گزارش پروژه پایتون و تحلیل‌ها

نگارندگان:

مهرداد مرادی 96104568

گلبرگ دخانی شبستری 97103316

بهار لنگری 97110409

غزل قلی زاده 96110045

رامان ابراهیمی 96103993

محمدحسین نوریان 96104665

دستورکار

کد هر بخش در اسکریپت پایتون به ترتیب کامنت گذاری شده است. در ادامه به توضیح هر بخش می پردازیم.

بخش صفر

```
#importing packages
import pandas as pd
import numpy as np
import re
import xlsxwriter
from sklearn import linear_model
#reading data
spotify = pd.read_excel('Spotify.xlsx')
```

در بخش اول، کتابخانه‌های pandas و numpy را فراخوانی می‌کنیم تا بتوانیم داده‌ها را بخوانیم و در ادامه بر روی آن‌ها کار انجام دهیم. همچنین کتابخانه‌های مورد نیاز برای نوشتن خروجی در اکسل و مدل رگرسیون را فراخوانی می‌کنیم.

در بخش دوم نیز، داده‌ها را می‌خوانیم و در یک دیتا فریم به نام spotify ذخیره می‌نماییم.

بخش اول

```
g = spotify['Genre']
#g has just one column
#this is for writing in excell
writer = pd.ExcelWriter('cis_project.xlsx', engine='xlsxwriter')
for i in range(len(g)):
    shelp=g[i].lower()
```

```
    if re.match('.*pop$', shelp):
        spotify.loc[i,'Genre']="Pop"
    elif re.match('.*hip hop$', shelp):
        spotify.loc[i,'Genre']="Hip Hop"
    elif re.match('.*rock$', shelp):
        spotify.loc[i,'Genre']="Rock"
    elif re.match('.*rap$', shelp):
        spotify.loc[i,'Genre']="Rap"
    elif re.match('.*\w\s\w.*', shelp):
        spotify.loc[i,'Genre']="Other"
    else:
        spotify.loc[i,'Genre']=g[i].capitalize()
output1=spotify["Genre"]
```

توضیحات: از آنجا که باید بتوان نام ژانر را چه با حروف کوچک و چه با حروف بزرگ شناسایی کرد، از تابع lower برای تبدیل همه حروف کلمه به حروف کوچک استفاده کردیم. سپس برای تبدیل نام ژانرهای آهنگ‌ها به فرمت استاندارد برای هر قسمت ریجکسی نوشته و تطبیق نام ژانر هر ریکورد را بعد از کوچک سازی حروف با فرمت مربوطه چک کرده و و تغییرات لازم را در آن ریکورد اعمال کرده‌ایم. گذاشتن علامت دلار در ریجکس به معنای آن است که کلمه قبل از این علامت در ریجکس باید در انتهای کلمه بیاید.

بخش دوم

```
#second Quesion
#first part: dealing with feat. in artists
for i in range(-1,602):
    j=i+1
    if "feat" in spotify.loc[j,'Artist']:
```

```
        names=spotify.loc[j,'Artist']
        names=names.replace('feat.','+')
        spotify.loc[j,'Artist']=names
#second Question
#second part: dealing with feat. in Title
for i in range(-1,602):
    if type(spotify.loc[i+1,'Title'])==str:
        if "feat" in spotify.loc[i+1,'Title']:
            names=spotify.loc[i+1,'Title']
            names=names.split(' (feat. ')
            spotify.loc[i+1,'Title']=names[0]#title column
            feat_artist=names[1][:-1]#artist column
            spotify.loc[i+1,'Artist']=spotify.loc[i+1,'Artist']+' + '+ str(feat_artist)


output2=spotify["Artist"]
```

توضیح الگوریتم این بخش:

برای بخش اول سوال، در یک حلقه از سطر اول تا سطر آخر را بررسی می‌کنیم و با یک شرط بررسی می‌کنیم که آیا در ستون مربوط به artist کلمه‌ی feat وجود دارد یا خیر که اگر وجود داشت، آن را با یک + عوض کرده و متن اصلاح شده را در همان سلول در ستون artist قرار می‌دهیم و بدین صورت، اصلاح مد نظر صورت می‌گیرد.

در بخش دوم که لازم است اصلاحاتی در ستون title ستون بگیرد، در یک حلقه به طور مشابه از سطر اول تا آخر را در نظر می‌گیریم، سپس در هر حلفه، به این علت که در بعضی سطرها مقدار title به صورت عددی بود، ابتدا بررسی می‌کنیم که مقدار مورد نظر، آیا به صورت رشته (string) است یا خیر، اگر بود بررسی اصلی خود را به صورت زیر آغاز می‌کنیم:

اگر در سلول مورد نظر، عبارت feat وجود داشت، آن مقدار را در یک متغیر دیگر ریخته و با دستور split نام خواننده‌ی مهمان و نام آهنگ را جدا می‌کنیم؛ سپس، نام آهنگ را در همین سلول جاگذاری کرده و در نهایت در ستون artist نام خواننده‌ی اصلی را برداشته و با ترکیب کردن آن با نام خواننده‌ی مهمان با استفاده از +، نام خوانندگان را نیز اصلاح می‌کنیم.

