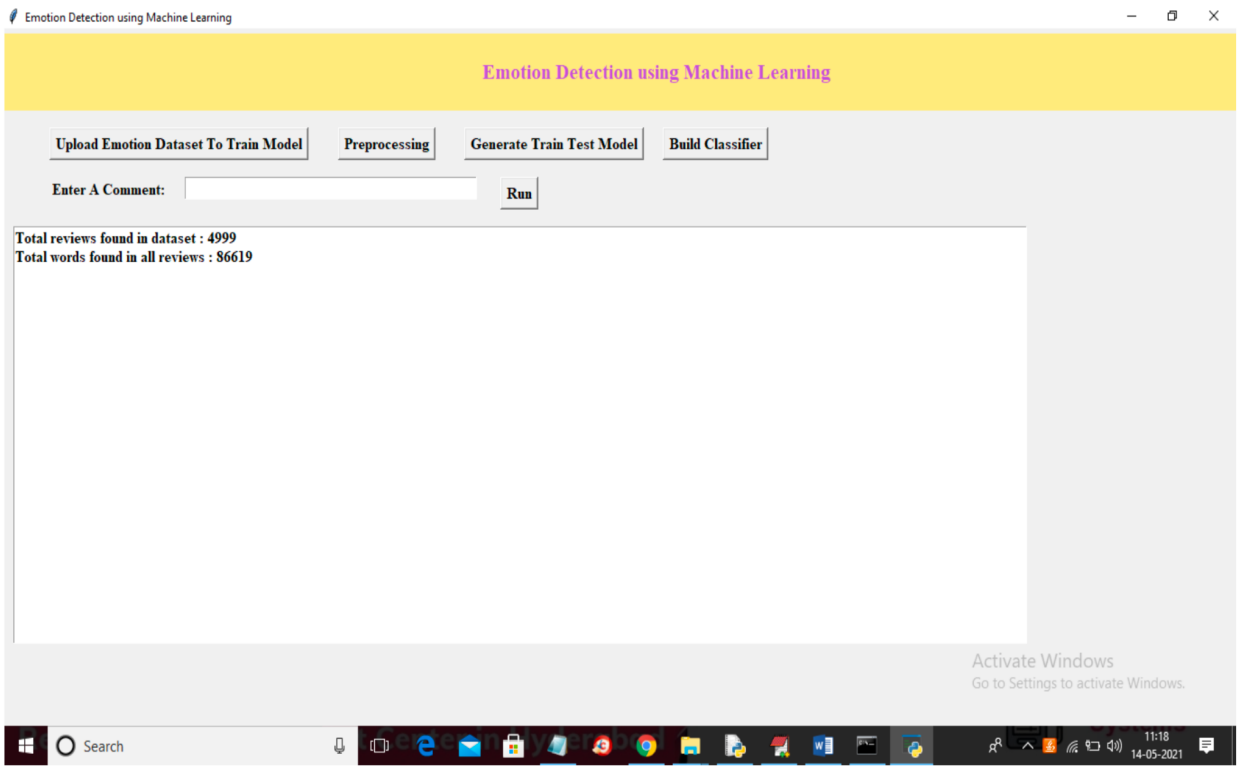
In above screen selecting and uploading ‘Amazon_Dataset.csv’ file and then click on “open” button to load dataset and to get below screen



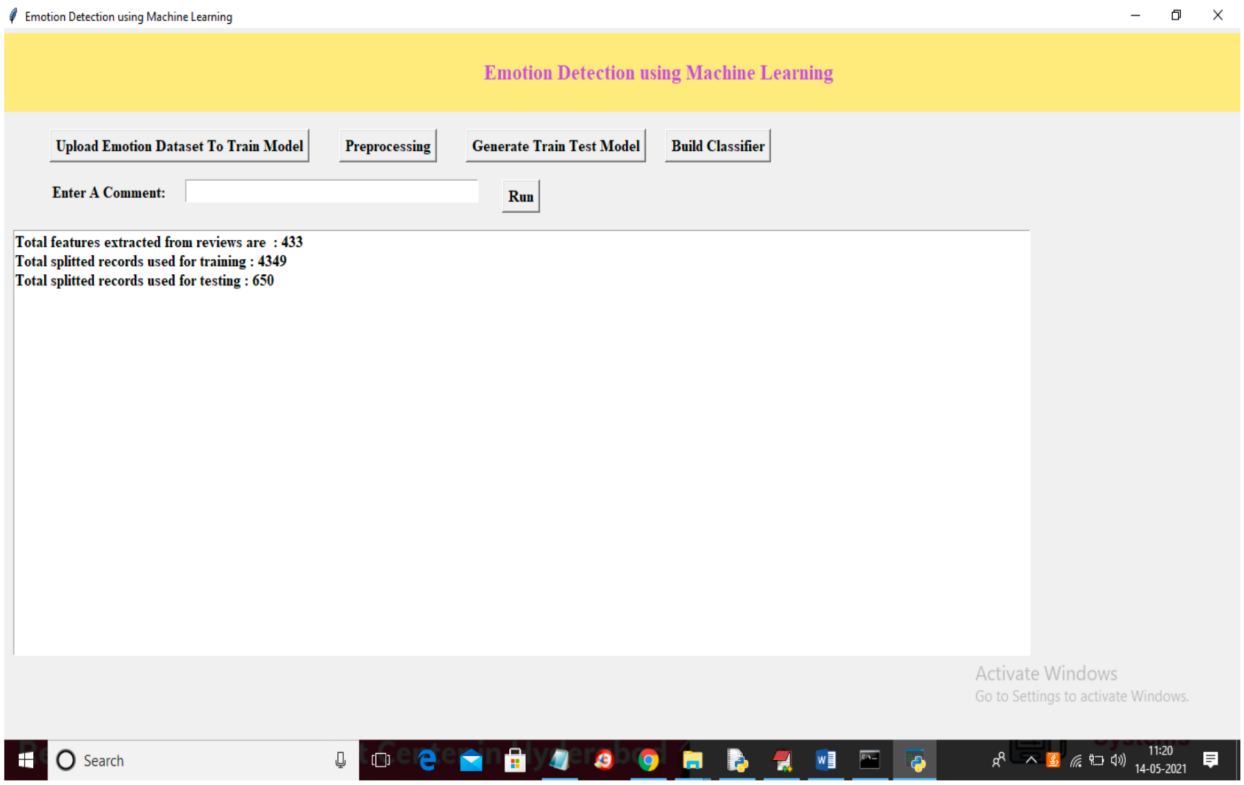
In above screen dataset loaded and now click on ‘Preprocessing’ button to read all reviews from dataset and then remove stop words, special symbols and make all reviews as clean text



