

Name- Utkarsh.Arvind.Bondade

Gender-Male

Age- 21

City-Nagpur

Pursuing- BTECH from Shri Guru Gobind Singhji Institute of Engineering and Technology(Nanded)

Branch- Electronics and Telecommunication Engineering(Final Year)

Subject- Teachnook Datascience 2 Assignment

Topic- Exploratory Data Analysis (EDA) on Spotify Song Attributes.

Dataset link-<https://www.kaggle.com/datasets/geomack/spotifyclassification>

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

```
sns.set_style("darkgrid")
```

```
##load dataset
df=pd.read_csv('/content/spotifyAnalysis.csv.csv')
df.drop( 'Unnamed: 0', axis=1,inplace=True)
df.head()
```

```
##data cleaning
df.isna().sum()
```

1-> **Content:** The Spotify Song Attributes dataset includes information about songs' audio features and metadata, such as song name, artist name, album name, release year, duration, popularity, danceability, energy, key, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo, and more.

2-> In this dataset by using `.isna().sum()` we can see that there are no missing values in the dataset and hence our **data is cleaned**.

```

[4] import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

[5] sns.set_style("darkgrid")

[6] #Load dataset
df=pd.read_csv('/content/spotifyAnalysis.csv')
df.drop(['Unnamed: 0', axis=1,inplace=True])
df.head()

acousticness  danceability  duration_ms  energy  instrumentalness  key  liveness  loudness  mode  speechiness  tempo  time_signature  valence  target  song_title  artist
0      0.0102      0.833      204600      0.434      0.021900      2      0.1650      -8.795      1      0.4310      150.062      4.0      0.286      1      Mask Off      Future
1      0.1990      0.743      326933      0.359      0.006110      1      0.1370      -10.401      1      0.0794      160.083      4.0      0.588      1      Redbone      Childish Gambino
2      0.0344      0.838      185707      0.412      0.000234      2      0.1590      -7.148      1      0.2890      75.044      4.0      0.173      1      Xanny Family      Future
3      0.6040      0.494      199413      0.338      0.510000      5      0.0922      -15.236      1      0.0261      86.468      4.0      0.230      1      Master Of None      Beach House
4      0.1800      0.678      392893      0.561      0.512000      5      0.4390      -11.648      0      0.0694      174.004      4.0      0.904      1      Parallel Lines      Junior Boys

[7] #data cleaning
df.isna().sum()

acousticness      0
danceability      0
duration_ms       0
energy            0
instrumentalness  0
key               0
liveness          0
loudness          0
mode              0
speechiness       0
tempo             0
time_signature    0
valence           0
target            0
song_title        0
artist            0
dtype: int64

```

```
df.info()
```

```
df.shape
```

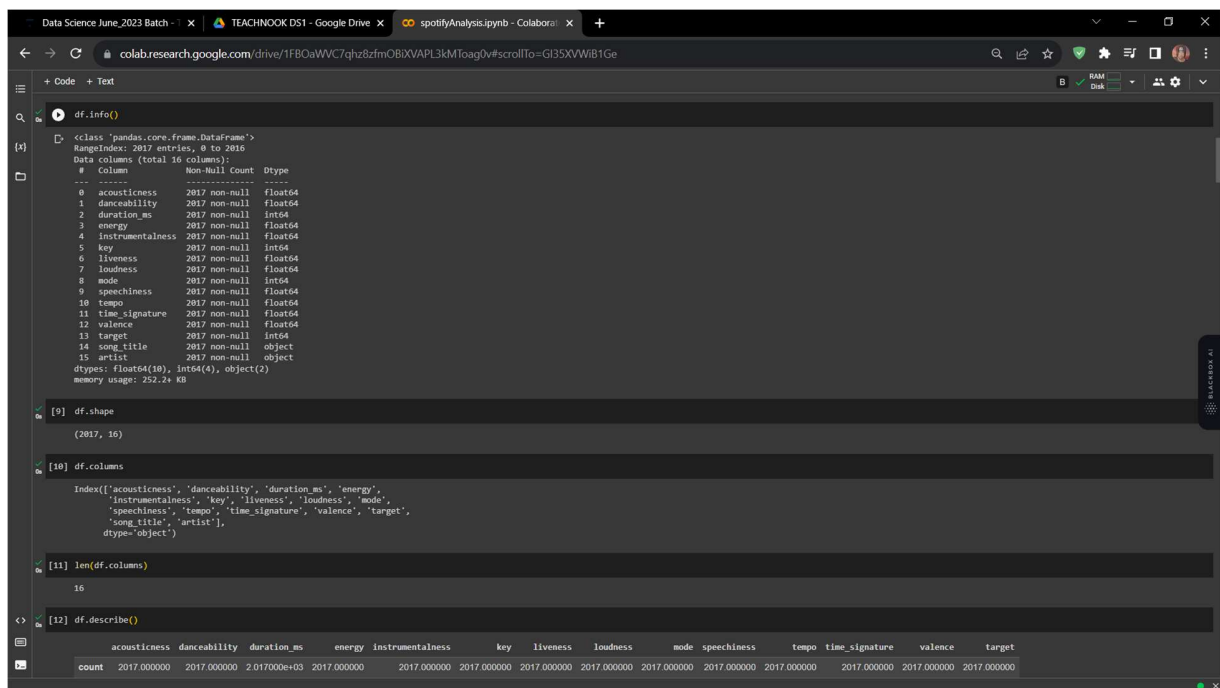
```
df.columns
```

```
len(df.columns)
```

```
df.describe()
```

3->Use cases: The dataset has been widely used for various purposes, including music recommendation systems, genre classification, sentiment analysis of songs, understanding

the characteristics of popular tracks, and exploring the relationship between audio features and listener preferences.



```
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2017 entries, 0 to 2016
Data columns (total 16 columns):
 #   Column              Non-Null Count  Dtype  
---  --
 0   acousticness        2017 non-null   float64
 1   danceability         2017 non-null   float64
 2   duration_ms         2017 non-null   int64   
 3   energy              2017 non-null   float64
 4   instrumentalness     2017 non-null   float64
 5   key                 2017 non-null   int64   
 6   liveness            2017 non-null   float64
 7   loudness            2017 non-null   float64
 8   mode               2017 non-null   int64   
 9   speechiness         2017 non-null   float64
10   tempo              2017 non-null   float64
11   time_signature      2017 non-null   float64
12   valence            2017 non-null   float64
13   target             2017 non-null   int64   
14   song_title         2017 non-null   object  
15   artist             2017 non-null   object  
dtypes: float64(10), int64(4), object(2)
memory usage: 252.2+ KB

[9] df.shape
(2017, 16)

[10] df.columns
Index(['acousticness', 'danceability', 'duration_ms', 'energy',
       'instrumentalness', 'key', 'liveness', 'loudness', 'mode',
       'speechiness', 'tempo', 'time_signature', 'valence', 'target',
       'song_title', 'artist'],
      dtype='object')

[11] len(df.columns)
16

[12] df.describe()

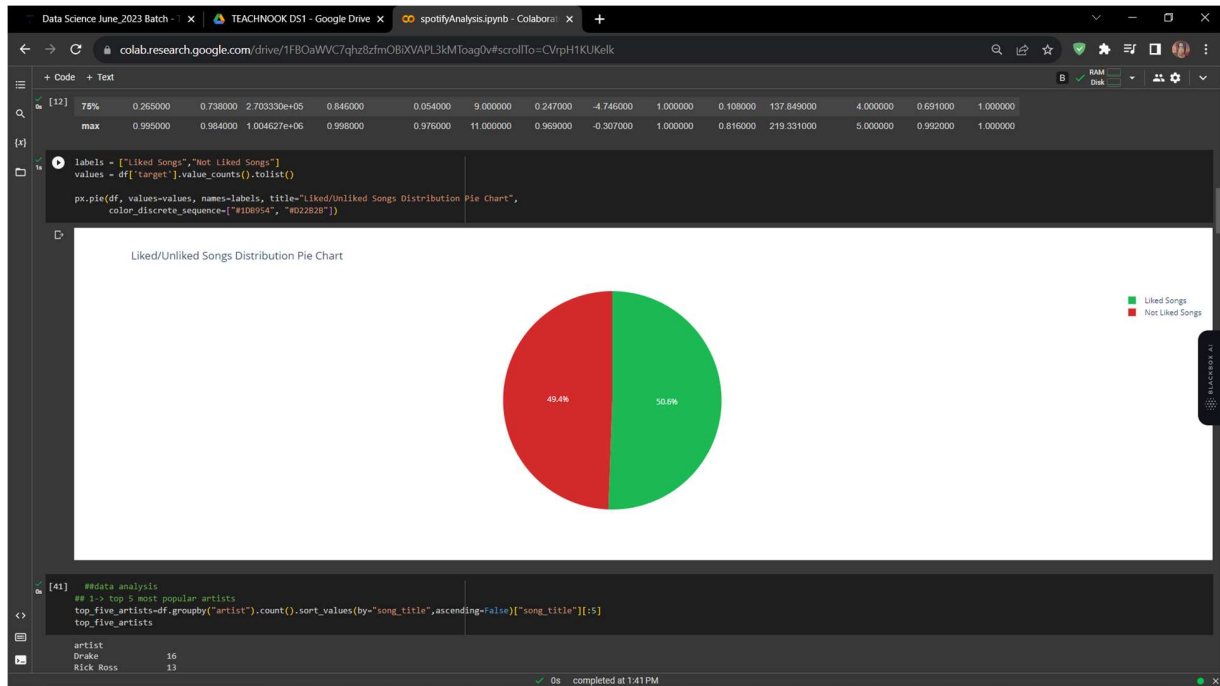
```

	acousticness	danceability	duration_ms	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_signature	valence	target
count	2017.000000	2017.000000	2.017000e+03	2017.000000	2017.000000	2017.000000	2017.000000	2017.000000	2017.000000	2017.000000	2017.000000	2017.000000	2017.000000	2017.000000

