from django.shortcuts import render

from django.template import RequestContext

from django.contrib import messages

import pymysql

from django.http import HttpResponse

import pandas as pd

import numpy as np

import re

from sklearn.feature\_extraction.text import TfidfVectorizer

from numpy import dot

from numpy.linalg import norm

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

from nltk.corpus import stopwords

stop\_words = set(stopwords.words('english'))

sid = SentimentIntensityAnalyzer()

def getReview(review):

review\_result = "none"

review = review.lower()

review = re.sub('[^A-Za-z]+', ' ', review)

sentiment\_dict = sid.polarity\_scores(review.strip())

compound = sentiment\_dict['compound']

if compound >= 0.05 :

review\_result = 'Positive'

return review\_result

dataset = pd.read\_csv("Dataset/amazon\_reviews.csv")

dataset = dataset.values

text = dataset[:,0]

label = dataset[:,1]

tfidf\_vectorizer = TfidfVectorizer(stop\_words=stop\_words, use\_idf=True, smooth\_idf=False, norm=None, decode\_error='replace', max\_features=1000,lowercase=True)

tfidf = tfidf\_vectorizer.fit\_transform(text).toarray()

df = pd.DataFrame(tfidf, columns=tfidf\_vectorizer.get\_feature\_names())

print(df.shape)

df = df.values

X = df[:, 0:1000]

def index(request):

if request.method == 'GET':

return render(request, 'index.html', {})

def Login(request):

if request.method == 'GET':

return render(request, 'Login.html', {})

def Register(request):

if request.method == 'GET':

return render(request, 'Register.html', {})

def Recommendation(request):

if request.method == 'GET':

return render(request, 'Recommendation.html', {})

def RecommendationAction(request):

if request.method == 'POST':

query = request.POST.get('t1', False)

test = query.lower().strip()

test = tfidf\_vectorizer.transform([test]).toarray()

test = test[0]

similarity = 0

review = 'Unable to get review for recommendation'

rating = 0

suggestion = "No suggestion available"

for j in range(len(X)):

review\_score = dot(X[j], test)/(norm(X[j])\*norm(test))

if review\_score > similarity:

similarity = review\_score

review\_type = getReview(text[j])

if review\_type == 'Positive':

review = text[j]

rating = label[j]

suggestion = "you have chosen best product"

output="<html><body><center><table border=1><tr><th><font size=3 color=black>Product Name</th>"

output+="<th><font size=3 color=black>Recommended Best Review</th>"

output+="<th><font size=3 color=black>Recommended Best Rating</th><th><font size=3 color=black>Suggestion</th></tr>"

output+="<tr><td><font size=3 color=black>"+query+"</td><td><font size=3 color=black>"+review+"</td><td><font size=3 color=black>"+str(rating)+"</td><td><font size=3 color=black>"+suggestion+"</td>"

#output+"</tr><br/><br/><br/><br/><br/><br/></table>"

context= {'data':output}

return render(request, 'Result.html', context)

def Signup(request):

if request.method == 'POST':

#user\_ip = getClientIP(request)

#reader = geoip2.database.Reader('C:/Python/PlantDisease/GeoLite2-City.mmdb')

#response = reader.city('103.48.68.11')

#print(user\_ip)

#print(response.location.latitude)

#print(response.location.longitude)

username = request.POST.get('username', False)

password = request.POST.get('password', False)

contact = request.POST.get('contact', False)

email = request.POST.get('email', False)

address = request.POST.get('address', False)

db\_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'Recommendation',charset='utf8')

db\_cursor = db\_connection.cursor()

student\_sql\_query = "INSERT INTO register(username,password,contact,email,address) VALUES('"+username+"','"+password+"','"+contact+"','"+email+"','"+address+"')"

db\_cursor.execute(student\_sql\_query)

db\_connection.commit()

print(db\_cursor.rowcount, "Record Inserted")

if db\_cursor.rowcount == 1:

context= {'data':'Signup Process Completed'}

return render(request, 'Register.html', context)

else:

context= {'data':'Error in signup process'}

return render(request, 'Register.html', context)

def UserLogin(request):

if request.method == 'POST':

username = request.POST.get('username', False)

password = request.POST.get('password', False)

utype = 'none'

con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'Recommendation',charset='utf8')

with con:

cur = con.cursor()

cur.execute("select \* FROM register")

rows = cur.fetchall()

for row in rows:

if row[0] == username and row[1] == password:

utype = 'success'

break

if utype == 'success':

file = open('session.txt','w')

file.write(username)

file.close()

context= {'data':'welcome '+username}

return render(request, 'UserScreen.html', context)

if utype == 'none':

context= {'data':'Invalid login details'}

return render(request, 'Login.html', context)