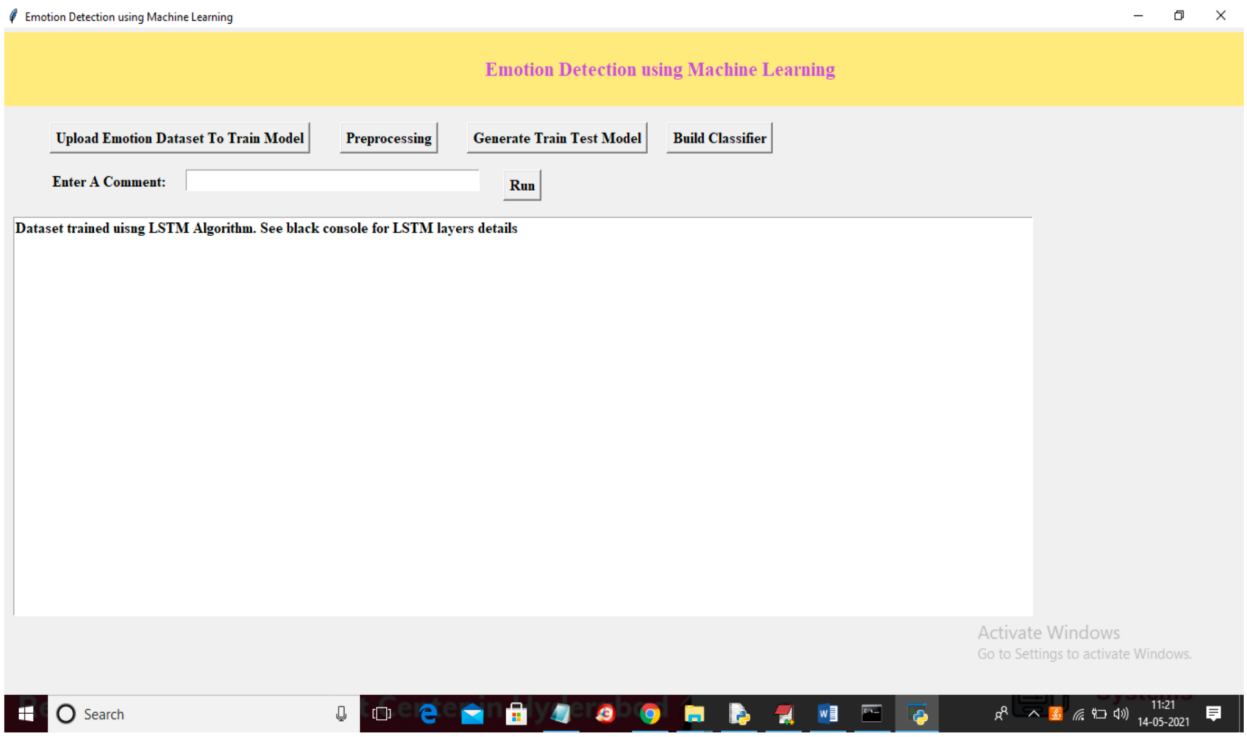
In above screen in black console, we can see all reviews are reading processing



