**CODE:**

#import inline as inline

import matplotlib

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import sqlite3

#%matplotlib inline

drugdata = pd.read\_csv("C:\\Users\\SANDHYA RAMESH\\Downloads\\Opinion-Mining-using-the-UCI-Drug-Review-Dataset-master\\Opinion-Mining-using-the-UCI-Drug-Review-Dataset-master\\drug.csv", encoding = "ISO-8859-1")

drugdata = drugdata.head(50)

X = drugdata.drop(['cleanReview','review','ratingSentiment','rating','vaderReviewScore','vaderSentimentLabel'], axis=1)

y = drugdata['vaderSentimentLabel']

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.20)

from sklearn.svm import SVC

svclassifier = SVC(kernel='linear')

svclassifier.fit(X\_train, y\_train)

# imports

from tkinter import \*

from tkinter import messagebox as ms

import sqlite3

# make database and users (if not exists already) table at programme start up

with sqlite3.connect('quit.db') as db:

c = db.cursor()

c.execute('CREATE TABLE IF NOT EXISTS user (username TEXT NOT NULL ,password TEX NOT NULL);')

db.commit()

db.close()

# main Class

class main:

global p

def \_\_init\_\_(self, master):

# Window

self.master = master

# Some Usefull variables

self.username = StringVar()

self.password = StringVar()

self.n\_username = StringVar()

self.n\_password = StringVar()

self.vaderSentiment = StringVar()

self.Id = StringVar()

self.review = StringVar()

# Create Widgets

self.widgets()

# Login Function

def login(self):

# Establish Connection

with sqlite3.connect('quit.db') as db:

c = db.cursor()

# Find user If there is any take proper action

find\_user = ('SELECT \* FROM user WHERE username = ? and password = ?')

c.execute(find\_user, [(self.username.get()), (self.password.get())])

result = c.fetchall()

if result:

self.logf.pack\_forget()

# self.head['text'] = self.username.get() + '\n Loged In'

self.head = Label(self.master, text='Details', font=('', 35), pady=10)

self.head.pack()

self.detf = Frame(self.master, padx=10, pady=10)

Label(self.detf, text='RATING: ', font=('', 20), pady=5, padx=5).grid(sticky=W)

Entry(self.detf, textvariable=self.vaderSentiment, bd=5, font=('', 15)).grid(row=0, column=1)

Label(self.detf, text='ID: ', font=('', 20), pady=5, padx=5).grid(sticky=W)

Entry(self.detf, textvariable=self.Id, bd=5, font=('', 15)).grid(row=1, column=1)

Label(self.detf, text='REVIEW: ', font=('', 20), pady=5, padx=5).grid(sticky=W)

Entry(self.detf, textvariable=self.review, bd=5, font=('', 15)).grid(row=2, column=1)

Button(self.detf, text=' Submit ', bd=3, font=('', 15), padx=5, pady=5, command=self.algo).grid()

# Button(self.detf,text = ' Create Account ',bd = 3 ,font = ('',15),padx=5,pady=5,command=self.cr).grid(row=2,column=1)

self.detf.pack()

else:

ms.showerror('Oops!', 'Username Not Found.')

def new\_user(self):

# Establish Connection

with sqlite3.connect('quit.db') as db:

c = db.cursor()

# Find Existing username if any take proper action

find\_user = ('SELECT \* FROM user WHERE username = ?')

c.execute(find\_user, [(self.username.get())])

if c.fetchall():

ms.showerror('Error!', 'Username Taken Try a Diffrent One.')

else:

ms.showinfo('Success!', 'Account Created!')

self.log()

# Create New Account

insert = 'INSERT INTO user(username,password) VALUES(?,?)'

c.execute(insert, [(self.n\_username.get()), (self.n\_password.get())])

db.commit()

# Frame Packing Methords

def log(self):

self.username.set('')

self.password.set('')

self.crf.pack\_forget()

self.head['text'] = 'LOGIN'

self.logf.pack()

def cr(self):

self.n\_username.set('')

self.n\_password.set('')

self.logf.pack\_forget()

self.head['text'] = 'Create Account'

self.crf.pack()

def algo(self):

a = self.vaderSentiment.get()

b = self.Id.get()

c = svclassifier.predict([[a,b]])

self.detf.pack\_forget()

#self.head['text'] = 'Result'

#self.result.pack()

Label(self.head, text=c, font=('', 20), pady=5, padx=5).grid(sticky=W)

self.head.pack()

# Draw Widgets

def widgets(self):

self.head = Label(self.master, text='WELCOME', font=('', 35), pady=10)

self.head.pack()

self.logf = Frame(self.master, padx=10, pady=10)

self.head = Label(self.logf, text='Login', font=('', 35), pady=10)

Label(self.logf, text='Username: ', font=('', 20), pady=5, padx=5).grid(sticky=W)

Entry(self.logf, textvariable=self.username, bd=5, font=('', 15)).grid(row=0, column=1)

Label(self.logf, text='Password: ', font=('', 20), pady=5, padx=5).grid(sticky=W)

Entry(self.logf, textvariable=self.password, bd=5, font=('', 15), show='\*').grid(row=1, column=1)

Button(self.logf, text=' Login ', bd=3, font=('', 15), padx=5, pady=5, command=self.login).grid()

Button(self.logf, text=' Create Account ', bd=3, font=('', 15), padx=5, pady=5, command=self.cr).grid(row=2,column=1)

self.logf.pack()

self.crf = Frame(self.master, padx=10, pady=10)

Label(self.crf, text='Username: ', font=('', 20), pady=5, padx=5).grid(sticky=W)

Entry(self.crf, textvariable=self.n\_username, bd=5, font=('', 15)).grid(row=0, column=1)

Label(self.crf, text='Password: ', font=('', 20), pady=5, padx=5).grid(sticky=W)

Entry(self.crf, textvariable=self.n\_password, bd=5, font=('', 15), show='\*').grid(row=1, column=1)

Button(self.crf, text='Create Account', bd=3, font=('', 15), padx=5, pady=5, command=self.new\_user).grid()

Button(self.crf, text='Go to Login', bd=3, font=('', 15), padx=5, pady=5, command=self.log).grid(row=2,column=1)

#self.result = Frame(self.master, padx=10, pady=10)

#Label(self.result, text=p, font=('',20), pady=5, padx=5).grid(sticky=W)

#self.result.pack()

# create window and application object

root = Tk()

# root.title("Login Form")

main(root)

root.mainloop()

**SCREEN SHOT:**



