Exploratory Data Analysis

Collins Tse Jacky Ke Rebecca Bachtra Christy Yau

Thursday 18th December, 2025

Curvenote

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

sns.set(style="whitegrid")
plt.rcParams["figure.figsize"] = (10, 5)

df = pd.read_csv("data/processed/spotify_clean.csv")

df.head()
```

	track_id	artists	album_name	track_name	popularity	duration_ms	explicit	danceability	energy	...	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature	track_genre	duration_min
0	5SuOikwiRyPMVoIQDJUgSV	Gen Hoshino	Comedy	Comedy	73	230666	False	0.676	0.461	...	0	0.1430	0.0322	1.01e-06	0.358	0.715	87.917	4	acoustic	3.844433
1	4qPNDBW1i3p13qLCt0Ki3A	Ben Woodward	Ghost (Acoustic)	Ghost - Acoustic	55	149610	False	0.420	0.166	...	1	0.0763	0.924	0.00101	0.101	0.267	77.489	4	acoustic	2.493500
2	1iJBSr7s7jYXzM8EGcbK5b	Ingrid Michaelson;ZAYN	To Begin Again	To Begin Again	57	210240	False	0.438	0.359	...	1	0.0557	0.210	0.000349	0.120	0.120	76.332	4	acoustic	3.504000
3	6lfxq3CG4xtTiEg7opyS2b	Kina Grannis	Crazy Rich Asians (Original Motion Picture Sou...	Can't Help Falling In Love	71	201933	False	0.266	0.0596	...	1	0.0363	0.905	0.000071	0.132	0.143	181.740	3	acoustic	3.365550
4	5vjLSffimiIP26QG5WcN2K	Chord Overstreet	Hold On	Hold On	82	198853	False	0.618	0.443	...	1	0.0526	0.469	0.000818	0.167	0.907	119.949	4	acoustic	3.314217

5 rows × 21 columns

```
print("Shape:", df.shape)
df.info()
```

```
Shape: (89741, 21)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 89741 entries, 0 to 89740
Data columns (total 21 columns):
 #   Column            Non -Null Count  Dtype
 - - -  - - - - - - -           - - - - - - - - - - - - - - - - -  - - - - - -
 0   track_id          89741 non -null  object
 1   artists           89740 non -null  object
 2   album_name        89740 non -null  object
 3   track_name        89740 non -null  object
 4   popularity        89741 non -null  int64
 5   duration_ms       89741 non -null  int64
 6   explicit          0 non -null      float64
 7   danceability      89741 non -null  float64
 8   energy            89741 non -null  float64
```

```
 9   key               89741 non -null  int64
 10  loudness          89741 non -null  float64
 11  mode              89741 non -null  int64
 12  speechiness       89741 non -null  float64
 13  acousticness      89741 non -null  float64
 14  instrumentalness  89741 non -null  float64
 15  liveness          89741 non -null  float64
 16  valence           89741 non -null  float64
 17  tempo             89741 non -null  float64
 18  time_signature    89741 non -null  int64
 19  track_genre       89741 non -null  object
 20  duration_min      89741 non -null  float64
dtypes: float64(11), int64(5), object(5)
memory usage: 14.4+ MB
```

1 Transposed Statistical Summary of Numeric Columns

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
popularity	89741.0	33.198432	20.580824	0.000	19.0000	33.000000	49.000000	1.000000e+02
duration_ms	89741.0	229141.81	122847.70	1000	173040.00	209293.00	260293.00	5.237295e+06
explicit	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
danceability	89741.0	0.562166	0.176691	0.000	0.4500	0.576000	0.692000	9.850000e-01
energy	89741.0	0.634458	0.256605	0.000	0.4570	0.676000	0.853000	1.000000e+00
key	89741.0	5.283549	3.559897	0.000	2.0000	5.000000	8.000000	1.100000e+01
loudness	89741.0	-8.499004	5.221490	-49.531	-10.3220	-7.185000	-5.108000	4.532000e+00
mode	89741.0	0.636966	0.480877	0.000	0.0000	1.000000	1.000000	1.000000e+00
speechiness	89741.0	0.087442	0.113277	0.000	0.0360	0.048900	0.085900	9.650000e-01
acousticness	89741.0	0.328289	0.338321	0.000	0.0171	0.188000	0.625000	9.960000e-01
instrumentalness	89741.0	0.173413	0.323848	0.000	0.0000	0.000058	0.097600	1.000000e+00
liveness	89741.0	0.216970	0.194884	0.000	0.0982	0.132000	0.279000	1.000000e+00
valence	89741.0	0.469477	0.262864	0.000	0.2490	0.457000	0.682000	9.950000e-01
tempo	89741.0	122.058336	30.117532	0.000	99.2640	122.013000	140.077000	2.433720e+02
time_signature	89741.0	3.897427	0.453435	0.000	4.0000	4.000000	4.000000	5.000000e+00
duration_min	89741.0	3.819030	1.882462	0.020	2.8840	3.554883	4.404883	8.728825e+01

2 Categorical Summary