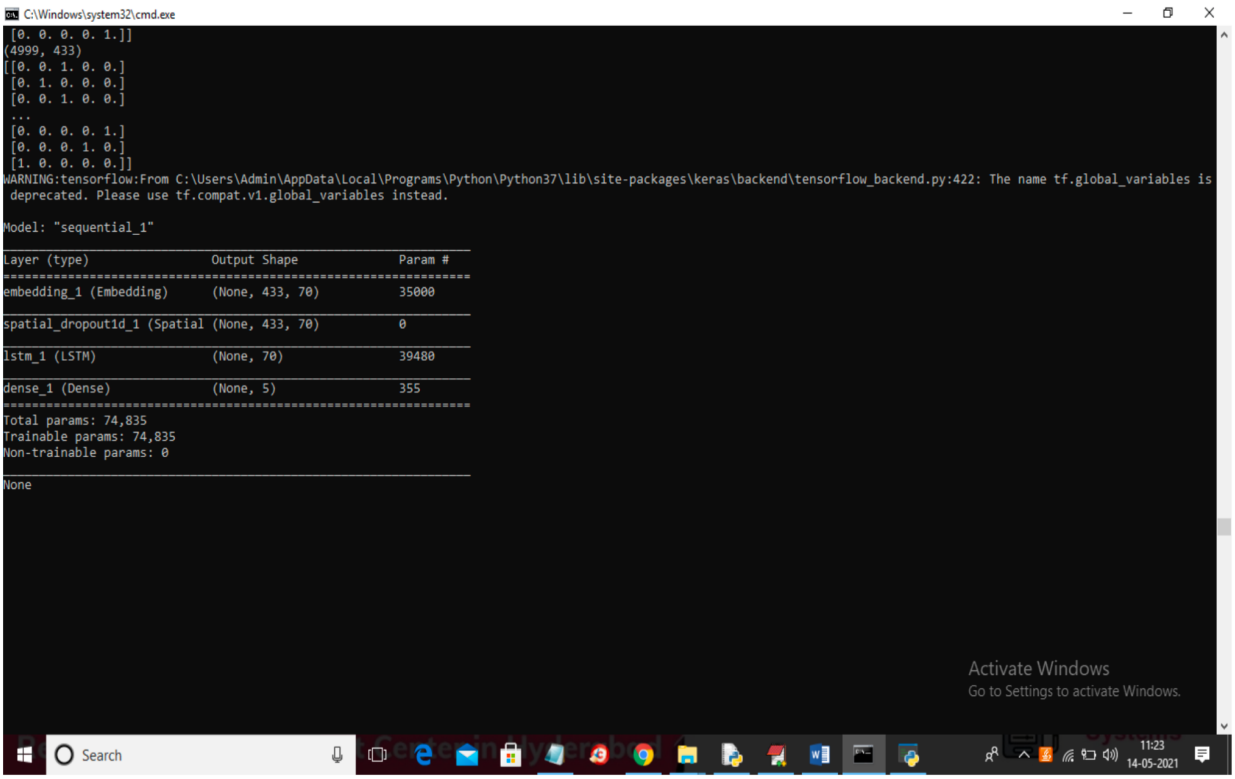
In above screen we can see dataset contains total 4999 reviews and all reviews contain 86619 words and now click on ‘Generate Train Test Model’ button to split dataset into train and test model where application used 80% dataset to train ML and 20% to test ML



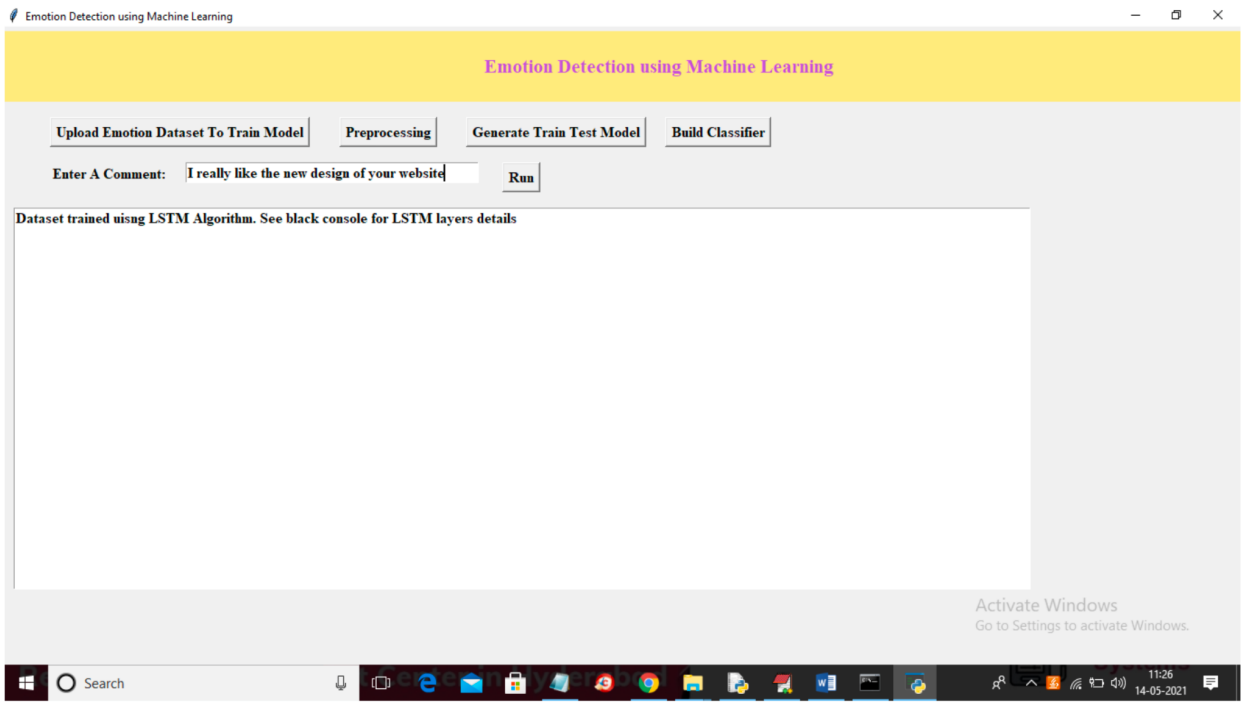
In above screen we can see application using 4349 reviews to train ML and 650 reviews to test ML and now train and test data is ready and now click on 'Build Classifier' button to build ML model