4->As we can see that by using `df.shape()` we can get the no of rows and columns and in this there are 2017-rows and 16 columns.

```
labels = ["Liked Songs", "Not Liked Songs"]
values = df['target'].value_counts().tolist()

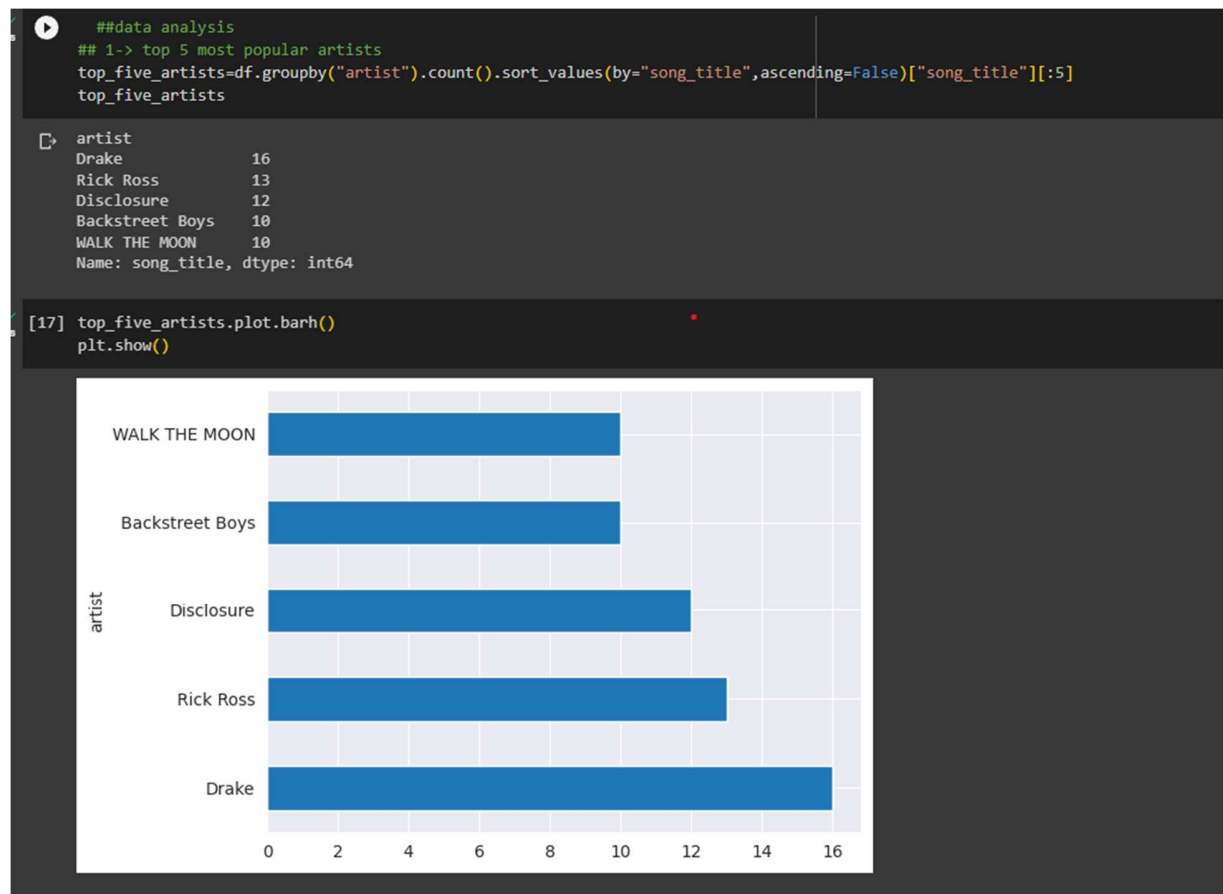
px.pie(df, values=values, names=labels, title="Liked/Unliked Songs
Distribution Pie Chart",
       color_discrete_sequence=["#1DB954", "#D22B2B"])
```



5->In this **PIE Chart** we can see the liked and disliked songs been represented in simple manner in percentage and there are **50.6 -liked songs** and **49.4 disliked songs**.

```
##data analysis
## 1-> top 5 most popular artists
top_five_artists=df.groupby("artist").count().sort_values(by="song_title",
ascending=False)["song_title"][:5]
top_five_artists
```

```
top_five_artists.plot.barh()  
plt.show()
```

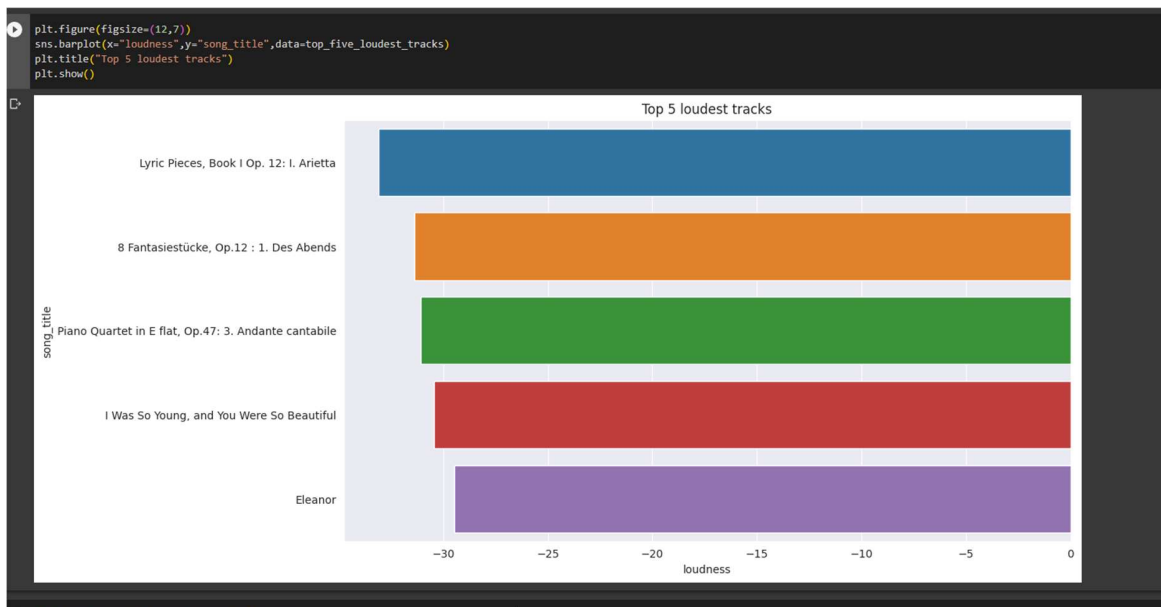


6->Top 5 most popular artists can be found using functions such as `.groupby()`, `.count()`, `.sort_values()`. And as we can see that 'Drake' is the most popular artist.

```
##2-> top 5 loudest tracks
top_five_loudest_tracks=df[["loudness","song_title"]].sort_values(by="loudness",ascending=True)[:5]
top_five_loudest_tracks
plt.figure(figsize=(12,7))
sns.barplot(x="loudness",y="song_title",data=top_five_loudest_tracks)
plt.title("Top 5 loudest tracks")
plt.show()
```

```
##2-> top 5 loudest tracks
top_five_loudest_tracks=df[["loudness","song_title"]].sort_values(by="loudness",ascending=True)[:5]
top_five_loudest_tracks
```

