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

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

```
df=pd.read_csv("/content/drive/MyDrive/datasets/spotify.csv")
df.columns
```

```
Index(['Unnamed: 0', 'acousticness', 'danceability', 'duration_ms', 'energy',
       'instrumentalness', 'key', 'liveness', 'loudness', 'mode',
       'speechiness', 'tempo', 'time_signature', 'valence', 'target',
       'song_title', 'artist'],
      dtype='object')
```

```
df.drop("Unnamed: 0", axis=1, inplace=True)
df.head()
```

	acousticness	danceability	duration_ms	energy	instrumentalness	key	l
0	0.0102	0.833	204600	0.434	0.021900	2	
1	0.1990	0.743	326933	0.359	0.006110	1	
2	0.0344	0.838	185707	0.412	0.000234	2	
3	0.6040	0.494	199413	0.338	0.510000	5	
4	0.1800	0.678	392893	0.561	0.512000	5	

```
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()
```

```
<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
```

```
df.shape
```

```
(2017, 16)
```

```
df.describe()
```

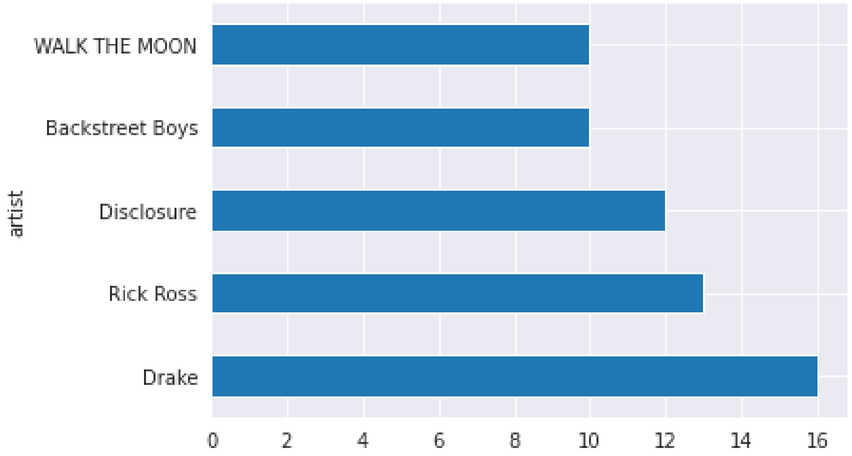
	acousticness	danceability	duration_ms	energy	instrumentalness
count	2017.000000	2017.000000	2.017000e+03	2017.000000	2017.000000
mean	0.187590	0.618422	2.463062e+05	0.681577	0.133286
std	0.259989	0.161029	8.198181e+04	0.210273	0.273162
min	0.000003	0.122000	1.604200e+04	0.014800	0.000000
25%	0.009630	0.514000	2.000150e+05	0.563000	0.000000
50%	0.063300	0.631000	2.292610e+05	0.715000	0.000076
75%	0.265000	0.738000	2.703330e+05	0.846000	0.054000
max	0.995000	0.984000	1.004627e+06	0.998000	0.976000

```
# Top 5 popular artist

top_five_artist=df.groupby("artist").count().sort_values(by="song_title", ascending=False)["song_title"][:5]
top_five_artist
```

```
artist
Drake              16
Rick Ross          13
Disclosure         12
Backstreet Boys    10
```

```
WALK THE MOON      10
Name: song_title, dtype: int64
```

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

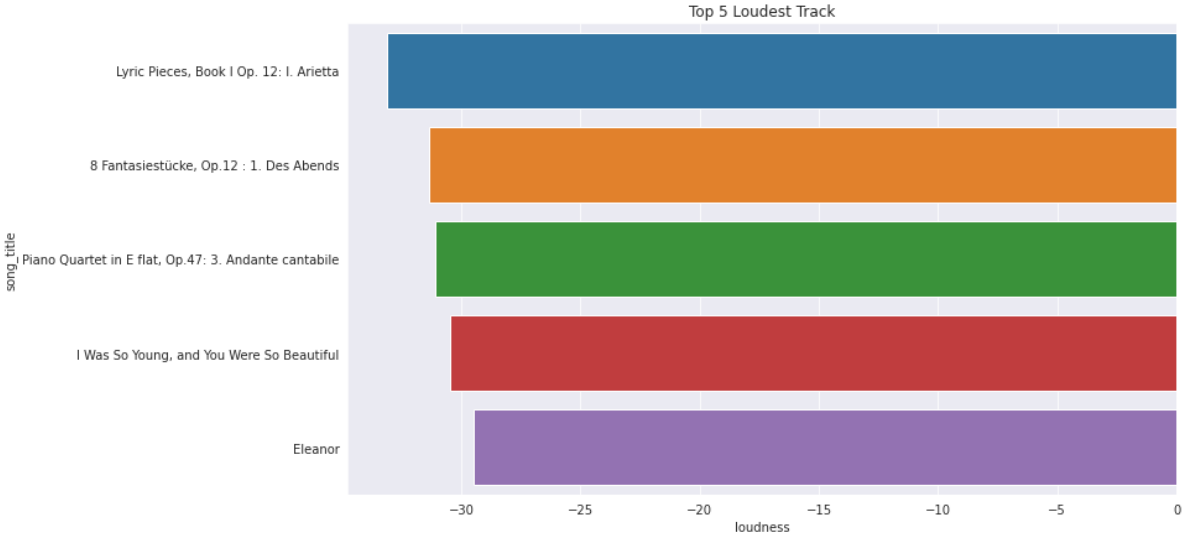


```
# Top 5 loudest track
top_loudest_track=df[["loudness","song_title"]].sort_values(by="loudness", ascending="true")[:5]
top_loudest_track
```

	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

```
plt.figure(figsize=(12,7))
sns.barplot(x="loudness",y="song_title",data=top_loudest_track)
```