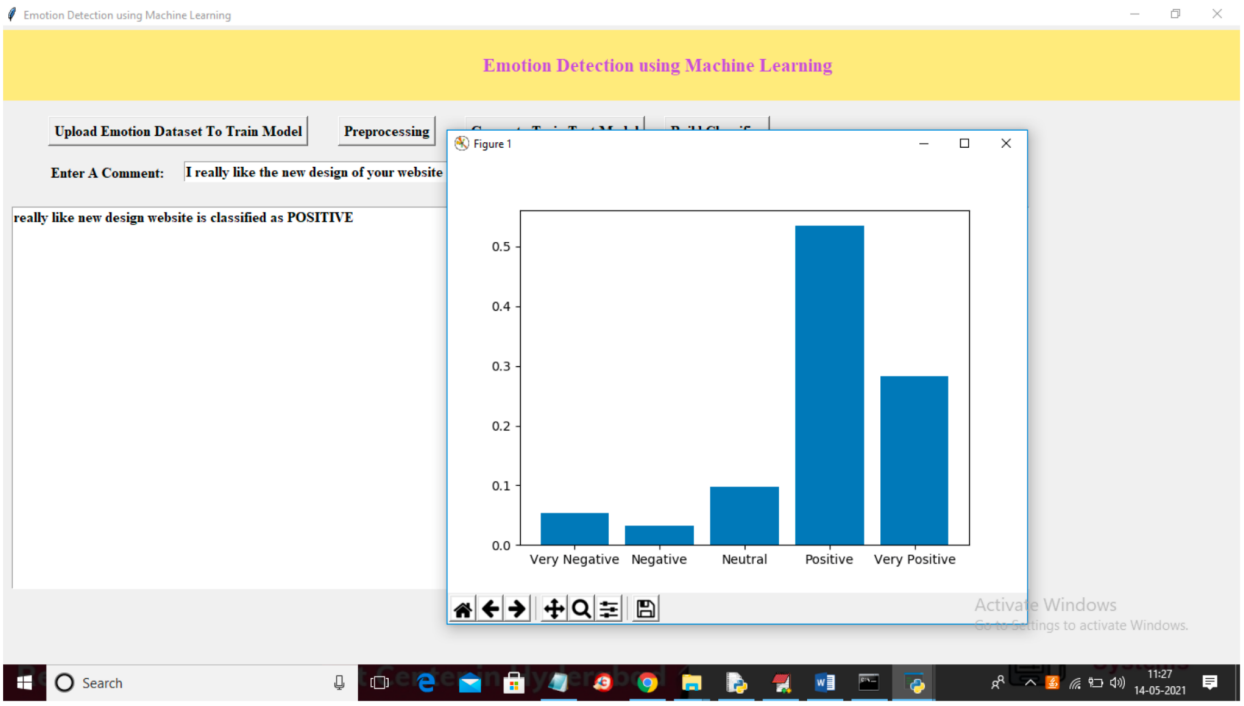
In above screen LSTM model is trained on above dataset and in below console we can see LSTM layer details



In above screen we trained LSTM with multiple layers and in first layer application using 433 X 70 features to train ML and 70 features to perform prediction. Now LSTM model is ready and now you can enter any TEXT data to predict emotion.



In above screen in COMMENT field, I entered some text and then click on ‘Run’ button to detect emotion from sentiment.



In above screen in text area, we can see given sentence predicted as 'POSITIVE' and in graph we can see polarity detection for given TEXT such as how much negativity, neutral and positivity detected and in graph we can see Positive has got highest polarity so TEXT classified as positive. Now test with another sentence.

Implementation Details

ABOUT PYTHON

Python is one of those uncommon language which can profess to be both basic

and incredible. You will be charmingly amazed to perceive that it is so natural to focus on the answer for the issue instead of on the grammar (for example the construction of the program that you are composing) of the language.