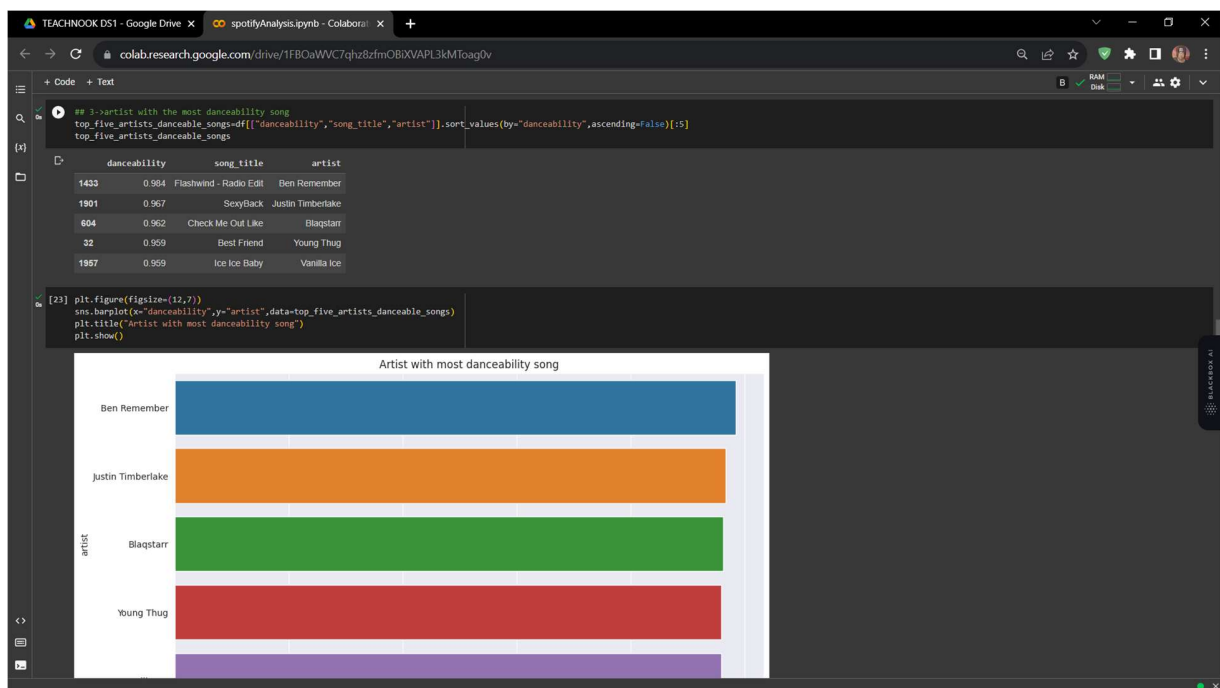
	loudness	song_title
1594	-33.097	Lyric Pieces, Book I Op. 12: I. Arietta
1596	-31.367	8 Fantasiestücke, Op.12 : 1. Des Abends
1598	-31.082	Piano Quartet in E flat, Op.47: 3. Andante can...
1531	-30.447	I Was So Young, and You Were So Beautiful
1549	-29.460	Eleanor



7-> **Top 5 loudest tracks** can also be determined and we can see that song title named 'Lyric Pieces' is the most loudest track in this dataset

```
## 3->artist with the most danceability song
top_five_artists_danceable_songs=df[["danceability","song_title","artist"]
].sort_values(by="danceability",ascending=False)[:5]
top_five_artists_danceable_songs
```

```
plt.figure(figsize=(12,7))
sns.barplot(x="danceability",y="artist",data=top_five_artists_danceable_songs)
plt.title("Artist with most danceability song")
plt.show()
```

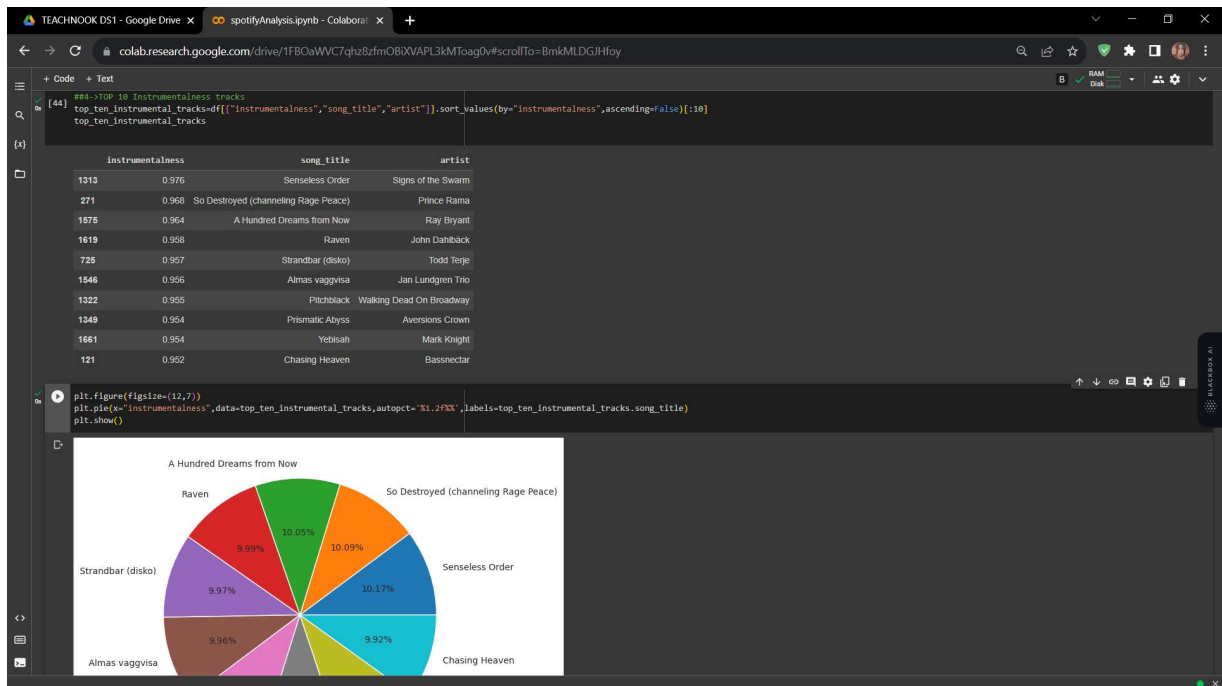


8-> We can also conclude the artist with most danceable song is 'Ben Rmember' by seeing it in the barplot.


```
##4->TOP 10 Instrumentalness tracks
```

```
top_ten_instrumental_tracks=df[["instrumentalness","song_title","artist"]]  
.sort_values(by="instrumentalness",ascending=False)[:10]  
top_ten_instrumental_tracks
```

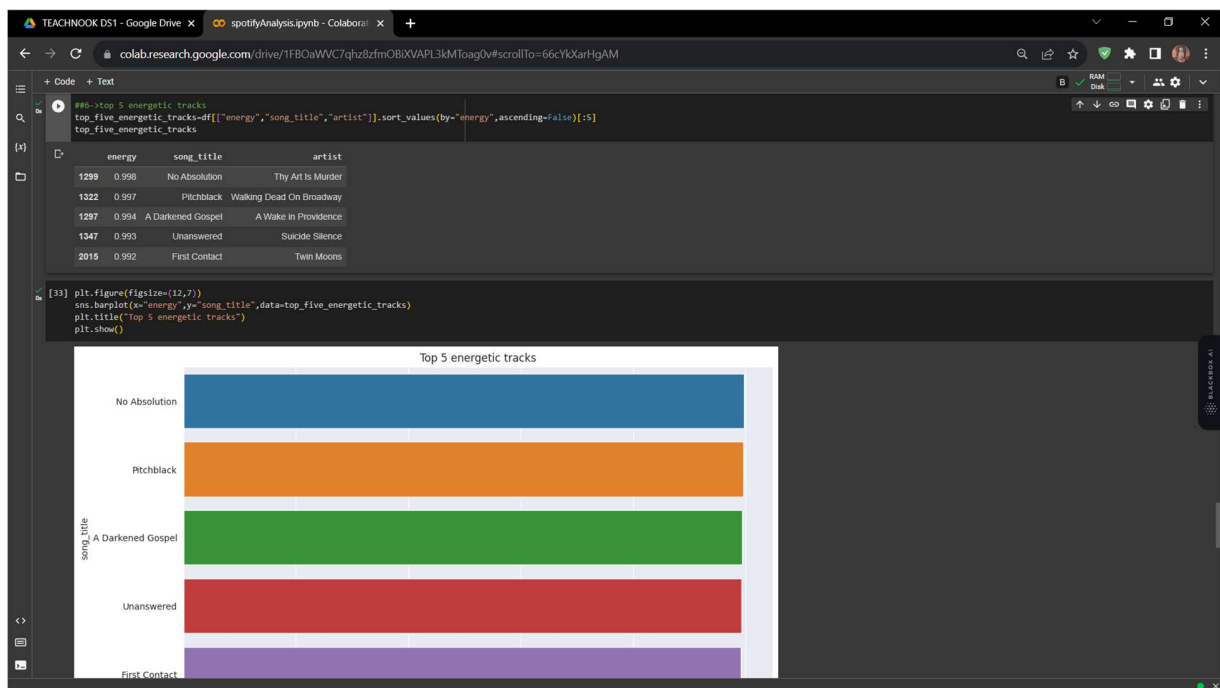
```
plt.figure(figsize=(12,7))  
plt.pie(x="instrumentalness",data=top_ten_instrumental_tracks,autopct='%1.  
2f%%',labels=top_ten_instrumental_tracks.song_title)  
plt.show()
```