بخش سوم

```
output3=spotify.describe(include='all')

output3.to_excel(writer,sheet_name="Describe")
```

توضیحات: در این قسمت برای نمایش اطلاعات همه ستون‌ها و مقادیر عددی و غیر عددی در داخل پرانتز describe مقدار پارامتر مربوطه را به all تغییر دادیم.

بخش چهارم

```
#fourth Question
q4=spotify.loc[spotify.bpm>0.9*206]

q4=q4.sort_values(by=['bpm'],ascending=False)

output4=q4[['Title','Artist','Genre','bpm','Popularity']]

output4.to_excel(writer,sheet_name="bpm")
```

توضیحات: در این بخش ابتدا همه‌ی سطرهایی که مقدار bpm آن‌ها از 0.9 مقدار حداکثر (که در دیدی که از داده‌ها در بخش سوم گرفته بودیم 206 بود) بیشتر باشد را جدا کرده و به صورت یک دیتا فریم (data frame) جدا می‌کنیم.

سپس این سطرها را به صورت نزولی بر اساس bpm مرتب می‌کنیم و در نهایت ستون‌های خواسته شده رو جدا کرده و به صورت اکسل ذخیره می‌کنیم.

بخش پنجم

```
db=[0,0,0,0,0,0,0,0,0,0]

dbt=[0,0,0,0,0,0,0,0,0,0]

for i in range(len(spotify)):
```

```
    if spotify["Year"][i]==2010:
        db[0]=db[0]+spotify["dB"][i]
        dbt[0]=dbt[0]+1
    elif spotify["Year"][i]==2011:
        db[1]=db[1]+spotify["dB"][i]
        dbt[1]=dbt[1]+1
    elif spotify["Year"][i]==2012:
        db[2]=db[2]+spotify["dB"][i]
        dbt[2]=dbt[2]+1
    elif spotify["Year"][i]==2013:
        db[3]=db[3]+spotify["dB"][i]
        dbt[3]=dbt[3]+1
    elif spotify["Year"][i]==2014:
        db[4]=db[4]+spotify["dB"][i]
        dbt[4]=dbt[4]+1
    elif spotify["Year"][i]==2015:
        db[5]=db[5]+spotify["dB"][i]
        dbt[5]=dbt[5]+1
elif spotify["Year"][i]==2016:
        db[6]=db[6]+spotify["dB"][i]
        dbt[6]=dbt[6]+1
    elif spotify["Year"][i]==2017:
        db[7]=db[7]+spotify["dB"][i]
        dbt[7]=dbt[7]+1
```

```
    elif spotify["Year"][i]==2018:
        db[8]=db[8]+spotify["dB"][i]
        dbt[8]=dbt[8]+1
    elif spotify["Year"][i]==2019:
        db[9]=db[9]+spotify["dB"][i]
        dbt[9]=dbt[9]+1


d = {'Year': [2010,2011,2012,2013,2014,2015,2016,2017,2018,2019], 'db_mean': [db[0]/dbt[0],db[1]/dbt[1],db[2]/dbt[2],db[3]/dbt[3],db[4]/dbt[4],db[5]/dbt[5],db[6]/dbt[6],db[7]/dbt[7],db[8]/dbt[8],db[9]/dbt[9]]}
db_mean = pd.DataFrame(data=d)
output5=db_mean.sort_values(by=['db_mean'],ascending=False)
output5.to_excel(writer,sheet_name="db_mean")
```

توضیحات: در هر مرحله به ازای هر ریکورد از یک سال مشخص، تعداد ریکوردهای مربوط به آن سال را یکی افزوده و نیز متغییر جمع db های مربوط به آن سال را تا این لحظه با عدد ریکورد جدید جمع کرده‌ایم. بدین ترتیب به ازای هر سال مشخص تعداد ریکوردها و جمع تمام مقادیر db به دست آمده است. سپس مقادیر میانگین را محاسبه کرده و آنها را به دیتافریم تبدیل کرده‌ایم و پس از آن بر اساس مقادیر میانگین دیتافریم را سورت نزولی کرده‌ایم. لازم به ذکر است می‌توانستیم تنها از دستورات زیر استفاده کنیم که general نیز می‌باشد.

```
output5=spotify.groupby('Year').mean()
[['dB']].sort_values('dB',ascending = False)
```