```
df.describe(include="object").T
```

	count	unique	top	freq
track_id	89741	89741	2hETkH7cOfqmz3LqZDHZf5	1
artists	89740	31437	George Jones	260
album_name	89740	46589	The Complete Hank Williams	110
track_name	89740	73608	Rockin' Around The Christmas Tree	48
track_genre	89741	113	acoustic	1000

Interpretation
track_id: All unique (as expected)
artists: 31k unique; most frequent = George Jones (260 tracks)
track_name: 73k unique titles
track_genre: 113 genres; most frequent = acoustic (1000 tracks)

3 Missing values summary

```
df.isna().sum()
```

```
track_id                0
artists                 1
album_name              1
track_name              1
popularity              0
duration_ms             0
explicit            89741
danceability            0
energy                  0
key                     0
loudness                0
mode                    0
speechiness             0
acousticness            0
instrumentalness        0
liveness                0
valence                 0
tempo                   0
```

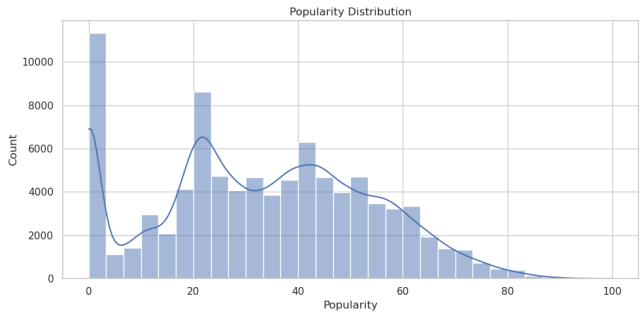
```
time_signature          0
track_genre             0
duration_min            0
dtype: int64
```

Interpretation

- artists, album_name, track_name: 1 missing value each
- explicit: All missing (excluded from final dataset)
- We can ignore this because we are only quantitative features

First, let's plot a histogram to show the frequency distribution of the prices:

```
sns.histplot(df["popularity"], bins=30, kde=True)
plt.title("Popularity Distribution")
plt.xlabel("Popularity")
plt.tight_layout()
plt.savefig('figures/popularity_distribution.png', bbox_inches='tight')
plt.show()
```



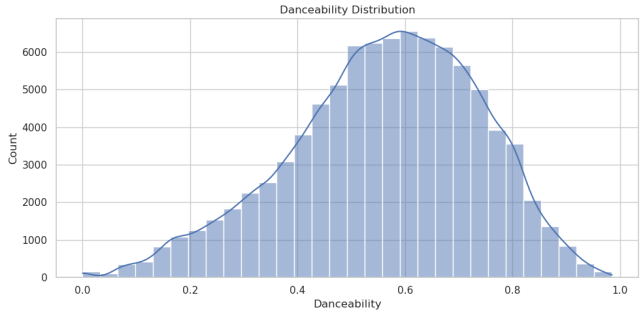
Interpretation

- Popularity is right-skewed
- Most songs have low popularity, with small peaks around 20–50
- Very few songs are extremely popular (>80)

4 More Features

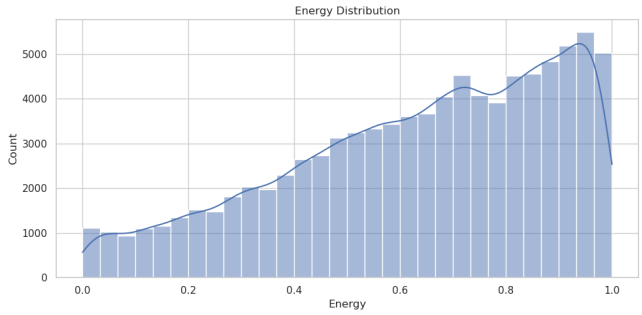
Let's look at more features we'll be using in our models.