Python introduction is: -

Python is a simple to learn, amazing programming language. It has productive

significant level information structures and a basic yet viable way to deal with object-oriented programming. Python's rich sentence structure and dynamic composing, along with its interpreted nature, make it an ideal language for fast application advancement in numerous spaces on most stages

About Python:

Python is a significant high-level programming language that is a broadly useful

programming. Made by Guido van Rossum and first declared in 1991, Python has a plan theory that accentuates and emphasizes on code coherence, prominently utilizing huge whitespace. It builds and empowers clear programming on both small and enormous scopes. In July 2018, Van Rossum ventured down as the forerunner in the language community area later after 30 years. Python includes a unique sort framework and programmed memory. It upholds numerous programming ideal models, including object-situated, basic, useful and procedural, and has a huge and extensive standard library. Python mediators are accessible for some working frameworks. CPython, the reference execution of Python, is open - source programming and has a local area-based advancement model. Python and CPython are overseen by the non-benefit Python Software Foundation.

Python Versions

1.Python 2.0 was delivered on 16 October 2000 and had many major and important new highlights, including a cycle-identifying garbage collector specialist and backing for Unicode. With this release, the advancement cycle turned out to be more straightforward, community upheld and transparent

2.Python 3.0 (at first called Python 3000 or py3k) was delivered on 3 December 2008 after a very long testing period of different versions. It is a significant correction of the language that isn't totally backward-compatible with past forms or versions. Be that as it may, a significant number of its significant highlights have been backported to the Python 2.6.x and 2.7.x series and arrivals of Python 3 incorporate the 2to3 utility, which automates the interpretation of Python 2 to Python 3 code.

3.Python 2.7's death-of-life date was at first set in 2015, and afterward delayed to 2020 out of worry that an enormous group of existing code couldn't undoubtedly be forward-ported to Python 3. In January 2017, Google declared work on a Python 2.7 to Go trans-compiler to improve execution under simultaneous responsibilities.

Features of Python

1. Simple - Python is a straightforward and moderate language. Perusing a decent Python program feels practically like understanding English (however severe and strict English!). This pseudocode nature of Python is probably the best strength. It permits you to focus on the answer for the issue instead of the language structure for example the actual language.
2. Easy to Learn - As you will see, Python is amazingly simple to begin with. Python has a remarkably basic language structure as of now referenced.
3. Free and Open Source - Python is an illustration of a FLOSS (Free/Library and Open-Source Programming). In basic terms, you can unreservedly convey duplicates of this product, read the product's source code, make changes to it, use bits of it in new free projects, and that you realize you can do these things. FLOSS depends on the idea of a community that shares information and knowledge. This is one main reason why Python is so acceptable - it has been made and improved by a local area or community who simply need to see a superior Python.
4. High-level Language - At the point when you compose programs in Python, you never need to worry about low-level subtleties or details, for example, dealing with the memory utilized by your program.

CONCLUSION AND FUTURE SCOPE

Conclusion:

In this project, a research to classify facial emotions over static emotion detection using machine learning was developed. This is a complex problem that has already been approached several times with different techniques. While good results have been achieved using feature engineering, this project focused on feature learning, which is one of ML promises. While the results achieved were not state-of-the-art, they were

slightly better than other techniques including feature engineering. It means that eventually ML techniques will be able to solve this problem given an enough number of labelled examples. While feature engineering is not necessary, image pre- processing boosts classification accuracy. Hence, it reduces noise on the input data. Nowadays, facial emotion detection software includes the use of feature engineering. Thus, emotion classification could be achieved by means of deep learning techniques.

Traditional emotion and sentiment detection involves exploitation reference dictionaries of however positive bound words are then scheming the common of those scores because of the sentiment of that text. Next is employing an easy cc model to create the classification. This is often done by generating “features” from the text then exploiting these options to predict a “label”. Associate in Nursing example of generating options is cacophonous the text up into words then exploitation these words and their frequencies in the text as options.

Future Scope:

The studies conducted using both ML and DL algorithms proposed system that applied many kinds of characteristics and models on the collected feedbacks based on text to determine the emotions and sentiments. The solution at last shows that, considering emoticon along with the text has a positive influence on the emotion and sentiment detection. It is also found that, DL algorithms works better than ML algorithms. Finally, this research concludes by outperforming existing researches that in the future, this research can be extended to the field of multilingual data.

The future of emotion and sentiment detection is going to continue to dig deeper, so much past the surface of the quantity of likes, comments, and shares, and aim to achieve, and really perceive, the importance of social media interactions and what they tell world regarding the shoppers behind the screens. This forecast additionally predicts broader applications for sentiment analysis – brands can still leverage this tool, however thus can people within the prominence, governments, non-profits, education centres, and plenty of alternative organizations.