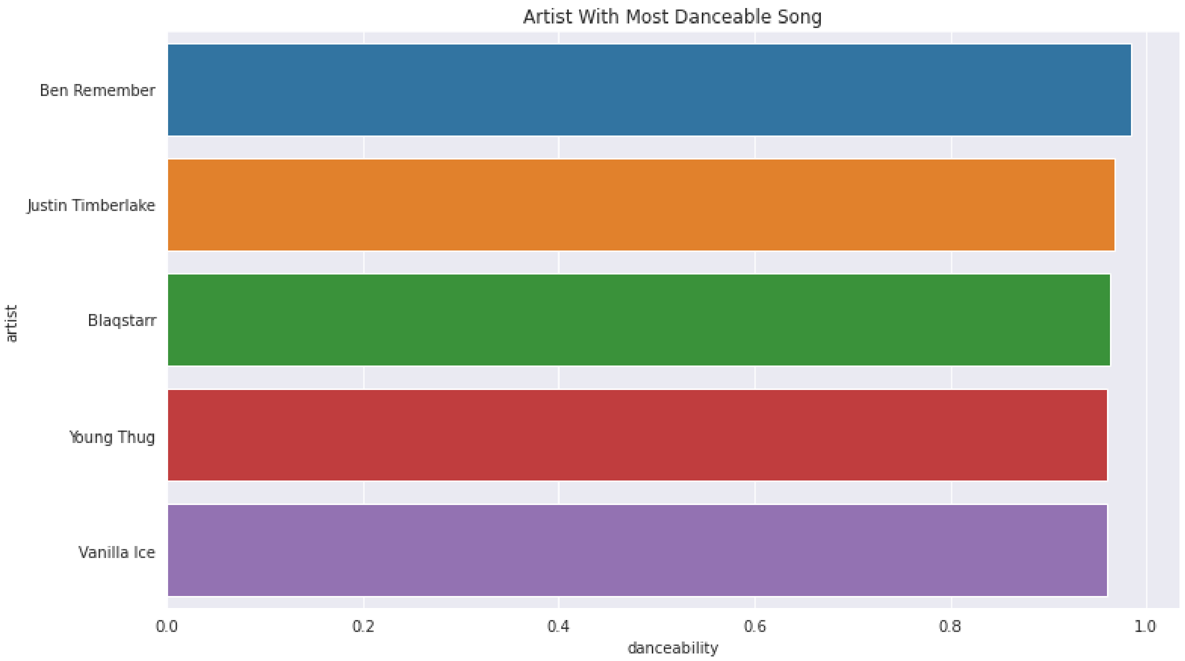
```
plt.title("Top 5 Loudest Track")
plt.show()
```



```
top_five_artist_danceable_song=df[["danceability", "song_title", "artist"]].sort_values(by="danceability", ascending=False)[:5]
top_five_artist_danceable_song
```

	danceability	song_title	artist
1433	0.984	Flashwind - Radio Edit	Ben Remember

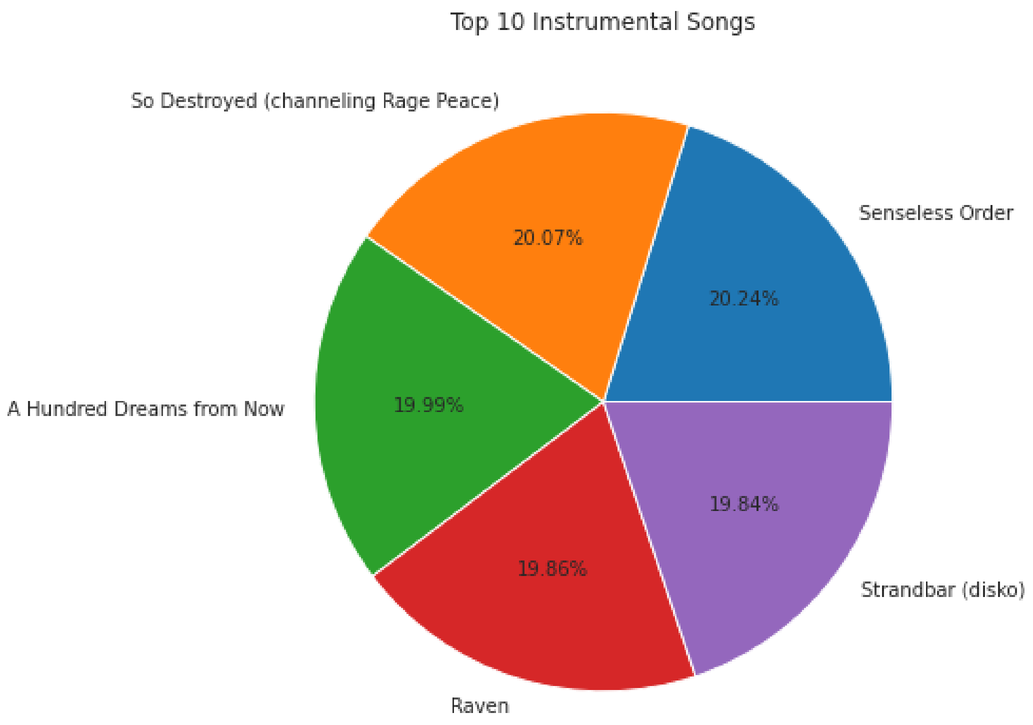
```
plt.figure(figsize=(12,7))
sns.barplot(x="danceability",y="artist", data=top_five_artist_danceable_song)
plt.title("Artist With Most Danceable Song")
plt.show()
```



```
# Top 1- instrumentalness Tracks
top_ten_instrumental_tracks=df[["instrumentalness", "song_title", "artist"]].sort_values(by="instrumentalness", ascending=False
top_ten_instrumental_tracks
```

	instrumentalness	song_title	artist
1313	0.976	Senseless Order	Signs of the Swarm
271	0.968	So Destroyed (channeling Rage Peace)	Prince Rama
1575	0.964	A Hundred Dreams from Now	Ray Bryant
1619	0.958	Raven	John Dahlbäck
725	0.957	Strandbar (disko)	Todd Terje

```
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.title("Top 10 Instrumental Songs")
plt.show()
```