9->Top 10 instrumentalness tracks can also be identified and from the pie chart it is observed that artist named 'Signs of the Swam' has the most instrumentalness track.

```
##6->top 5 energetic tracks
top_five_energetic_tracks=df[["energy","song_title","artist"]].sort_values
(by="energy",ascending=False)[:5]
top_five_energetic_tracks
```

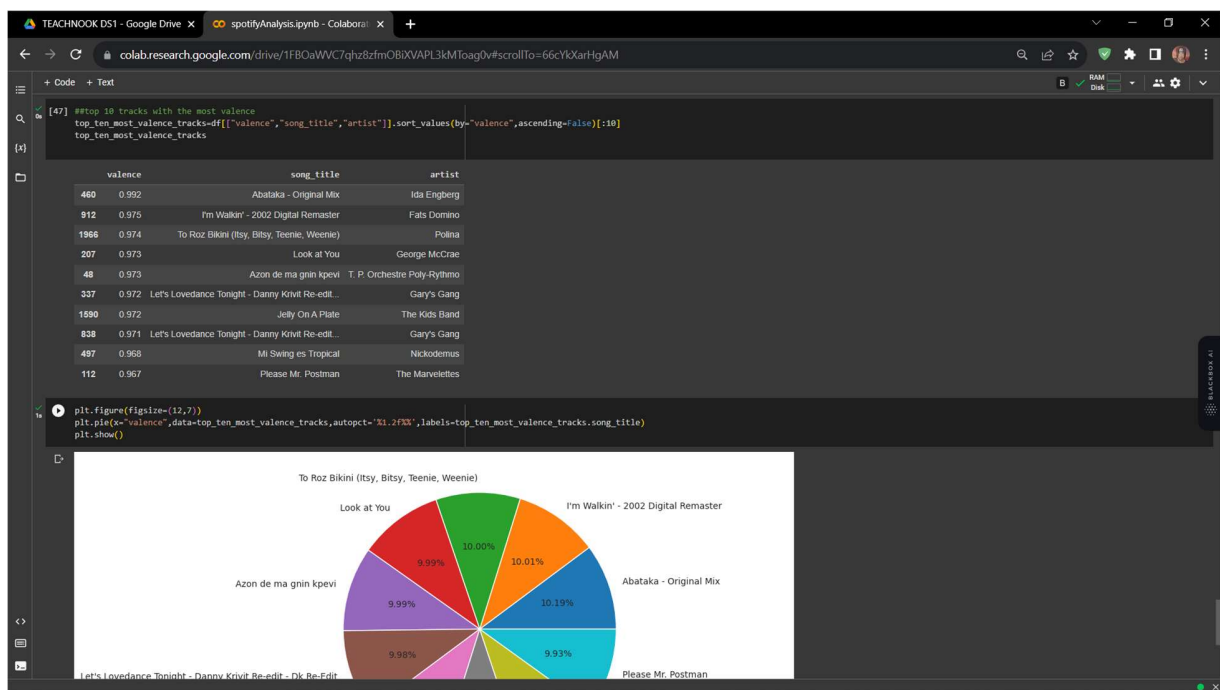
```
plt.figure(figsize=(12,7))
sns.barplot(x="energy",y="song_title",data=top_five_energetic_tracks)
plt.title("Top 5 energetic tracks")
plt.show()
```



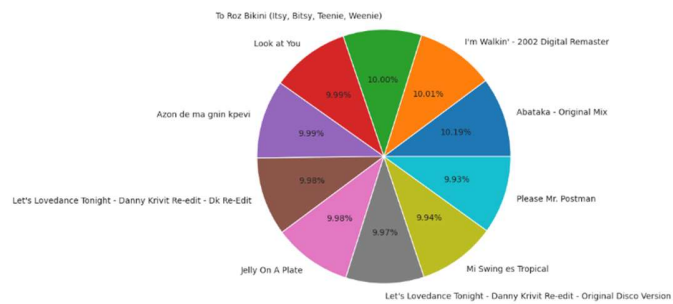
10->**Top 5 energetic songs** are given by artist 'The Art is murder', 'Walking dead On Broadway', 'A wake in Providence', 'Suicide Silence', 'Twin Moons'. And the most energetic song is 'No Absolution' by 'The art is murder'.

```
##top 10 tracks with the most valence
top_ten_most_valence_tracks=df[["valence","song_title","artist"]].sort_val
ues(by="valence",ascending=False)[:10]
top_ten_most_valence_tracks
```

```
plt.figure(figsize=(12,7))
plt.pie(x="valence",data=top_ten_most_valence_tracks,autopct='%1.2f%%',lab
els=top_ten_most_valence_tracks.song_title)
plt.show()
```



11->Top 10 tracks with **most valence** can be extracted and it is seen that the artist named 'Ida Engberg' has highest valence.