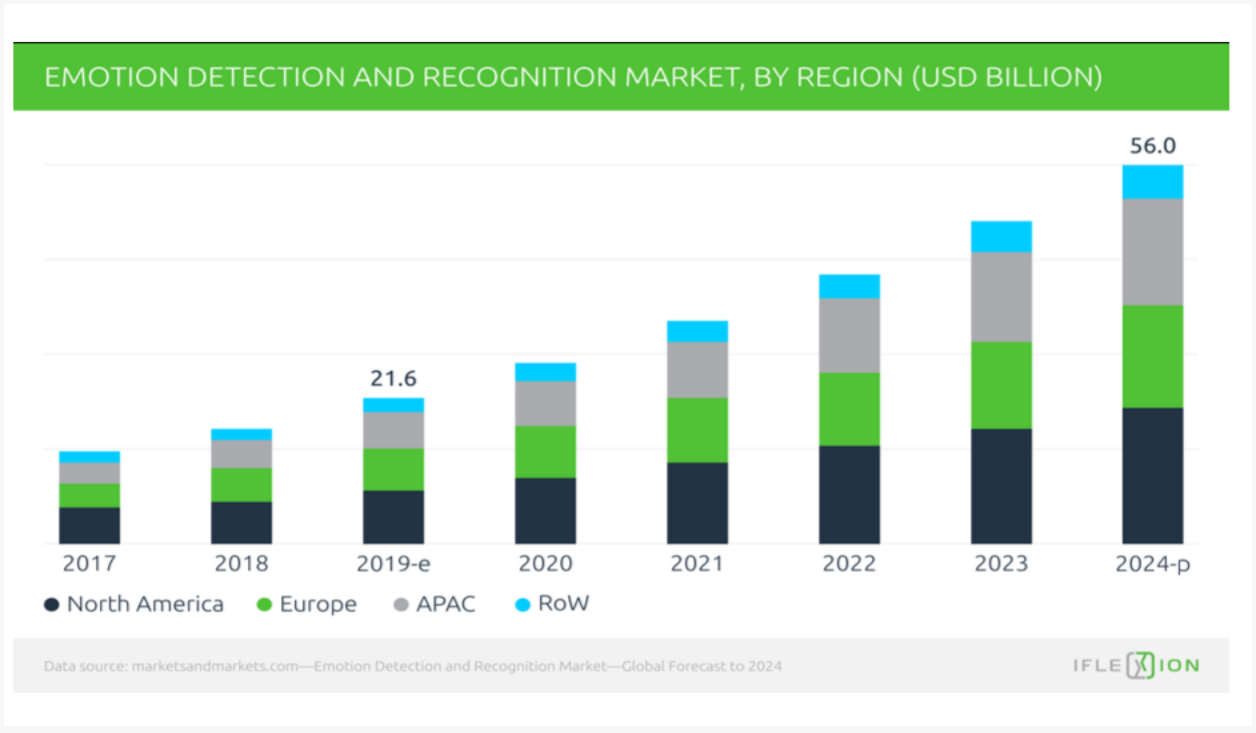
In the future, the results of several alternative feature choice strategies may be explored. extra researches may be performed for evaluating the impact of diverse areas and domain-based factors. Enhancing the applying of sentiment analysis to extended

areas could lead to fascinating facts. within the future, additional mixtures of n-grams and have allowance giving associate improved accuracy than this one may be used. This analysis is just connected to sentiment categorization into 2 basic categories referred to as binary classification. Sentiment will belong to either a positive feeling or a negative feeling.

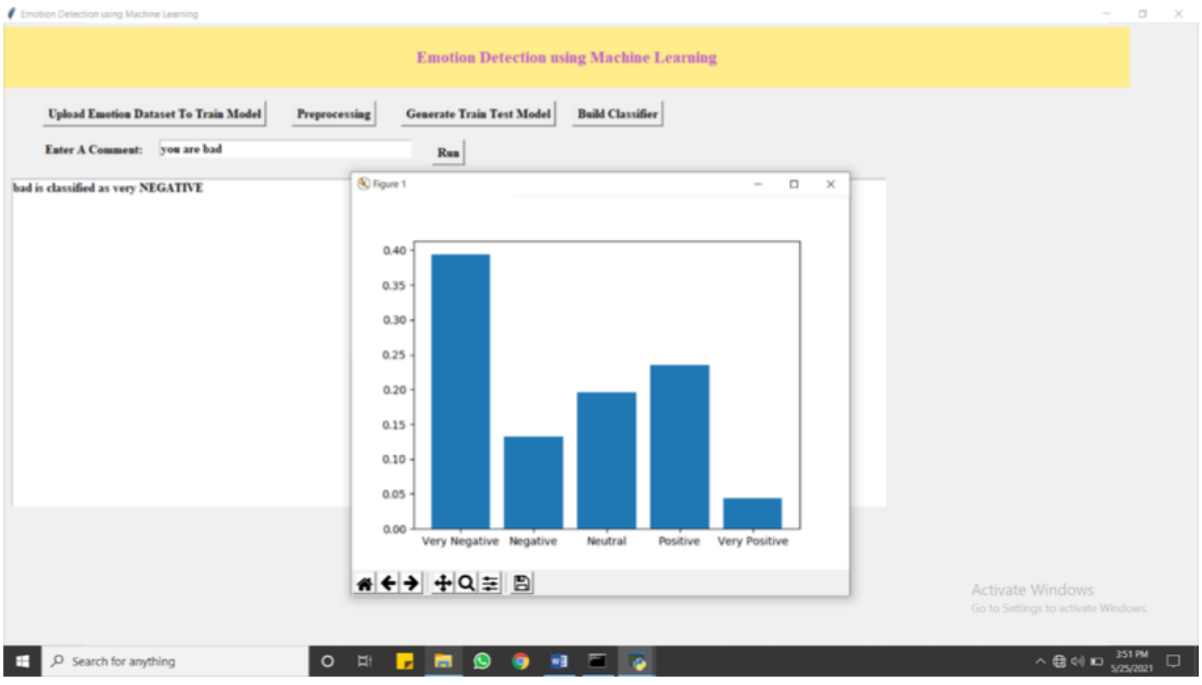
For improvement within the future, quite one category for sentiment categorization is thought-about like neutral, negative, positive, etc. during this paper, the stress is on discovering options that seem clearly within the variety of nouns or nominal phrases within the evaluations. The implicit feature study is unbroken for future work. Since the ensemble learning approaches need a lot of computation time, parallel computing strategies is explored to handle this issue. the most constraint of ensemble learning approaches is that the absence of result depiction and also the data gathered by ensembles is hard for human understanding. So, the advance of ensemble interpretability is another important analysis track.