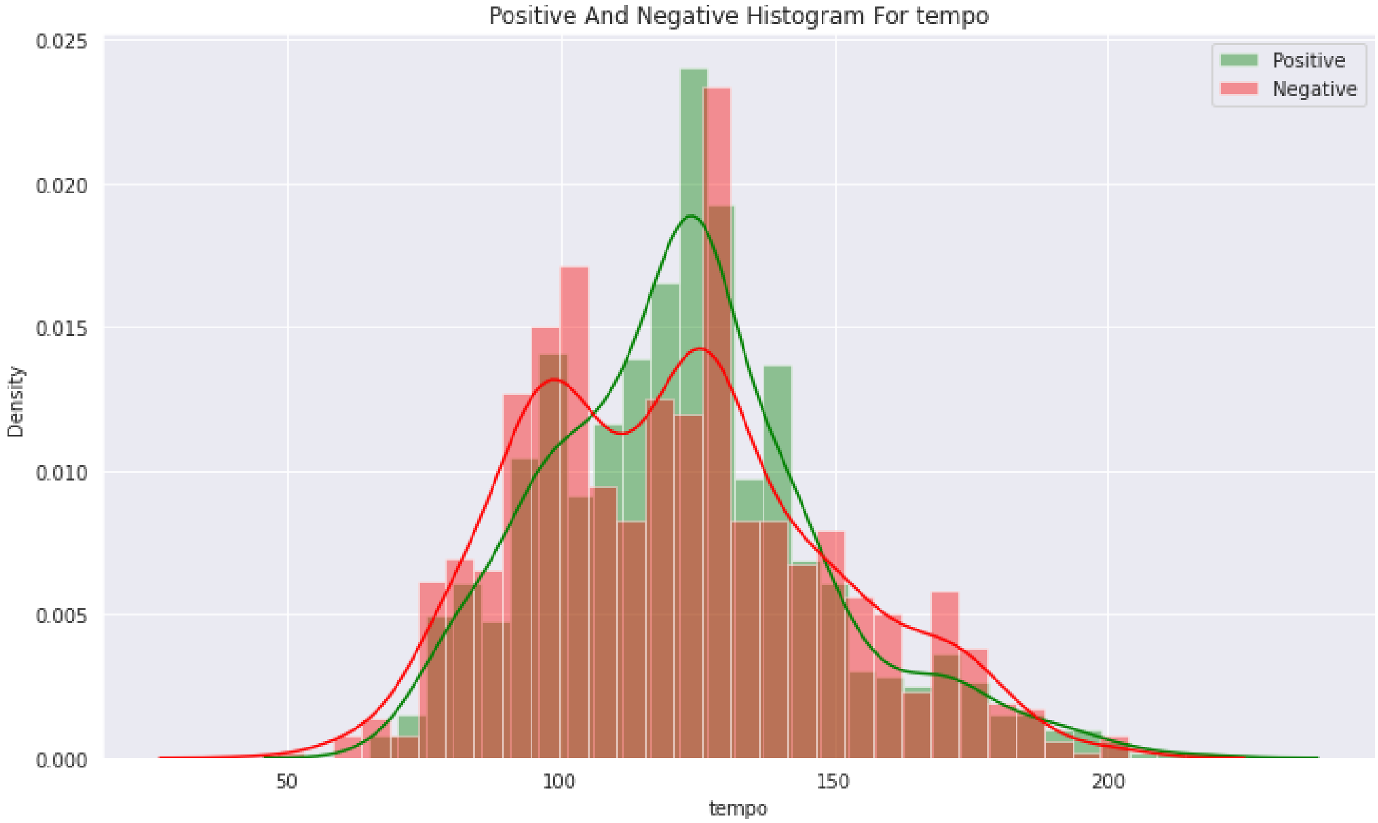
```
df.info()
```

```
<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
```

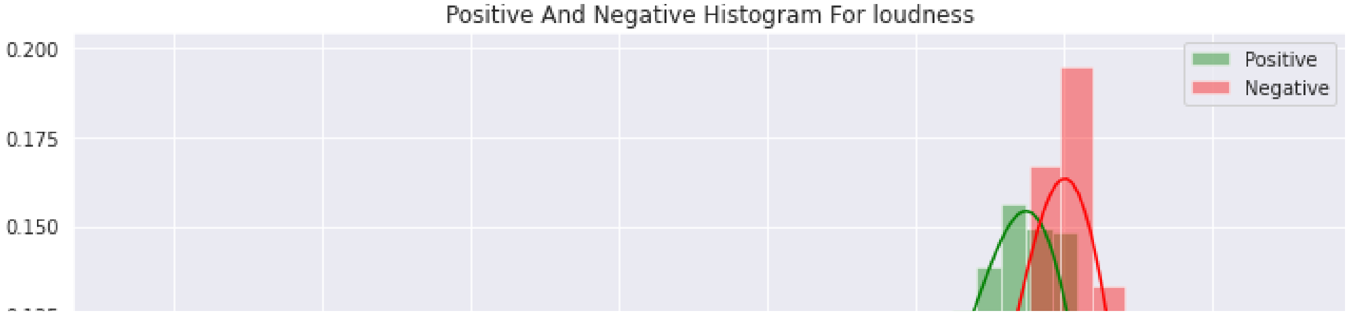
```
interest_feature_cols = ["tempo", "loudness","acousticness","danceability","duration_ms","energy","instrumentalness","liveness"
```