بخش ششم

```
#Question 6
#first: we create a list of all artists in songs (it has all song's artists)
artists_list=[]
for i in range(603):
  if('+' in spotify.loc[i+1,'Artist']):
    artists=spotify.loc[i+1,'Artist']
    artists=artists.split(" + ")
    artists_list=artists_list+artists
  else:
    artists=spotify.loc[i+1,'Artist']
    artists_list.append(artists)

#now we creat a dictionary of this list, keys are artists and values are the number of their songs in spotify dataset!
artists_dict={}
for c in range(len(artists_list)):
    artists_dict[artists_list[c]] = artists_list.count(
        artists_list[c]
        )
#now, we creat data frame with two columns: Artist and number of presence in s ongs
df=pd.DataFrame(artists_dict.items(),columns=["Artist",'#no'])
#Question 6 continue
Best_artists=df.loc[df['#no']>2]
```

```
Best_artists=Best_artists.sort_values(by="#no")

output6=Best_artists.copy()
output6.to_excel(writer,sheet_name="Best_Artists")
```

توضیحات: در بخش اول کد، ابتدا یک لیست خالی درست می‌کنیم؛ سپس در یک حلقه، تام سطرهای دیتاست را در نظر گرفته و در ستون artist نام هر خواننده‌ای که دیدیم، آن را به لیست اضافه می‌کنیم به این صورت که اگر + وجود داشت روی + جداسازی کرده و لیست حاصل را به کلی اضافه می‌کنیم و اگر + وجود نداشت، همان تک خواننده را به این لیست اضافه می‌کنیم.

در بخش دوم، روی لیست حاصل یک حلقه تعریف می‌کنیم و برای هر آیتمی که در آن وجود دارد، آن ایتم (نام خواننده) را به عنوان کلید و تعداد تکرار آن در لیست را به عنوان مقدارش در دیکشنری، به دست می‌آوریم و نتیجه را به صورت یک دیکشنری به دست می‌آوریم.

در بخش بعدی، دیکشنری به دست آمده را به یک دیتا فریم تبدیل می‌کنیم، به این صورت که مقادیر کلید را در ستونی به نام Artist و مقادیر مقدار را در ستون #no ذخیره می‌کنیم که تعداد تکرار را می‌دهد.

در نهایت در بخش آخر، طبق خواسته‌ی سوال، هنرمندانی که تعداد تکرار بالاتر از 2 داند را جدا کرده و در یک دیتا فریم ذخیره می‌کنیم. این دیتا فریم را بر اساس تعداد تکرار به صورت صعودی مرتب کرده و در خود ذخیره می‌کنیم و بدین صورت خواسته‌ی سوال به دست می‌آید.

در نهایت، خروجی مورد نظر را در اکسل ذخیره می‌کنیم.

بخش هفتم

```
#Question 7

#first part: Hit_Songs

Hit_Songs=spotify.loc[spotify['Popularity']>=80]

Hit_Songs.to_excel(writer,sheet_name="Hit_Songs")
```

#Question 7 continue

```
Hit_Songs_Count=Hit_Songs.groupby(by='Year').Year.count()
Hit_Songs_Count.to_excel(writer,sheet_name="Hit_Songs_Count")
```

توضیحات: در قسمت اول، hit songs شناسایی شده‌اند به این صورت که آهنگ‌هایی که محبوبیت بالای 80 داشتند را جدا نمودیم و در یک دیتا فریم ذخیره نمودیم و به صورت اکسل خروجی می‌گیریم.

در بخش بعدی، hit songs را روی سال طبقه بندی کردیم و در هر سال شمارش انجام دادیم و در یک دیتا فریم، حاصل را ذخیره کردیم و به صورت اکسل مد نظر سوال، ذخیره می‌کنیم.

بخش هشتم

```
lm = linear_model.LinearRegression()
y=np.array(spotify["Popularity"])
x=np.array(spotify["Accousticness"])

model1 = lm.fit(x.reshape(-1, 1),y)
coef=list(model1.coef_)
print("{0:.4f}*x+{1:.4f}=y".format(coef[0],model1.intercept_))
spotify.to_excel(writer,sheet_name="data")
```

توضیحات: مدل رگرسیون خطی‌ای بین محبوبیت و آکوستیکنس فیت کرده‌ایم ، که نتایج آن به شرح زیر است.

```
0.0187*x+66.2533=y
```

با توجه به مثبت بودن شیب این معادله ، جواب سوال شما بله است، یعنی به نظر می‌رسد با افزایش آکوستیکنس محبوبیت هر آهنگ نیز افزایش می‌یابد.