import sqlite3

import pandas as pd

from sklearn.neighbors import NearestNeighbors

class Recommend:

    def \_\_init\_\_(self,x,y,z):

        self.db = sqlite3.connect(x)

        self.user\_input = y

        self.n\_output = z

    def content(self):

        cur = self.db.execute("select video\_no,words from video;")

        data = []

        index=[]

        for i in cur:

            index.append(i[0])

            data.append(list(i[1]))

        df = pd.DataFrame(data=data,index=index,columns=[i for i in range(len(data[0]))])

        watched = []

        cur = self.db.execute('select video\_no from user where user\_id = (?);',(self.user\_input,))

        for i in cur:

            watched.append(i[0])

        model\_nn = NearestNeighbors(metric='cosine', algorithm='brute', n\_neighbors=self.n\_output, n\_jobs=-1)

        model\_nn.fit(df)

        indices=model\_nn.kneighbors(df.iloc[watched],return\_distance=False)

        result=[]

        for i in range(len(indices[0])):

            for j in range(len(indices)):

                k=indices[j][i]

                if k not in watched:

                    if k not in result:

                        result.append(k)

            if len(result)>=self.n\_output:

                break

        return result

    def collaborate(self):

        # top\_users = 5 #obtaining their watched list to recommend

        users\_latest\_vid = 10 # no of videos to be taken from each user

        user\_id = []

        cur = self.db.execute('select distinct(user\_id) from user;')

        for i in cur:

            user\_id.append(i[0])

        video\_no = []

        cur = self.db.execute("select video\_no from video;")

        for i in cur:

            video\_no.append(i[0])

        co\_data = pd.DataFrame(index=user\_id,columns=video\_no)

        watched\_vid = []

        for i in range(len(user\_id)):

            cur = self.db.execute('select video\_no from user where user\_id=(?);',(user\_id[i],))

            x=[]

            for j in cur:

                x.append(j[0])

                co\_data.at[user\_id[i],j[0]]=1

            watched\_vid.append(x)

        co\_data = co\_data.fillna(0)

        model\_nn = NearestNeighbors(metric='cosine', algorithm='auto', n\_neighbors=len(user\_id), n\_jobs=-1)

        model\_nn.fit(co\_data)

        indices=model\_nn.kneighbors(co\_data.loc[[self.user\_input]],return\_distance=False)

        indices = indices[0][1:]

        co\_result = []

        for i in indices:

            for j in watched\_vid[i][:users\_latest\_vid]:

                if j not in co\_result:

                    if co\_data.\_get\_value(index=self.user\_input,col=j)!=1:

                        co\_result.append(j)

            if len(co\_result)>=self.n\_output:

                break

        return co\_result

    def combine(self,content,collab):

        tot\_set = []

        for i in collab:

            if len(tot\_set)>=(self.n\_output//2):

                break

            if i not in tot\_set:

                tot\_set.append(i)

        for i in content:

            if len(tot\_set)>=self.n\_output:

                break

            if i not in tot\_set:

                tot\_set.append(i)

        return tot\_set

    def display(self,result):

        cur = self.db.execute('select video\_title,video\_link,video\_thumbnail from video where video\_no in {};'.format(tuple(result)))

        print("-------------------recommendend videos(content and collaborative)------------------------")

        for i in cur:

            print(i)

    def get(self,result):

        cur = self.db.execute('select video\_title,video\_link,video\_thumbnail from video where video\_no in {};'.format(tuple(result)))

        res=[]

        for i in cur:

            res.append(list(i))

        return res

# video = Recommend('dataset.db', 2, 10)

# content = video.content()

# collab = video.collaborate()

# result = video.combine(content,collab)

# video.display(result)