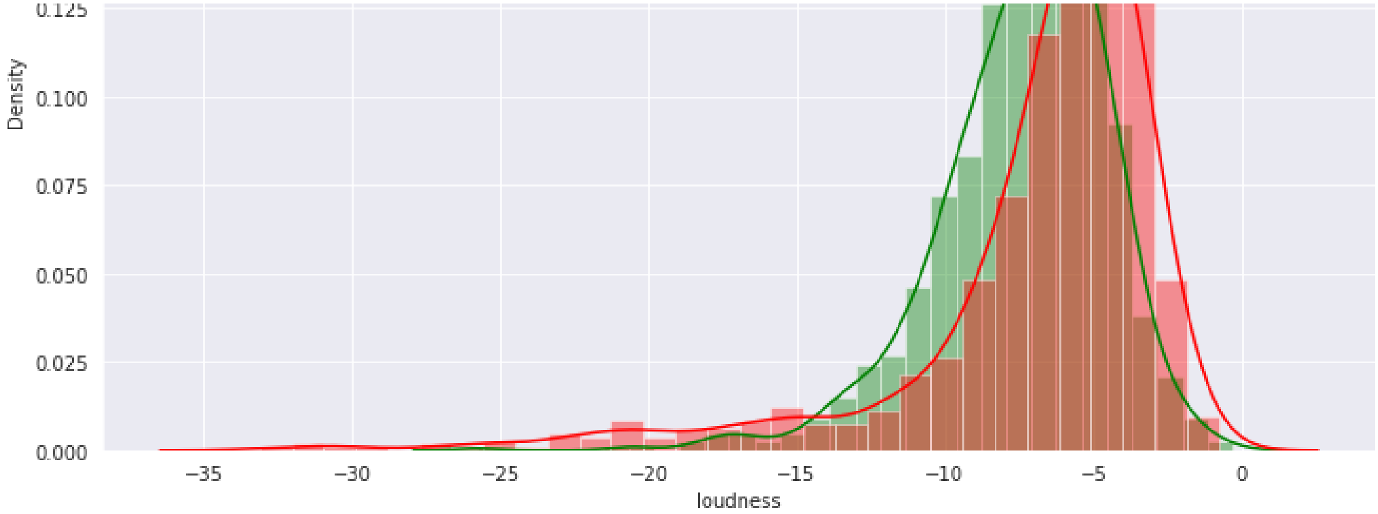
```
for feature_col in interest_feature_cols:
  pos_data = df[df["target"]== 1][feature_col]
  neg_data = df[df["target"]== 0][feature_col]

  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 For {feature_col}")
  plt.show()
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
```

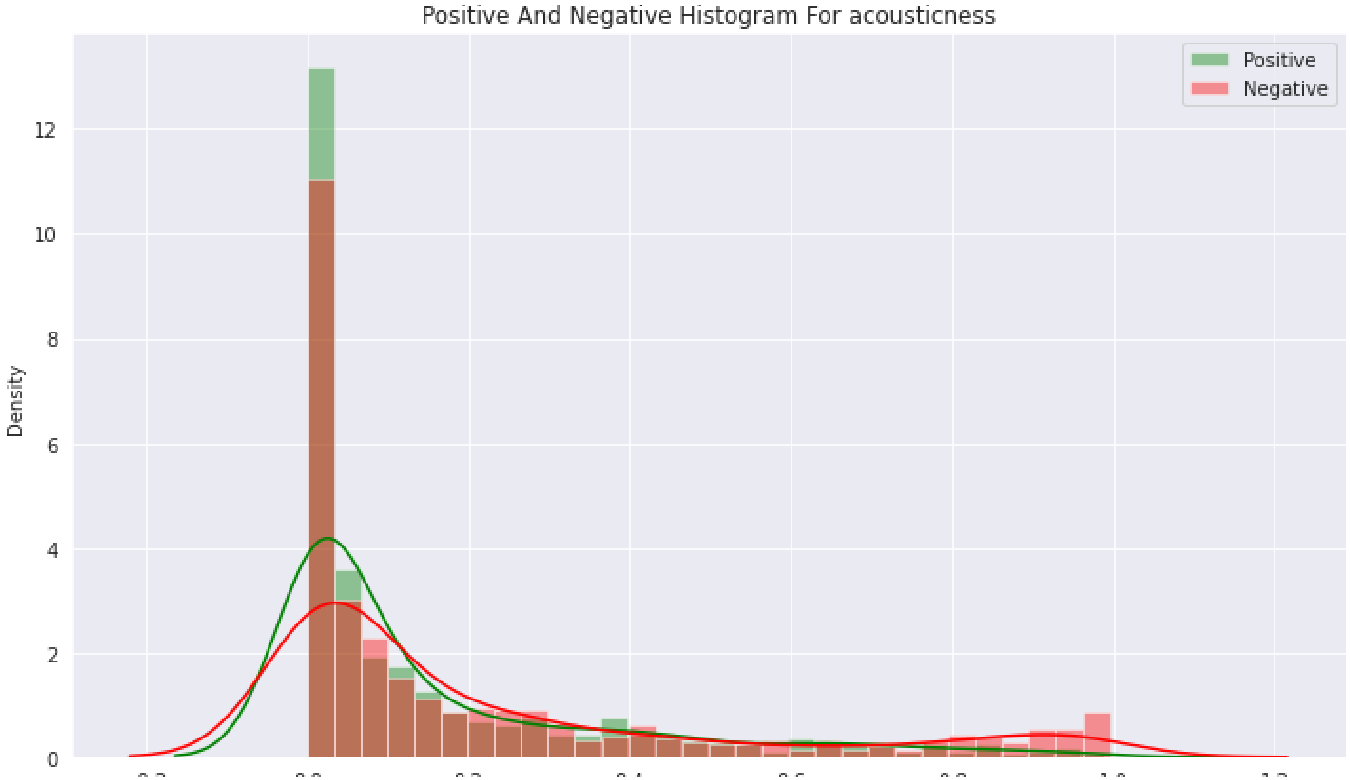


```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
```