Future opinion associate degree lysis structures would like a wider and in-depth public knowledge base which is able to end in an improved thought of linguistic communication opinions and can be more practical to link the gap among multimodal and machine wise knowledge. a mix of scientific theories relating to feeling with the applied technology targets to research sentiments of linguistic communication script can end in extra bio-inspired ways to style intelligent opinion analysis structures ready to handle linguistics data, analogy creation, learning actual data, and perceiving, distinctive, and sensing emotions.

Technologies based on facial expression recognition (FER, also known as affect recognition) form a significant part of the emotion recognition market, estimated to reach a value of $56 billion by 20245.



Expected Output:



The comment given here is “You are bad” and when implemented the project we get an output showing in a bar graph that it is “very negative” and also it shows in the form of text saying “bad is classified as very NEGATIVE”.

References:

[1] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems., pages 1097–1105, 2012.

[2] Amazon Web Services, Inc. Ec2 instance types. http://aws.amazon.com/ec2/ instance- types/, 2016. [Online; Accessed 27 May 2016].

[3] Andrej Karpathy. Cs231n convolutional neural networks for visual recognition. http://cs231n.github.io/convolutional-networks/, 2015. [Online; Accessed 24 May 2016].

[4] Andrej Karpathy. Stanford university: Cs231n convolutional neural networks for visual recognition. http://cs231n.github.io/neural-networks-1, 2016. Online; Accessed 07 June 2016.

