Spotify Data

I have great pleasure towards the Music lovers, Listeners and artists. Music is my free time to have some passive monthly income. I want to create better music by analyzing Spotify dataset and understanding music recommendation. My current process is to first process the data, then do exploratory data analysis, then do machine learning on the data, and finally deploy the model. My goal is to do a deep dive analysis of music to see what type of music, lyrical items and artistically make more profit effectively.

Spotify API documentation: https://developer.spotify.com/documentation/web-api/
Spotify Dataset: https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.preprocessing import OrdinalEncoder, OneHotEncoder, LabelEncoder
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.metrics.pairwise import cosine_similarity
import seaborn as sns
import os
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
import matplotlib
matplotlib.rcParams['figure.dpi']=150
```

In [3]:
```
import os
!ls -lrtsh ./data/*.*
```

In [4]:
```
from kaggle.api.kaggle_kaggle_authenticate import KaggleApi
!ls ~/.kaggle/*.*
```

In [5]:
```
!mkdir -p ./data
!kaggle datasets download -d maharshipandya/-spotify-tracks-dataset -p ./data/ --unzip
```

In [6]:
```
!ls -lrtsh ./data/*.*
```

In [7]:
```
pd.set_option('display.max_columns', 50)
pd.set_option('display.max_rows', 50)
df = pd.read_csv('./data/dataset.csv')
df.shape, df.dtypes, df.info(), df.describe(), df.head(10), df.tail(10)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 114000 entries, 0 to 113999
Data columns (total 21 entries):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Unnamed: 0        114000 non-null int64
 1   track_id          114000 non-null object
 2   artists           113999 non-null object
 3   album_name        113999 non-null object
 4   track_name        113999 non-null object
 5   popularity        114000 non-null int64
 6   duration_ms       114000 non-null int64
 7   explicit          114000 non-null bool
 8   danceability      114000 non-null float64
 9   energy            114000 non-null float64
 10  key               114000 non-null int64
 11  loudness          114000 non-null float64
 12  mode              114000 non-null int64
 13  speechiness       114000 non-null float64
 14  acousticness      114000 non-null float64
 15  instrumentalness  114000 non-null float64
 16  liveness          114000 non-null float64
 17  valence           114000 non-null float64
 18  tempo             114000 non-null float64
 19  time_signature    114000 non-null int64
 20  track_genre       114000 non-null object
dtypes: bool(1), int64(9), object(5), float64(6)
memory usage: 17.6+ MB
None
```

	Unnamed: 0	track_id	artists	album_name	track_name	popularity	duration_ms	explicit	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature	track_genre
0	0	5SuOikwiRyPMVoIQDJUgSV	Gen Hoshino	Comedy	Comedy	73	230666	False	0.676	0.463	1	-6.746	0	0.143	0.0322	1.01e-06	0.358	0.715	87.917	4	acoustic
1	1	4qPNDBW1i5ZZO0Q4Rry8qR	Men I Trust	Dancing Nowhere	Dancing Nowhere	55	205333	False	0.695	0.594	11	-6.316	1	0.0363	0.0572	0.000251	0.0823	0.545	150.062	4	acoustic
2	2	1VlkjyS5dOnP9oxE6MQabU	Chris Stapleton	Traveller	Tennessee Whiskey	73	204800	False	0.407	0.307	7	-10.828	1	0.0307	0.792	0.000062	0.0956	0.267	82.149	4	acoustic
