MUSICAL SONGS ANALYSIS

A data analysis and visualization project

Spotify is a Swedish audio streaming and media services provider founded in April 2006. It is the world’s largest music streaming service provider and has over 381 million monthly active users, which also includes172 million paid subscribers. We will be exploring and qualify data about music and drawing valuable insights

This dataset contains information regarding song’s genre, artists name, tracks name, songs popularity, acousticness, danceability, loudness, artist name and id.

Project Scope

In this project we are going to find the most listening songs, most rated songs, popularity of songs, year wise songs distribution, correlation between variables, remove empty values, merge two csv files, different graphs and plots to show comparison and relation between various fields and columns of dataset, duration and can predict the upcoming songs popularity and success ration.

Link to Dataset

<https://www.kaggle.com/datasets/zaheenhamidani/ultimate-spotify-tracks-db>

<https://www.kaggle.com/datasets/lehaknarnauli/spotify-datasets?select=artists.csv>

Queries

1.find the top 5 most popular songs

2. find out artist with most danceability songs

3. find the most instrumentalness songs

4.find out duration of songs in different genres.

5.find out the total no of songs per year

6.transform the release date(dd/mm/yy) as showing only year(yy)

7.find correlation heatmap between any two variables

8.convert duration from milliseconds to seconds only

9.find out the relation between loudness and energy

### 10. Creating Visualization with Correlation Using Pearson Method

### Importing libraries

import numpy as np

from sklearn import datasets

import matplotlib.pyplot as plt

import pandas as pd

import seaborn as sns

### Reading csv file1

a1=pd.read\_csv('C:/Users/pc/OneDrive/Desktop/DAV/tracks.csv')

a1.head()

### 

### Reading csv file2

a2=pd.read\_csv('C:/Users/pc/OneDrive/Desktop/DAV/SpotifyFeatures.csv')

a2.head()

### 

### Finding null values

pd.isnull(a1).sum()

### 

### Counting total null values

pd.isnull(a1).sum().sum()

### 

### Information about database

a1.info

### 

### Sorting according to release date by this year to last years

sort\_a1=a1.sort\_values('release\_date’, ascending=False).head(8)

sort\_a1

### 

### Descriptive statistics of Spotify tracks

a1.describe().transpose()

### 

### Top 10 most popular songs

most\_popular=a1.query('popularity>90',inplace=False).sort\_values('popularity',ascending=False)

most\_popular[:10]

### 

### Indexing according to release date

a1.set\_index("release\_date",inplace=True)

a1.index=pd.to\_datetime(a1.index)

a1.head(8)

### 

### Finding artist at given index

a1[["artists"]].iloc[15]

### 

### Changing duration from milliseconds to seconds

a1["duration"]=a1["duration\_ms"].apply(lambda x:round(x/1000))

a1.drop("duration\_ms",inplace=True,axis=1)

a1.duration.head()

### 

### Correlation heatmap between variables using Pearson method

corr\_df=a1.drop(["key","mode","explicit"],axis=1).corr(method="pearson")

plt.figure(figsize=(14,6))

heatmap=sns.heatmap(corr\_df,annot=Truevmin=-1,vmax=1,center=0,cmap="inferno",linewidths=1,linecolor="black")

heatmap.set\_title("correlation heatmap between variables")

heatmap.set\_xticklabels(heatmap.get\_xticklabels(),rotation=67)

### 

### 

### Making sample

sample=a1.sample(int(0.005\*len(a1)))

print(sample)

print(len(sample))

### 

### Correlation between loudness and energy

plt.figure(figsize=(10,6))

sns.regplot(data=sample,y="loudness",x="energy",color="c").set(title="loudness vs energy correlation")

### 

### Relation between popularity and correlation

plt.figure(figsize=(10,6))

sns.regplot(data=sample,y="popularity",x="liveness",color="m").set(title="popularity vs liveness correlation")

### 

transform the release date(dd/mm/yy) as showing only year(yy)

a1['dates']=a1.index.get\_level\_values('release\_date')

a1.dates=pd.to\_datetime(a1.dates)

years=a1.dates.dt.year

years

### 

### No of songs per year

sns.distplot(years).set(title="number of songs per year")

### 

### Year vs songs duration

total\_dur=a1.duration

fig\_dims=(20,7)

fig,ax=plt.subplots(figsize=fig\_dims)

fig=sns.barplot(x=years,y=total\_dur,ax=ax,errwidth=True).set(title="year vs duration")

plt.xticks(rotation=60)

### 

### 

### Line graph between year and duration

total\_dur=a1.duration

sns.set\_style(style="whitegrid")

fig\_dims=(10,5)

fig,ax=plt.subplots(figsize=fig\_dims)

fig=sns.lineplot(x=years,y=total\_dur,ax=ax).set(title="year vs duration")

plt.xticks(rotation=60)

### 

### 

plt.title("duration of songs in different genres")

sns.color\_palette("rocket",as\_cmap=True)

sns.barplot(y="genre",x="duration\_ms",data=a2)

plt.xlabel("duration in milli seconds")

plt.ylabel("genres")

### 

### 

sns.set\_style(style="darkgrid")

plt.figure(figsize=(8,5))

famous=a2.sort\_values("popularity",ascending=False).head(10)

sns.barplot(y='artist\_name',x="popularity",data=famous).set(title="top 5 artist by popularity")

### 

sns.set\_style(style="darkgrid")

plt.figure(figsize=(8,5))

famous=a2.sort\_values("popularity",ascending=False).head(10)

sns.scatterplot(y='genre',x="popularity",data=famous).set(title="top 5 genre by popularity")

### 