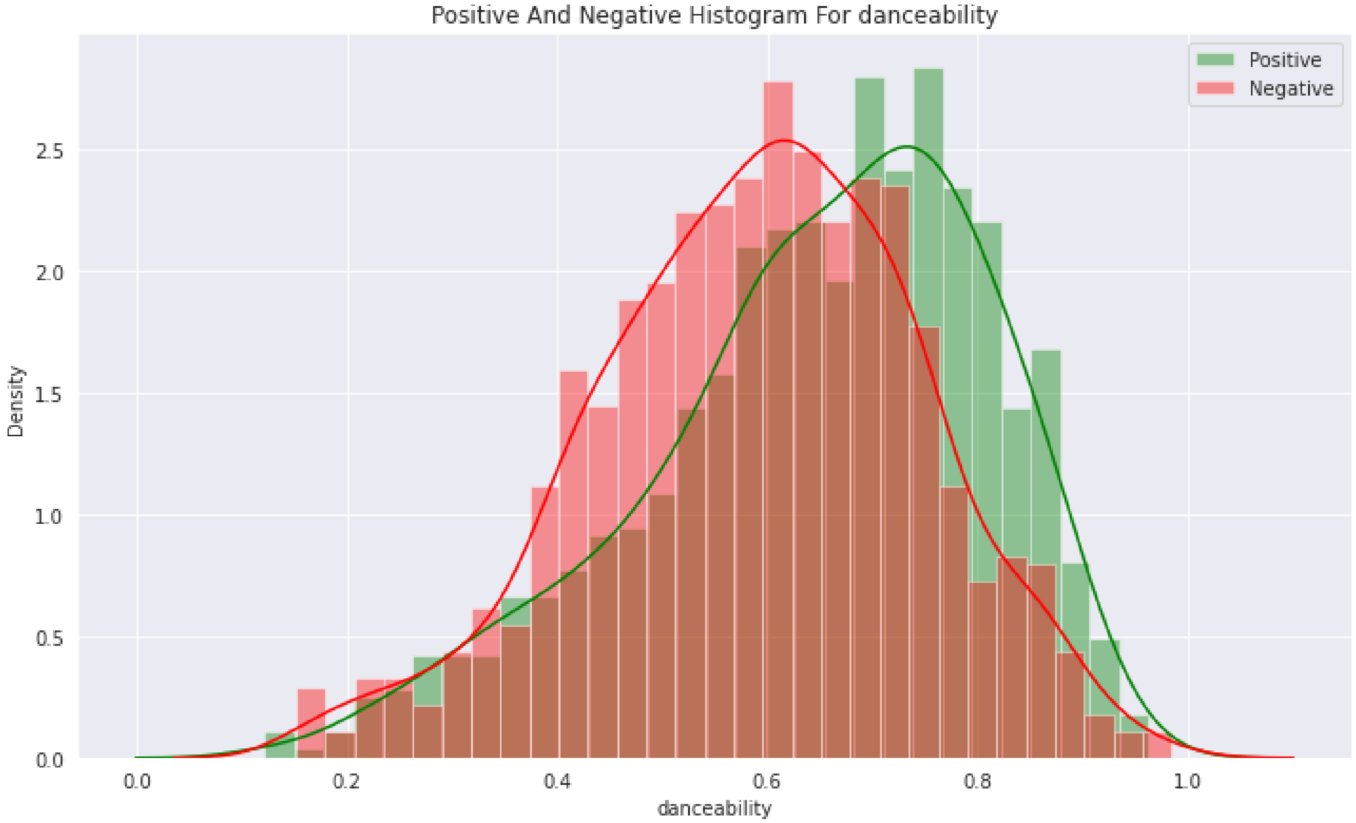




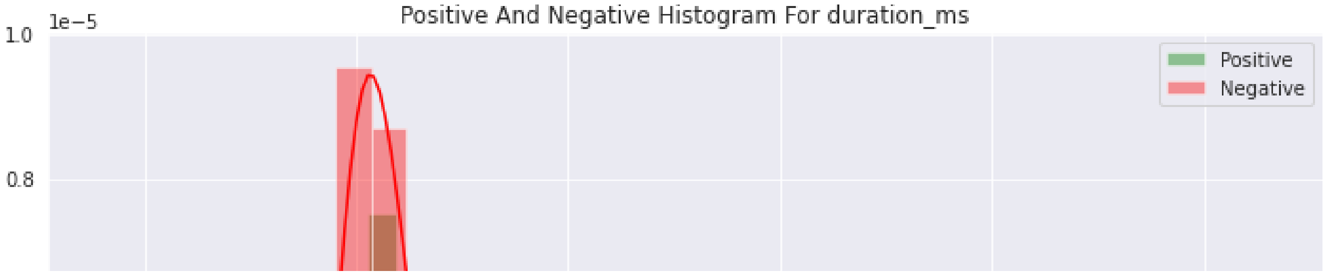
```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
```

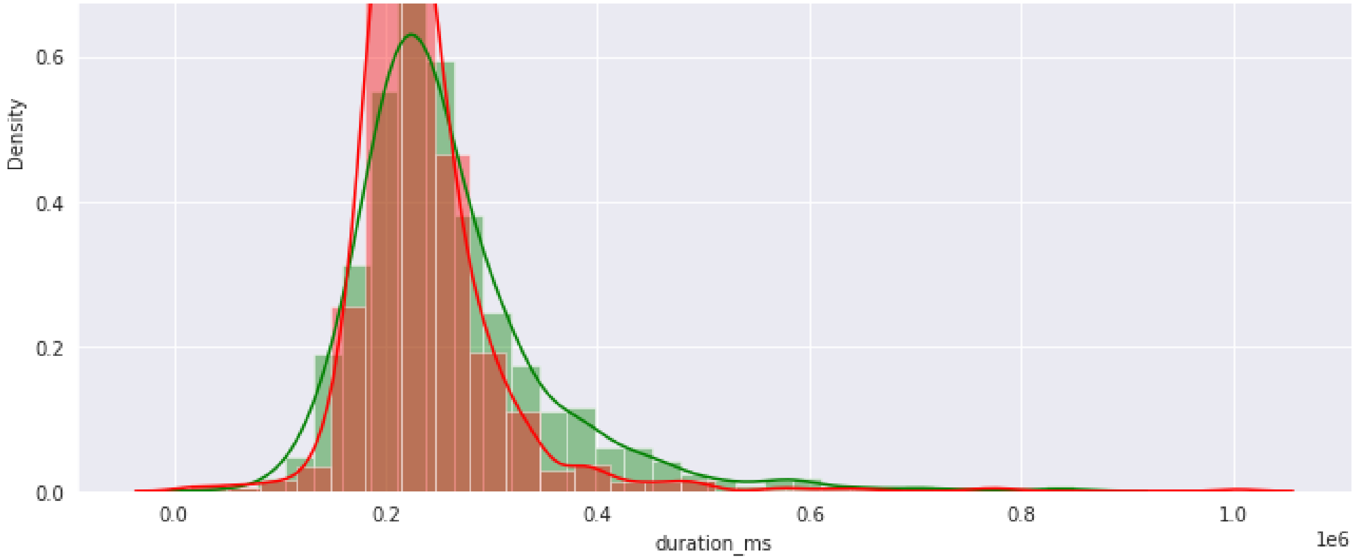


```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
```