3	3	3tZJ4hevrpHUWkgNW7ufOu	Chase Rice	Lifestyle of a Country Boy	Lifestyle of a Country Boy	57	199960	False	0.703	0.544	7	-8.771	1	0.0466	0.294	0	0.0601	0.632	117.008	4	acoustic
4	4	2Gz0Ovq0YZ0Ldl0Ldl0Ldl0L	Chase Rice	Ready Set Roll	Ready Set Roll	57	199960	False	0.704	0.444	7	-10.204	1	0.0291	0.265	0	0.0633	0.636	100.029	4	acoustic
5	5	7ouMVWpa8QgC6qzZa67yora	Chase Rice	Ready Set Roll	Jackson Hole	54	199893	False	0.549	0.702	2	-5.610	1	0.0351	0.0822	0	0.0875	0.456	97.032	4	acoustic
6	6	1WKMMN3xnzKXWUcSJJbmqB	Chase Rice	Ready Set Roll	Drinkin' Problem	54	199893	False	0.624	0.595	7	-7.043	1	0.0271	0.0822	0	0.0924	0.486	92.002	4	acoustic
7	7	1XJuZ9ZcnxiXlfPc9FvwBZ	Chase Rice	Ready Set Roll	Beautiful Crazy	54	199893	False	0.621	0.492	8	-7.777	1	0.0268	0.0797	0	0.0747	0.477	115.028	4	acoustic
8	8	6Vc5wAMmXdKIAM7WUoEb7N	Chase Rice	Ready Set Roll	Ready Set Roll	53	199960	False	0.704	0.444	7	-10.204	1	0.0291	0.265	0	0.0633	0.636	100.029	4	acoustic
9	9	6eyvnePUiRQJcrcK3rw2J7	Chase Rice	Ready Set Roll	Ready Set Roll (Deluxe)	53	199960	False	0.704	0.444	7	-10.204	1	0.0291	0.265	0	0.0633	0.636	100.029	4	acoustic
110000	110000	7IzJuXiwAn825Omw5qPyNS	Ariana Grande	Eternal Sunshine	Bang Bang	67	189933	False	0.718	0.723	0	-4.929	1	0.0332	0.0804	0	0.0863	0.718	99.945	4	pop
110001	110001	3BfmJa5W6uiZ4a0Vj6CNxG	Morgan Wallen	One Thing At A Time	Last Night	82	148960	False	0.675	0.467	5	-7.777	0	0.0363	0.305	0	0.0962	0.316	75.009	4	pop
110002	110002	2triE12bZV30WcVxRQhG1W	Chris Lake	Lost You To Love Me	Lost You To Love Me	69	188627	False	0.741	0.667	7	-5.041	1	0.0295	0.0465	0.000244	0.353	0.239	124.008	4	pop
110003	110003	2S5b0gZ5l0Y8QWMwnQvBQe	Teddy Swims	Lost Your Love	Lost Your Love	77	239626	False	0.654	0.470	1	-5.351	1	0.0244	0.304	0	0.103	0.414	98.017	4	pop
110004	110004	00qczGi7CNKhW3giJkD7uW	Lee Brice	Love like This	Love like This	75	239080	False	0.404	0.483	9	-7.230	1	0.0317	0.449	0	0.0895	0.095	139.885	4	pop
110005	110005	1R1Squ8Emb7zTcNXxhfaQP	Luke Combs	Growin' Up	Growin' Up	71	211373	False	0.508	0.392	2	-8.777	1	0.0299	0.816	0	0.116	0.181	135.860	4	pop
110006	110006	7EJMuMnY5tCYPlCRcSaaGA	Cody Johnson	Human	Human	70	229400	False	0.429	0.526	7	-6.687	1	0.0309	0.396	0	0.0629	0.494	79.198	4	pop
110007	110007	3Ofmpqh0TxHU6dROq6Mxw2	Luke Combs	Growin' Up	Beautiful Crazy	71	187067	False	0.587	0.704	2	-5.423	1	0.0290	0.297	0	0.0747	0.687	103.967	4	pop
110008	110008	7ubJ0Ldl0Ldl0Ldl0Ldl0Ld	Ashley McBryde	Girl Going Out	Girl Going Out	69	189000	False	0.652	0.762	1	-4.788	1	0.0288	0.0201	0	0.0817	0.754	124.992	4	pop
110009	110009	7ubJ0Ldl0Ldl0Ldl0Ldl0Ld	Ashley McBryde	Girl Going Out	Girl Going Out	69	189000	False	0.652	0.762	1	-4.788	1	0.0288	0.0201	0	0.0817	0.754	124.992	4	pop

In [8]:
```
df.sample(10)
```

	Unnamed: 0	track_id	artists	album_name	track_name	popularity	duration_ms	explicit	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature	track_genre
32	32	0SGkqnVQo9KPytSri1Zfp7	Lil Peep	Hellboy	Star Shopping	65	154480	False	0.683	0.448	8	-8.771	1	0.0401	0.305	0	0.101	0.445	75.017	4	emo
33	33	1B1ZJ0Ldl0Ldl0Ldl0Ldl0Ld	Lil Peep	Hellboy	Hellboy	65	213333	False	0.583	0.542	4	-7.987	1	0.0568	0.0888	0.000025	0.351	0.351	159.814	4	emo