SCREENSHOTS AND CODE

Main.py

```
from tkinter import messagebox

from tkinter import *

from tkinter import simpledialog

import tkinter

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

from tkinter import simpledialog

from tkinter import filedialog

from keras.utils.np_utils import to_categorical

from emoji import UNICODE_EMOJI

import os

from sklearn.feature_extraction.text import CountVectorizer

from keras.preprocessing.text import Tokenizer

from keras.preprocessing.sequence import pad_sequences

import re

from string import punctuation

from keras.models import Sequential

from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D

from sklearn.model_selection import train_test_split

from nltk.corpus import stopwords

from keras.models import model_from_json
```

```
stop_words = set(stopwords.words('english'))

os.environ["PYTHONIOENCODING"] = "utf-8"



main = tkinter.Tk()

main.title("Emotion Detection using Machine Learning") #designing main screen

main.geometry("1300x1200")


global filename

global X, Y

global X_train, X_test, Y_train, Y_test

global tokenizer

global XX

global model

global positive_score,negative_score,neutral_score


def upload():

    global filename

    filename = filedialog.askopenfilename(initialdir="dataset")

    text.delete('1.0', END)

    text.insert(END,filename+" loaded\n");


def checkInput(inputdata):

    option = 0
```

```
    try:
        s = float(inputdata)
        option = 0
    except:
        option = 1
    return option

def clean_doc(doc):
    tokens = doc.split()
    table = str.maketrans('', '', punctuation)
    tokens = [w.translate(table) for w in tokens]
    tokens = [word for word in tokens if word.isalpha()]
    stop_words = set(stopwords.words('english'))
    tokens = [w for w in tokens if not w in stop_words]
    tokens = [word for word in tokens if len(word) > 1]
    tokens = ' '.join(tokens) #here upto for word based
    return tokens

def Preprocessing():
    global X
    global Y
    X = []
    Y = []
    text.delete('1.0', END)
    train = pd.read_csv(filename,encoding = "ISO-8859-1")
```

```
count = 0
for i in range(len(train)):
    #ids = train.get_value(i,0,takeable = True)
    sentiment = train.get_value(i,0,takeable = True)
    tweet = train.get_value(i,1,takeable = True)
    check = checkInput(tweet)
    if check == 1:
        tweet = tweet.lower().strip()
        tweet = clean_doc(tweet)
        print(str(i)+" == "+str(sentiment)+" "+tweet)
        '''
        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)
            Y.append((sentiment-1))
            count = count + len(arr)
    X = np.asarray(X)
    Y = np.asarray(Y)
    Y = np.nan_to_num(Y)
    print(Y)
    #Y = np.asarray(Y)#pd.get_dummies(train['sentiment']).values
    Y = to_categorical(Y)
    print(Y)
    text.insert(END,'Total reviews found in dataset : '+str(len(X))+"\n")
    text.insert(END,'Total words found in all reviews : '+str(count)+"\n")


def generateModel():
    text.delete('1.0', END)
    global XX
    global tokenizer
    global X_train, X_test, Y_train, Y_test
    max_fatures = 500
    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)
    text.insert(END,'Total features extracted from reviews are  : '+str(X_train.shape[1])+"\n")
    text.insert(END,'Total splitted records used for training : '+str(len(X_train))+"\n")
    text.insert(END,'Total splitted records used for testing : '+str(len(X_test))+"\n")


def buildClassifier():
    global model
    global XX
    global Y
    text.delete('1.0', END)
    embed_dim = 70
    lstm_out = 70
    max_fatures = 500


    indices = np.arange(XX.shape[0])
    np.random.shuffle(indices)


    XX = XX[indices]
    Y = Y[indices]


    print(XX.shape)
    print(Y)
    if os.path.exists('model/model.json'):
        with open('model/model.json', "r") as json_file:
```

```
            loaded_model_json = json_file.read()
            model = model_from_json(loaded_model_json)
        json_file.close()
        model.load_weights("model/model_weights.h5")
        model._make_predict_function()
        text.insert(END,'Dataset trained uisng LSTM Algorithm. See black console for LSTM layers details\n')
    else:
        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(5,activation='softmax'))
        model.compile(loss = 'categorical_crossentropy', optimizer='adam',metrics = ['accuracy'])
        print(model.summary())
        batch_size = 256
        model.fit(XX, Y, epochs = 100, batch_size=batch_size, verbose = 1)
        model.save_weights('model/model_weights.h5')
        model_json = model.to_json()
        with open("model/model.json", "w") as json_file:
            json_file.write(model_json)
        json_file.close()
        text.insert(END,'Dataset trained uisng LSTM Algorithm. See black console for LSTM layers details\n')
    print(model.summary())
```

```
def module1():
    global positive_score,negative_score,neutral_score
    text.delete('1.0', END)
    sentence = tf1.get()
    sentence = sentence.lower().strip()
    sentence = clean_doc(sentence)
    textdata = sentence.strip()
    mytext = [textdata]
    twts = tokenizer.texts_to_sequences(mytext)
    twts = pad_sequences(twts, maxlen=433, dtype='int32', value=0)
    sentiment = model.predict(twts,batch_size=1,verbose = 2)[0]
    print(sentiment)
    result = np.argmax(sentiment)
    if result == 0:
        text.insert(END,sentence+' is classified as very NEGATIVE\n\n')
    if result == 1:
        text.insert(END,sentence+' is classified as NEGATIVE\n\n')
    if result == 2:
        text.insert(END,sentence+' is classified as NEUTRAL\n\n')
    if result == 3:
        text.insert(END,sentence+' is classified as POSITIVE\n\n')
    if result == 4:
        text.insert(END,sentence+' is classified as Very Positive\n\n')

    height = sentiment
```

```
    bars = ('Very Negative', 'Negative','Neutral','Positive','Very Positive')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, height)
    plt.xticks(y_pos, bars)
    plt.show()



font = ('times', 16, 'bold')
title = Label(main, text='Emotion Detection using Machine Learning')
title.config(bg='LightGoldenrod1', fg='medium orchid')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)

font1 = ('times', 12, 'bold')
text=Text(main,height=22,width=140)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=10,y=200)
text.config(font=font1)



font1 = ('times', 12, 'bold')
```

```
uploadButton = Button(main, text="Upload Emotion Dataset To Train Model", command=upload)

uploadButton.place(x=50,y=100)

uploadButton.config(font=font1)


preButton = Button(main, text="Preprocessing", command=Preprocessing)

preButton.place(x=370,y=100)

preButton.config(font=font1)


modelButton = Button(main, text="Generate Train Test Model", command=generateModel)

modelButton.place(x=510,y=100)

modelButton.config(font=font1)


classifierButton = Button(main, text="Build Classifier", command=buildClassifier)

classifierButton.place(x=730,y=100)

classifierButton.config(font=font1)


l1 = Label(main, text='Enter A Comment:')

l1.config(font=font1)

l1.place(x=50,y=150)


tf1 = Entry(main,width=40)

tf1.config(font=font1)

tf1.place(x=200,y=150)

runButton = Button(main, text="Run", command=module1)

runButton.place(x=550,y=150)

runButton.config(font=font1)


#main.config(bg='OliveDrab2')

main.mainloop()
```