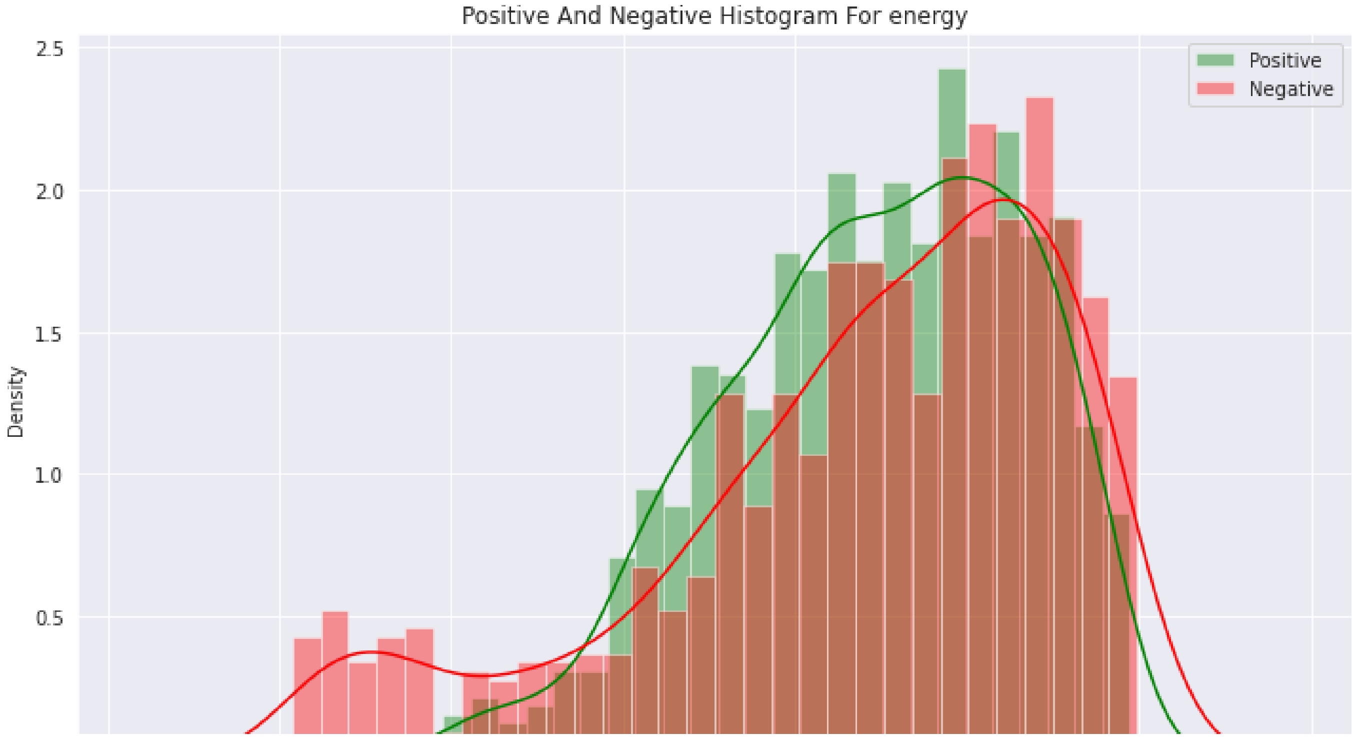


```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
```