34	34	1Ldl0Ldl0Ldl0Ldl0Ldl0Ld	Lil Peep	Hellboy	The Kid	65	151173	False	0.743	0.431	11	-6.771	0	0.342	0.805	0	0.351	0.039	95.052	4	emo
35	35	1Ldl0Ldl0Ldl0Ldl0Ldl0Ld	Lil Peep	Hellboy	Save That Money	65	191826	False	0.653	0.462	1	-5.828	1	0.0482	0.296	0	0.138	0.482	75.023	4	emo
36	36	1Ldl0Ldl0Ldl0Ldl0Ldl0Ld	Lil Peep	Hellboy	Praise The Lord (Da Blitz)	65	172866	False	0.746	0.448	8	-5.657	1	0.0888	0.0405	0.000119	0.117	0.743	140.048	4	emo
37	37	1Ldl0Ldl0Ldl0Ldl0Ldl0Ld	Lil Peep	Hellboy	Runaway	65	151333	False	0.591	0.374	9	-9.291	1	0.0392	0.164	0.000094	0.632	0.339	159.772	4	emo
38	38	1Ldl0Ldl0Ldl0Ldl0Ldl0Ld	Lil Peep	Hellboy	Star Gaze	65	151333	False	0.737	0.415	4	-8.771	1	0.0363	0.294	0	0.207	0.414	159.809	4	emo
39	39	1Ldl0Ldl0Ldl0Ldl0Ldl0Ld	Lil Peep	Hellboy	Star Shopping	65	154480	False	0.683	0.448	8	-8.771	1	0.0401	0.305	0	0.101	0.445	75.017	4	emo

In [9]:
```
df.describe()
```

	Unnamed: 0	popularity	duration_ms	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature
count	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000
mean	56999.500000	33.238535	228031.153958	0.566800	0.641231	5.309645	-8.258960	0.637553	0.084692	0.314907	0.156651	0.213553	0.474068	122.035027	3.904035
std	32951.824374	22.305078	107422.641309	0.173542	0.251700	3.559987	5.029337	0.480709	0.105733	0.332640	0.309555	0.190475	0.259610	29.977925	0.432087
min	0.000000	0.000000	8586.000000	0.000000	0.000000	0.000000	-49.531000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	28499.750000	17.000000	174066.000000	0.456000	0.472000	2.000000	-10.013000	0.000000	0.035900	0.015100	0.000000	0.098100	0.260000	99.649000	4.000000
50%	56999.500000	35.000000	213373.000000	0.570000	0.685000	5.000000	-6.643000	1.000000	0.048900	0.169000	0.000042	0.132000	0.464000	122.017000	4.000000
75%	85499.250000	50.000000	262773.000000	0.695000	0.854000	8.000000	-4.779000	1.000000	0.084500	0.598000	0.049100	0.273000	0.683000	141.070000	4.000000
max	113999.000000	100.000000	5237295.000000	0.985000	1.000000	11.000000	4.532000	1.000000	0.965000	0.996000	1.000000	0.996000	0.995000	243.372000	5.000000

In [10]:
```
df[df.duplicated(keep=False)]
```

In [11]:
```
df.isnull().sum()
```

In [12]:
```
df[df.isnull().sum(axis=1)>0]
```

In [13]:
```
df.nunique()
```

In [14]:
```
df.drop_duplicates(inplace=True)
```

In [15]:
```
df.shape
```

In [16]:
```
df.drop(columns=['Unnamed: 0'],inplace=True)
```

In [17]:
```
df.sample(10)
```

	track_id	artists	album_name	track_name	popularity	duration_ms	explicit	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature	track_genre
32	0SGkqnVQo9KPytSri1Zfp7	Lil Peep	Hellboy	Star Shopping	65	154480	False	0.683	0.448	8	-8.771	1	0.0401	0.305	0	0.101	0.445	75.017	4	emo
33	1B1ZJ0Ldl0Ldl0Ldl0Ldl0Ld	Lil Peep	Hellboy	Hellboy	65	213333	False	0.583	0.542	4	-7.987	1	0.0568	0.0888	0.000025	0.351	0.351	159.814	4	emo
34	1Ldl0Ldl0Ldl0Ldl0Ldl0Ld	Lil Peep	Hellboy	The Kid	65	151173	False	0.743	0.431	11	-6.771	0	0.342	0.805	0	0.351	0.039	95.052	4	emo
35	1Ldl0Ldl0Ldl0Ldl0Ldl0Ld	Lil Peep	Hellboy	Save That Money	65	191826	False	0.653	0.462	1	-5.828	1	0.0482	0.296	0	0.138	0.482	75.023	4	emo
36	1Ldl0Ldl0Ldl0Ldl0Ldl0Ld	Lil Peep	Hellboy	Praise The Lord (Da Blitz)	65	172866	False	0.746	0.448	8	-5.657	1	0.0888	0.0405	0.000119	0.117	0.743	140.048	4	emo
37	1Ldl0Ldl0Ldl0Ldl0Ldl0Ld	Lil Peep	Hellboy	Runaway	65	151333	False	0.591	0.374	9	-9.291	1	0.0392	0.164	0.000094	0.632	0.339	159.772	4	emo