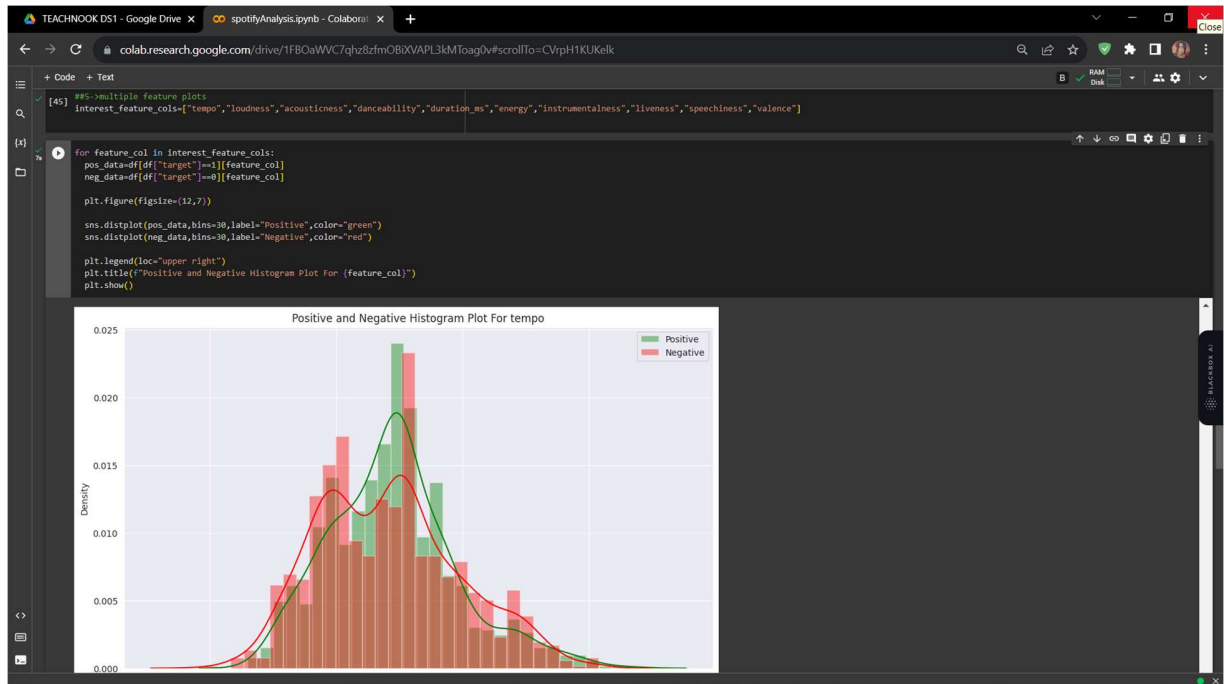
```
##5->multiple feature plots
interest_feature_cols=["tempo","loudness","acousticness","danceability","d
uration_ms","energy","instrumentalness","liveness","speechiness","valence"
]
```

```
for feature_col in interest_feature_cols:
    pos_data=df[df["target"]==1][feature_col]
    neg_data=df[df["target"]==0][feature_col]

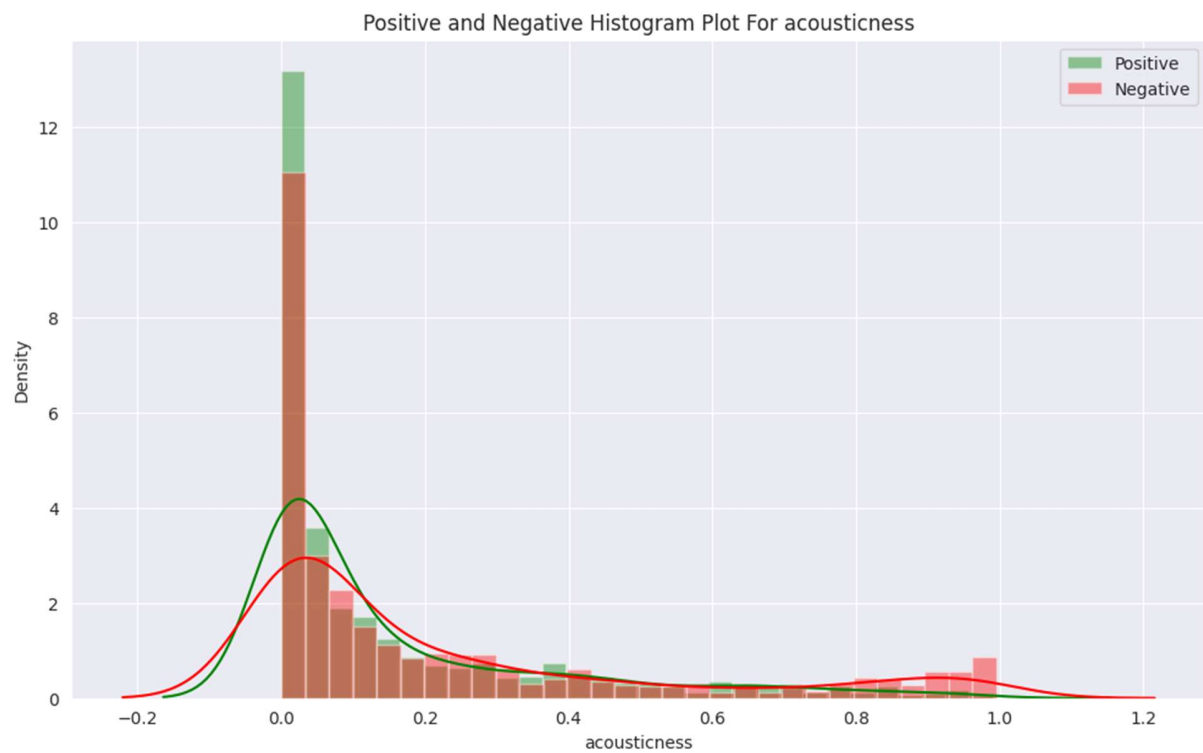
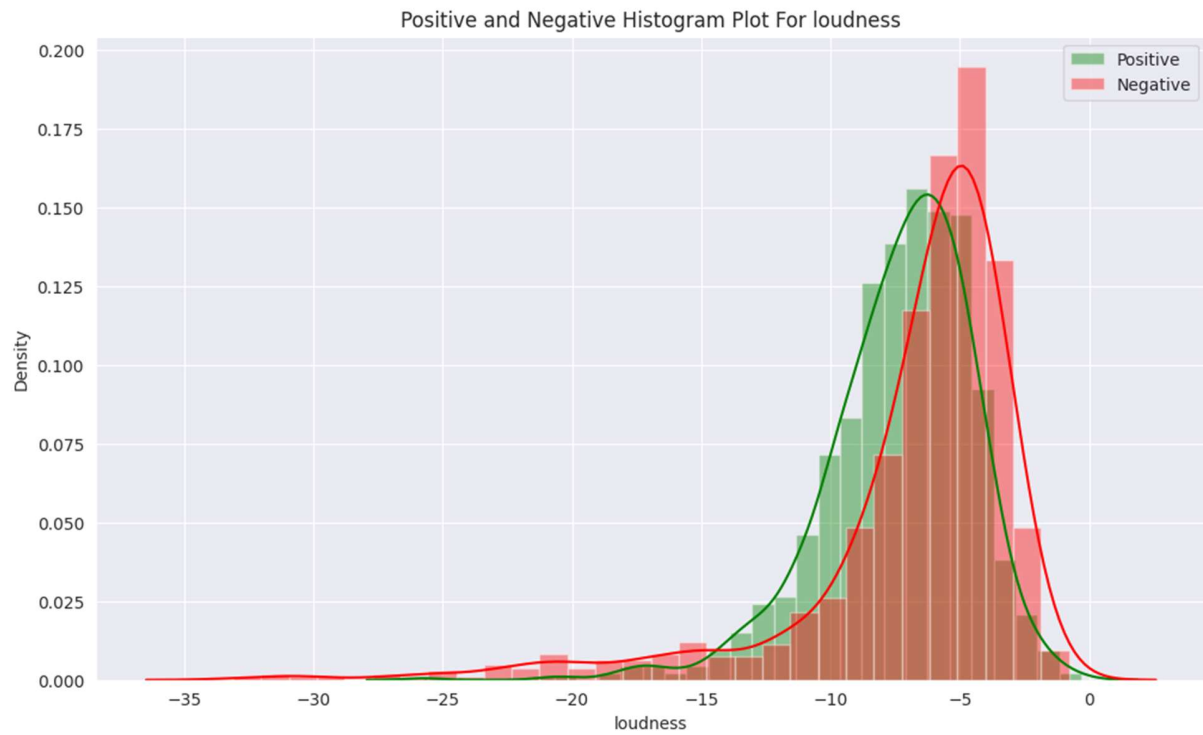
    plt.figure(figsize=(12,7))

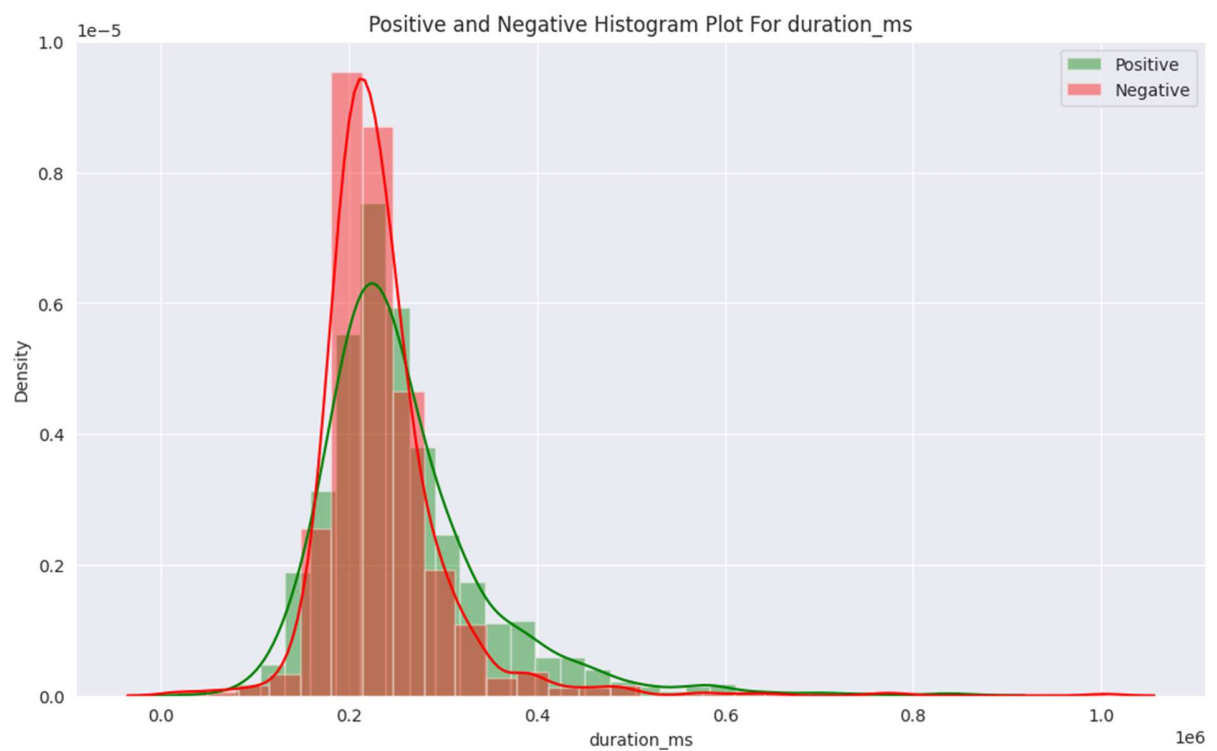
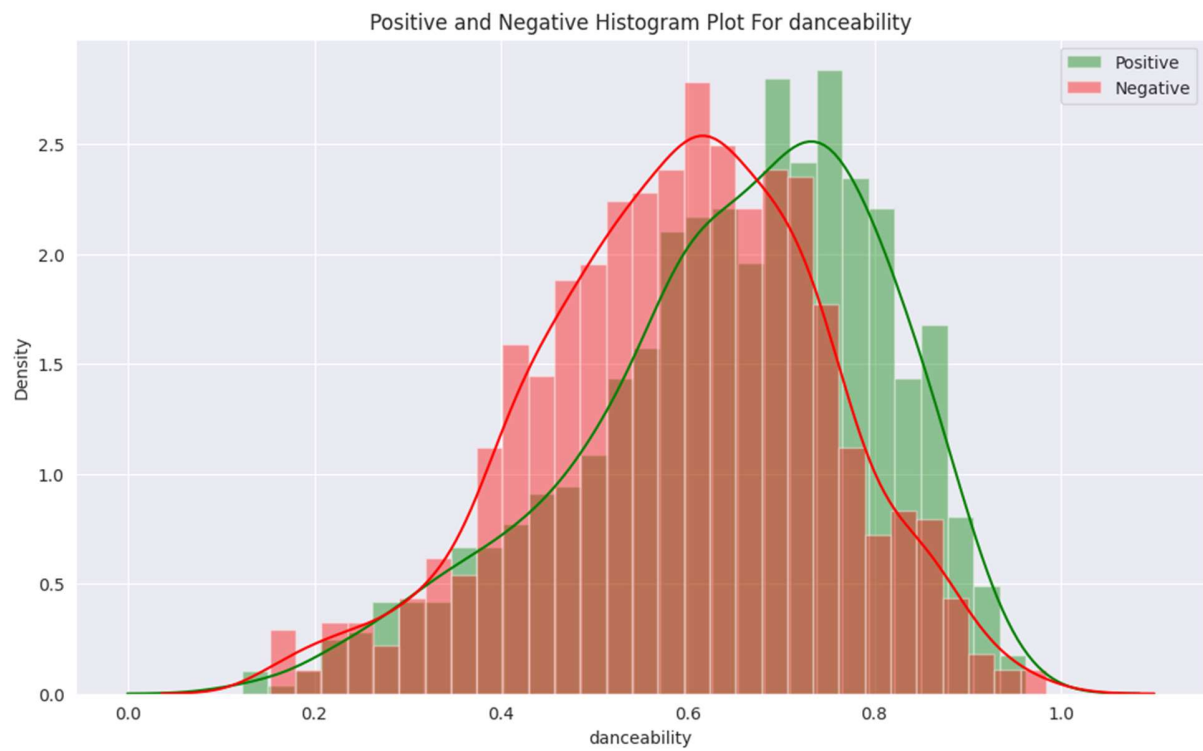
    sns.distplot(pos_data,bins=30,label="Positive",color="green")
    sns.distplot(neg_data,bins=30,label="Negative",color="red")

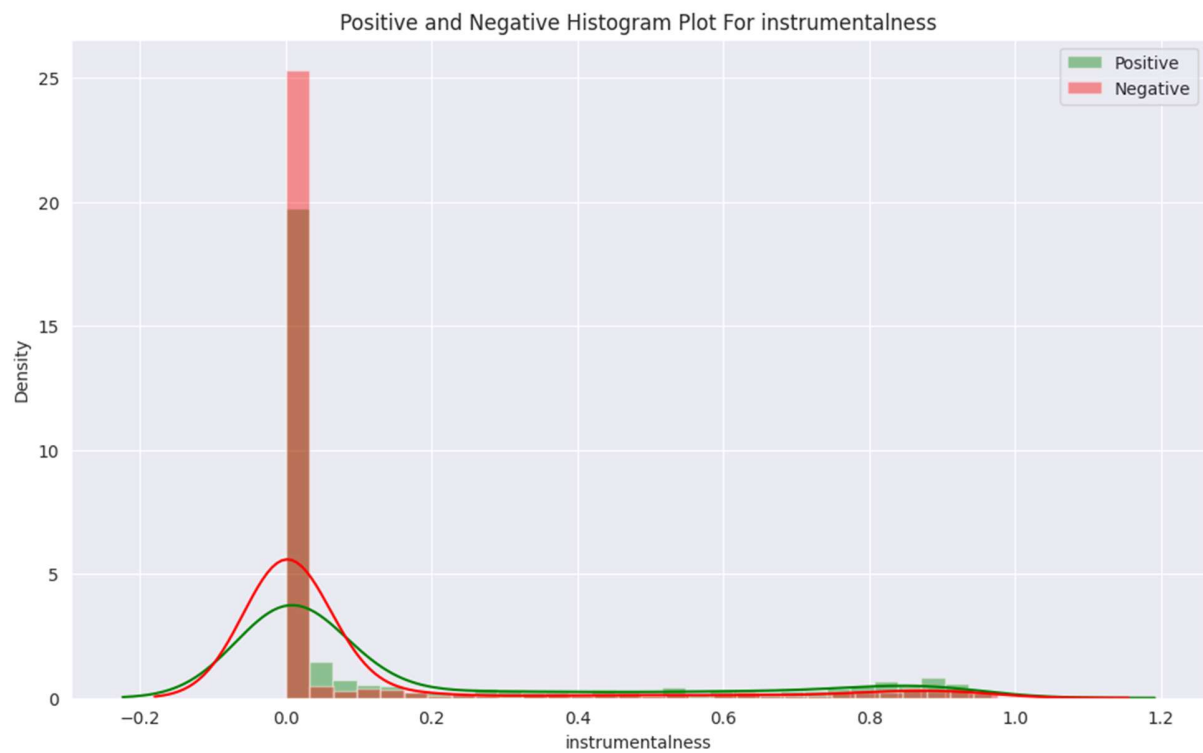
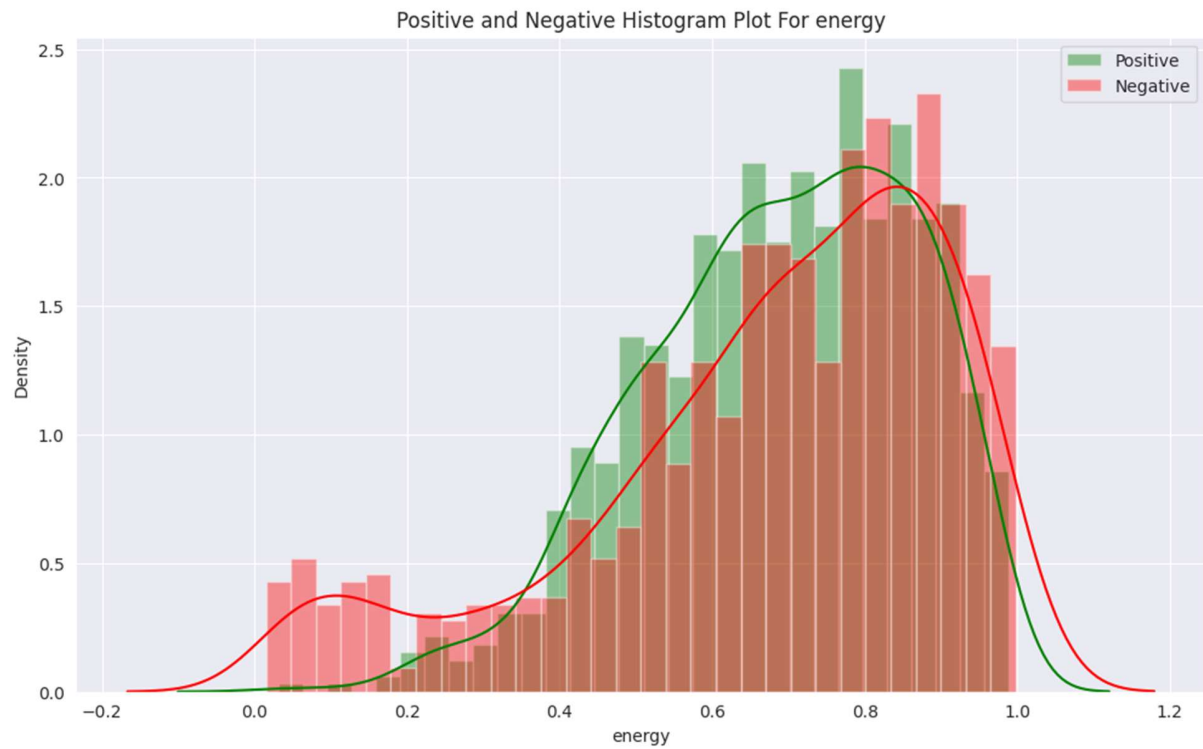
    plt.legend(loc="upper right")
    plt.title(f"Positive and Negative Histogram Plot For {feature_col}")
    plt.show()
```