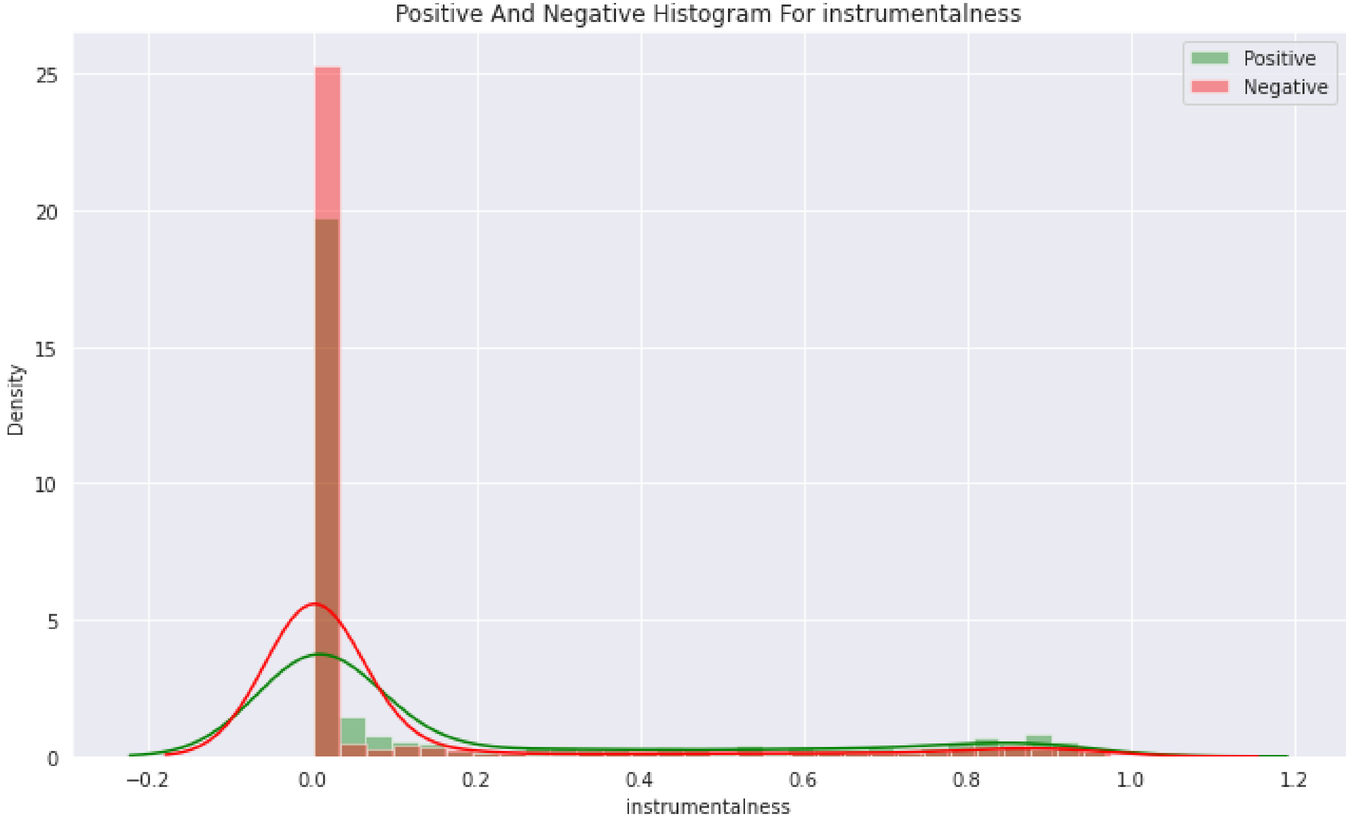




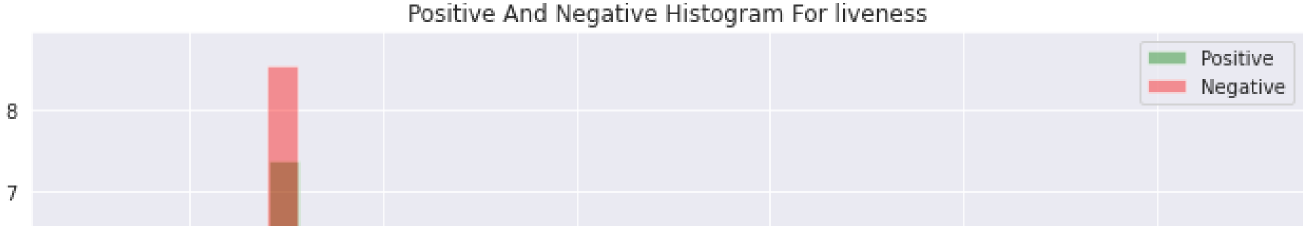
```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
```

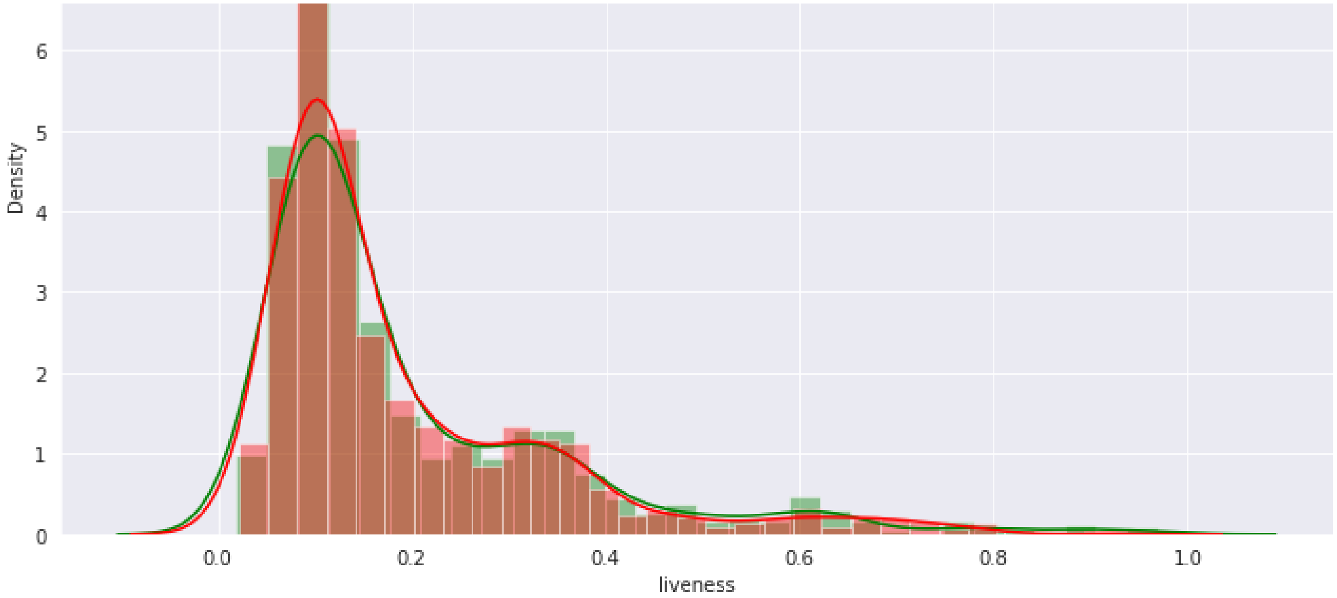


```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
```