38	1Ldl0Ldl0Ldl0Ldl0Ldl0Ld	Lil Peep	Hellboy	Star Gaze	65	151333	False	0.737	0.415	4	-8.771	1	0.0363	0.294	0	0.207	0.414	159.809	4	emo
39	1Ldl0Ldl0Ldl0Ldl0Ldl0Ld	Lil Peep	Hellboy	Star Shopping	65	154480	False	0.683	0.448	8	-8.771	1	0.0401	0.305	0	0.101	0.445	75.017	4	emo

In [18]:
```
df.track_genre.nunique()
```

In [19]:
```
df.track_genre.sample(10)
```

In [20]:
```
df.track_genre.value_counts
```

In [21]:
```
df.popularity.plot.hist;
```

In [22]:
```
df.popularity.sort_values
```

In [23]:
```
df.popularity.sort_values.tail(10)
```

In [24]:
```
df.popularity.sort_values.tail(10).sum()
```

In [25]:
```
df[df.popularity>99]
```

In [26]:
```
df[df.popularity>99].track_genre.value_counts
```

In [27]:
```
df[df.popularity>99].track_genre.nunique(),df.track_genre.nunique()
```

In [28]:
```
df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']})
```

In [29]:
```
df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.sum()
```

In [30]:
```
df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.sum()/df.popularity.sum()
```

In [31]:
```
df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.sum()/df.popularity.count()
```

In [32]:
```
df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.sum()/df[df.popularity>0].popularity.count()
```

In [33]:
```
df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.count()/df.track_genre.nunique()
```

In [34]:
```
df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.count()/df.track_genre.nunique()*100
```

In [35]:
```
df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.count()/df.track_genre.nunique()*100,df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.count()/df.track_genre.nunique()*100
```

In [36]:
```
df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.count()/df.track_genre.nunique()*100,df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.count()/df.track_genre.nunique()*100,df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.count()/df.track_genre.nunique()*100
```

In [37]:
```
df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.count()/df.track_genre.nunique()*100,df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.count()/df.track_genre.nunique()*100,df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.count()/df.track_genre.nunique()*100,df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.count()/df.track_genre.nunique()*100
```

In [38]:
```
df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.count()/df.track_genre.nunique()*100,df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.count()/df.track_genre.nunique()*100,df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.count()/df.track_genre.nunique()*100,df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.count()/df.track_genre.nunique()*100,df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.count()/df.track_genre.nunique()*100
```

In [39]:
```
df[df.popularity>99].groupby('track_genre').agg({'popularity':['mean','min','max','sum','count','nunique']}).popularity.count()/df.track_genre.nunique()*100,
```

In [1]:
```
movies[movies_matrix.shape]
```

Out[1]:
```
array([[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
        -1.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0
```