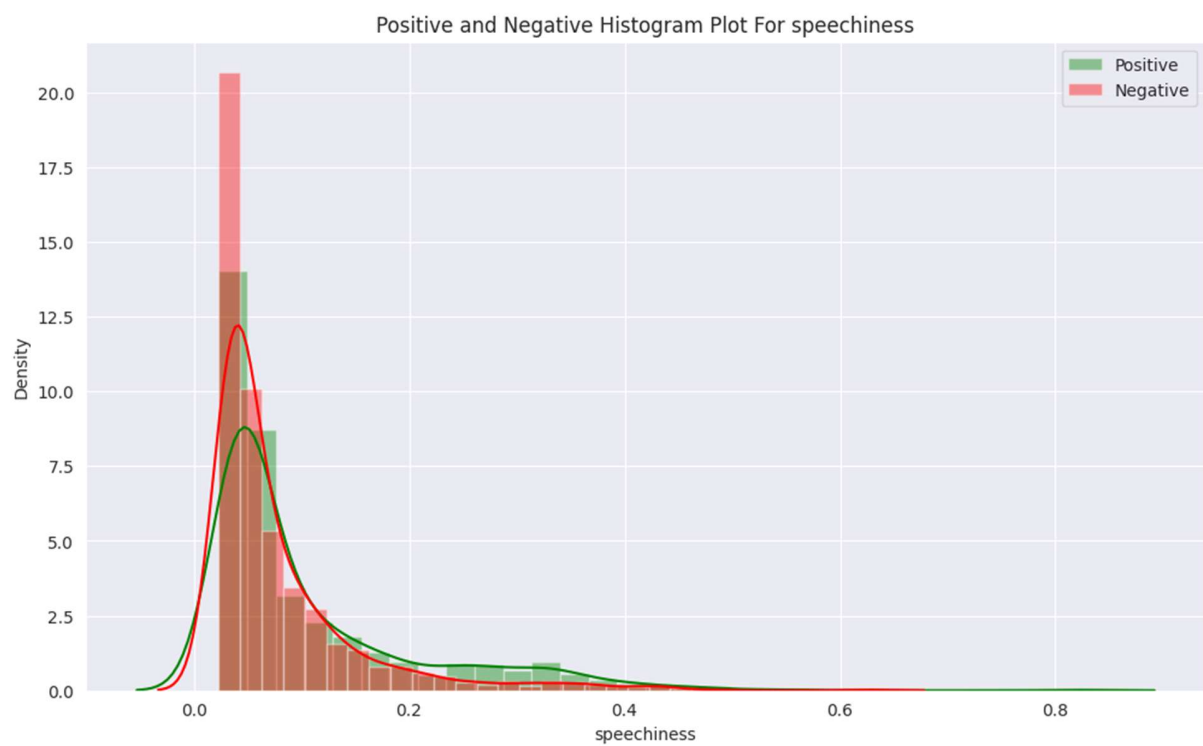
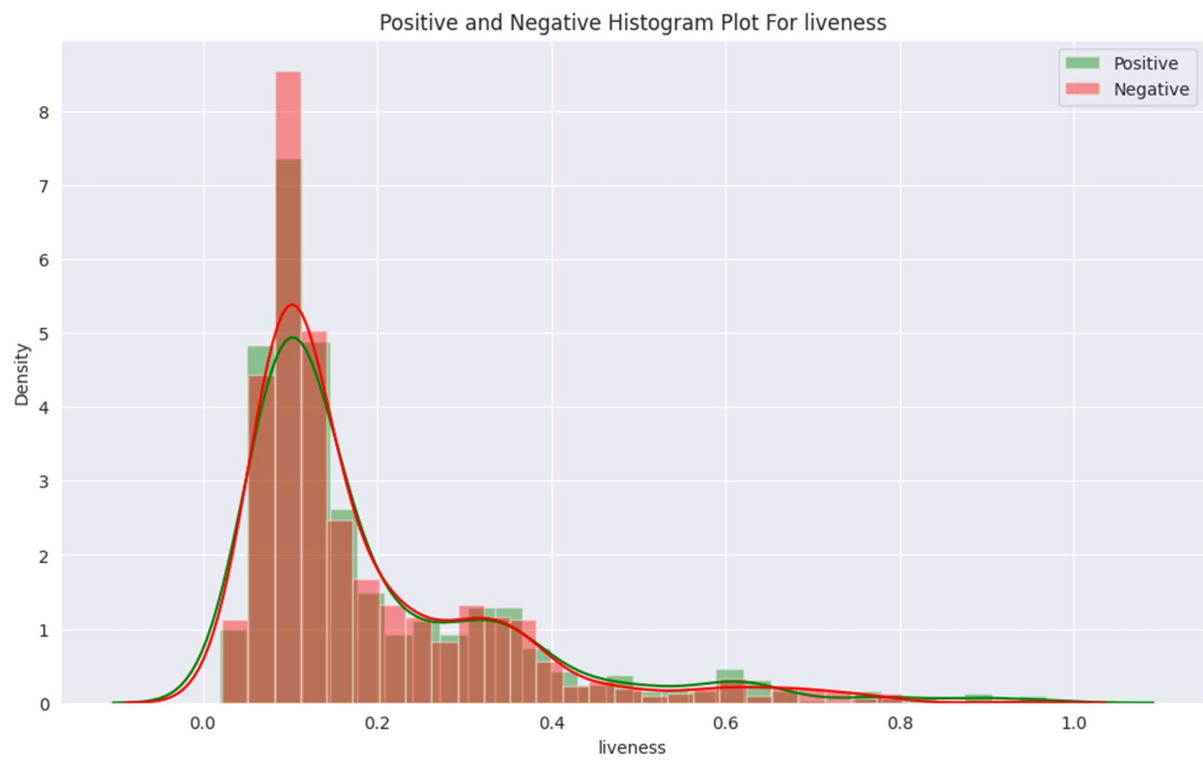