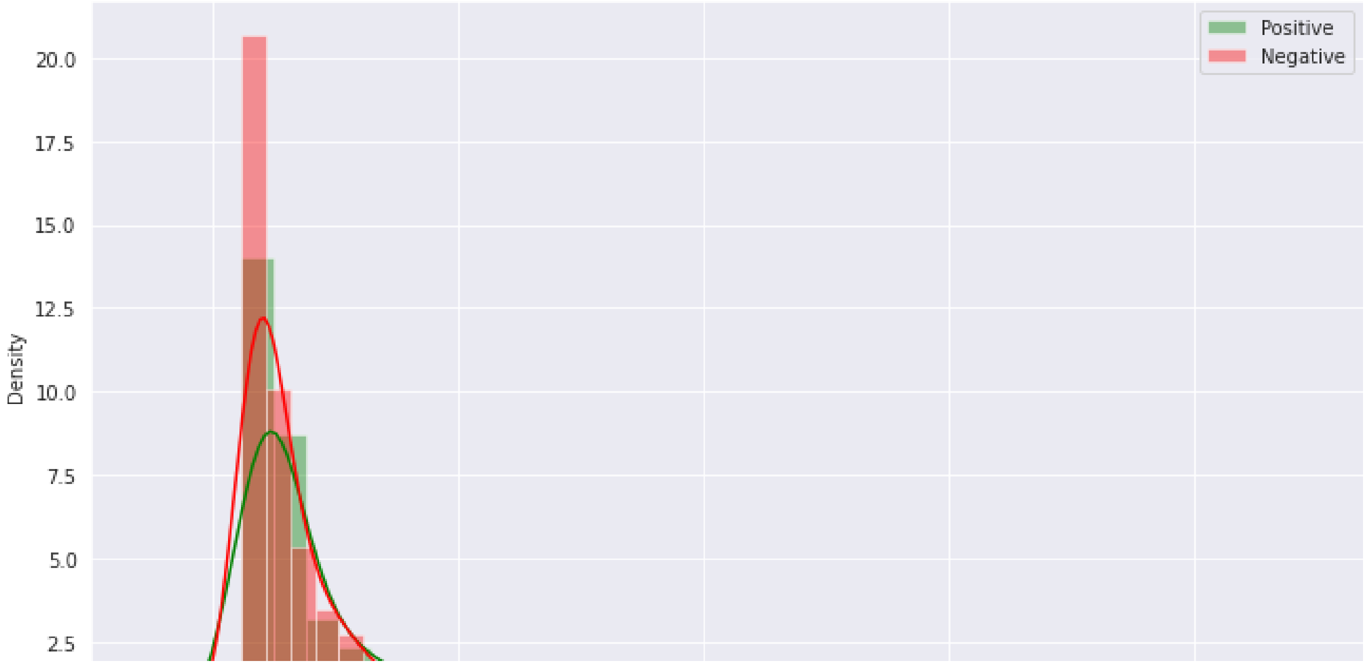
```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
```

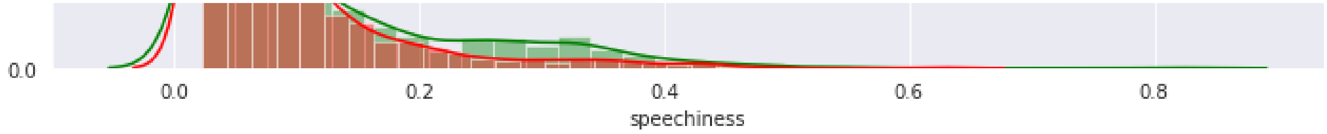




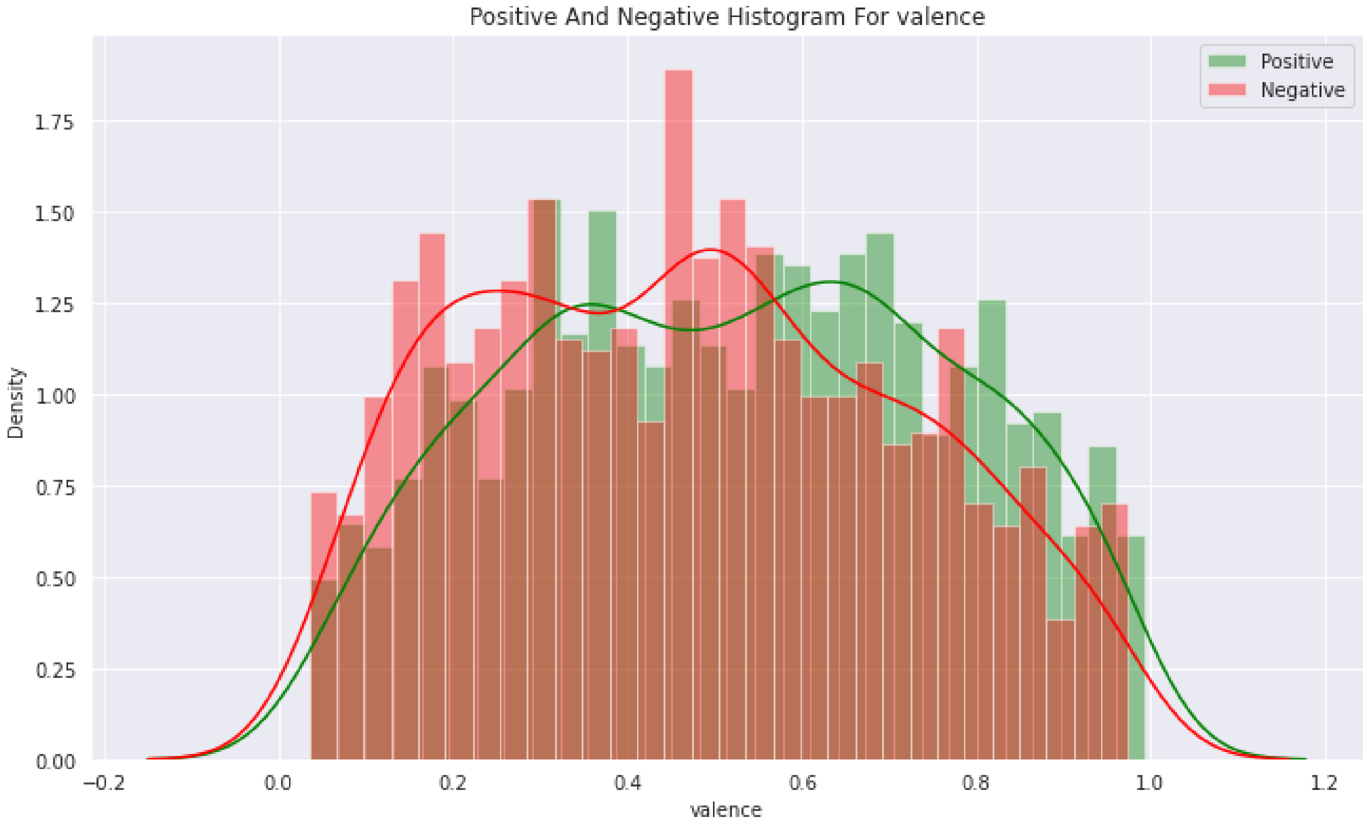
```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
```

Positive And Negative Histogram For speechiness





```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function a
  warnings.warn(msg, FutureWarning)
```



✓ 4s completed at 14:26 ● ×