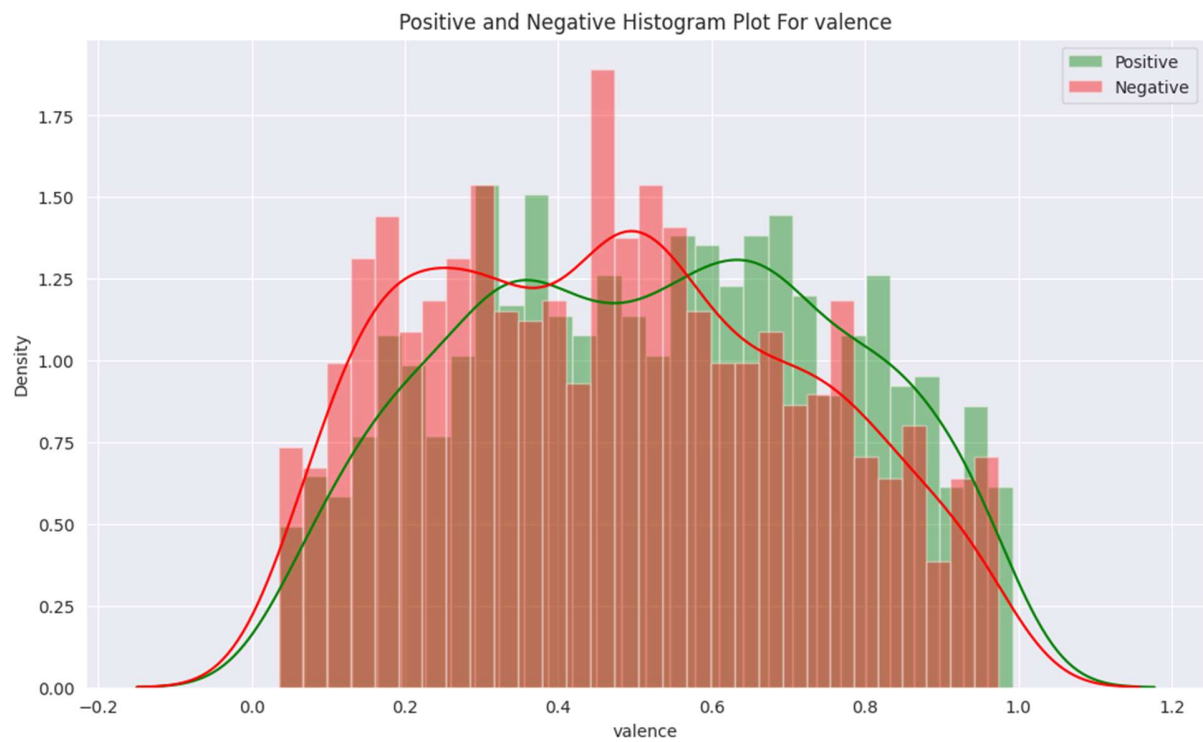
12-> We can also create **multiple feature plots** for positive and negative histogram of all the features listed above.





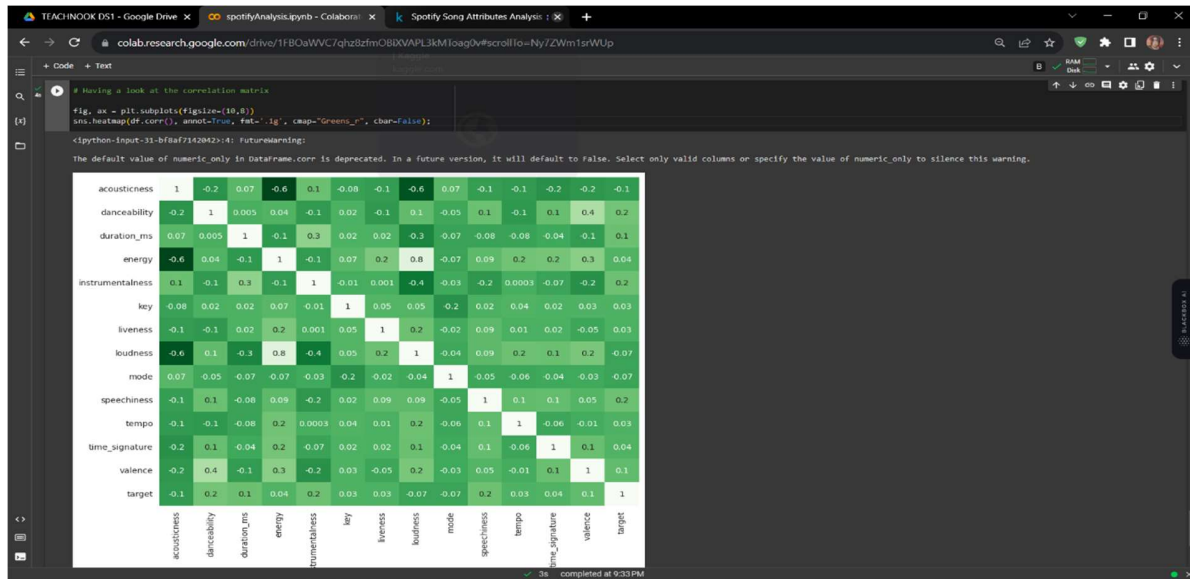






```
# Having a look at the correlation matrix

fig, ax = plt.subplots(figsize=(10,8))
sns.heatmap(df.corr(), annot=True, fmt='.1g', cmap="Greens_r",
cbar=False);
```



13->Audio Feature Correlations:

Heatmaps can show which audio features tend to co-occur or have strong correlations.

For instance, you might notice that songs with high valence (positivity) also tend to have higher energy levels.

14->Comparing Songs:

By looking at the heatmap, you can quickly identify songs with similar or dissimilar audio features. This can be helpful for music recommendation systems or for finding songs that share certain characteristics.

15->Outliers:

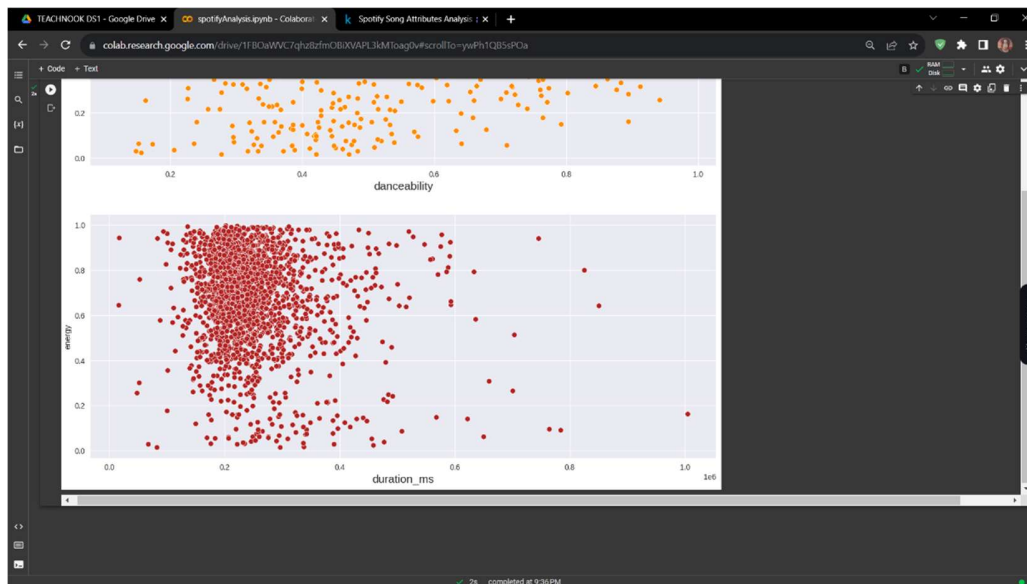
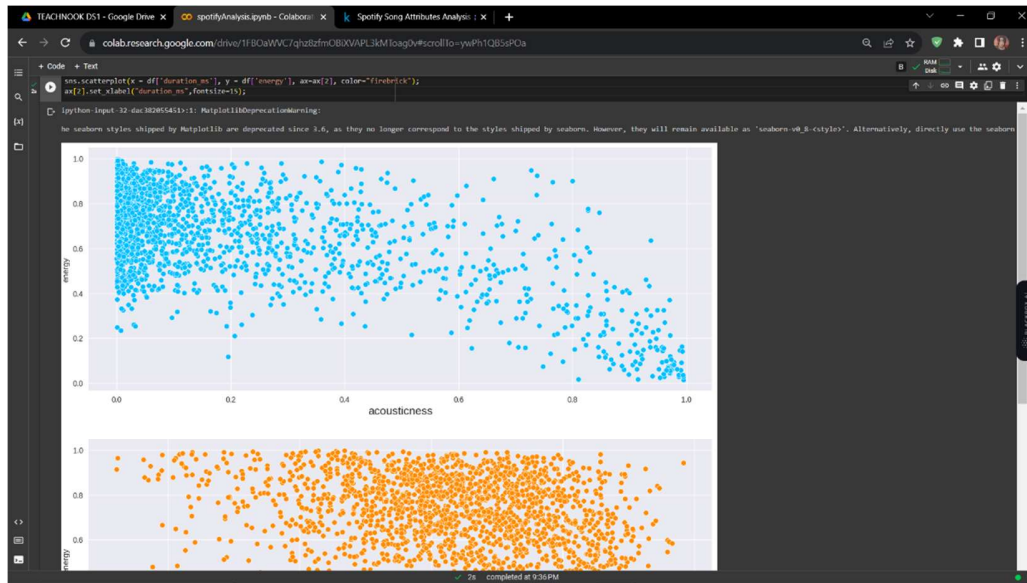
Heatmaps can highlight songs with unusual combinations of audio features, potentially helping to discover unique or standout tracks.

```
plt.style.use("seaborn")
fig, ax = plt.subplots(3,1, figsize=(15,20))

sns.scatterplot(x = df['acousticness'], y = df['energy'], ax=ax[0],
color="deepskyblue");
ax[0].set_xlabel("acousticness",fontsize=15);

sns.scatterplot(x = df['danceability'], y = df['energy'], ax=ax[1],
color="darkorange");
ax[1].set_xlabel("danceability",fontsize=15);

sns.scatterplot(x = df['duration_ms'], y = df['energy'], ax=ax[2],
color="firebrick");
ax[2].set_xlabel("duration_ms",fontsize=15);
```



Terms description ->

Acousticness

A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic. (≥ 0 , ≤ 1)

danceability

Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.

duration_ms integer The duration of the track in milliseconds.

energy

Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.

instrumentalness

Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.

key

The key the track is in. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C#/Db, 2 = D, and so on. If no key was detected, the value is -1. (≥ -1 , ≤ 11)

liveness

Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.

loudness

The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typically range between -60 and 0 db.

Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.

speechiness

Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.

tempo

The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.

time_signature

An estimated time signature. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure). The time signature ranges from 3 to 7 indicating time signatures of "3/4", to "7/4". (≥ 3 , ≤ 7)

valence

A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry). (≥ 0 , ≤ 1)