```
sns.histplot(df["danceability"], bins=30, kde=True)
plt.title("Danceability Distribution")
plt.xlabel("Danceability")
plt.tight_layout()
plt.savefig('figures/dance_distribution.png', bbox_inches='tight')
plt.show()
```

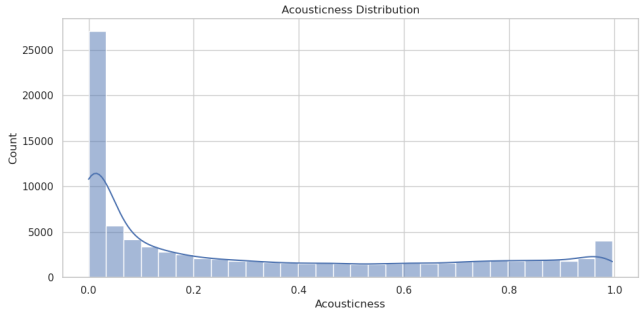


We can see danceability follows a fairly normal distribution with mean around 0.6

```
sns.histplot(df["energy"], bins=30, kde=True)
plt.title("Energy Distribution")
plt.xlabel("Energy")
plt.tight_layout()
plt.savefig('figures/energy_distribution.png', bbox_inches='tight')
plt.show()
```

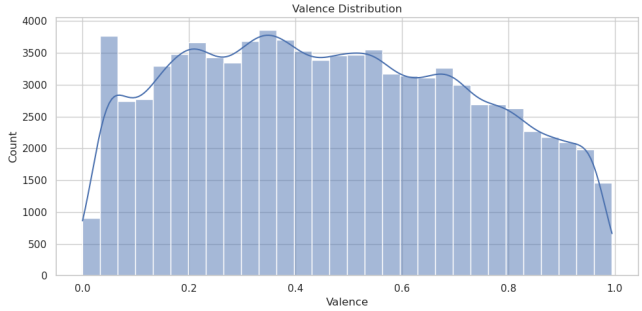



```
sns.histplot(df["acousticness"], bins=30, kde=True)
plt.title("Acousticness Distribution")
plt.xlabel("Acousticness")
plt.tight_layout()
plt.savefig('figures/acoustic_distribution.png', bbox_inches='tight')
plt.show()
```



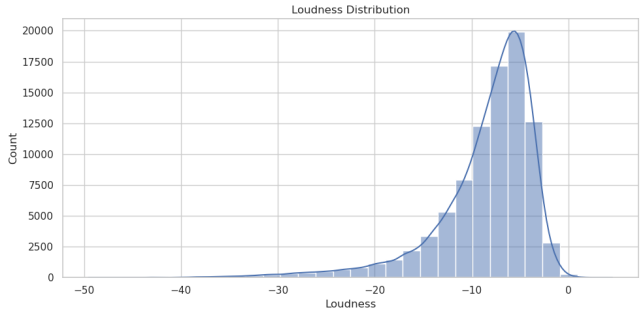
Acousticness is heavily skewed $\rightarrow$ $mostsongsareNOTacoustic$.

```
sns.histplot(df["valence"], bins=30, kde=True)
plt.title("Valence Distribution")
plt.xlabel("Valence")
plt.tight_layout()
plt.savefig('figures/valence_distribution.png', bbox_inches='tight')
plt.show()
```



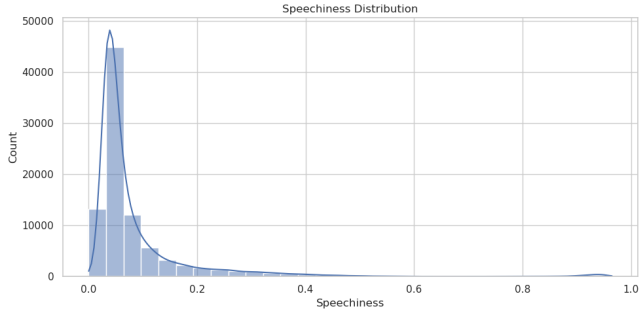
The Valence is fairly uniform — songs range from sad to happy evenly.

```
sns.histplot(df["loudness"], bins=30, kde=True)
plt.title("Loudness Distribution")
plt.xlabel("Loudness")
plt.tight_layout()
plt.savefig('figures/loudness_distribution.png', bbox_inches='tight')
plt.show()
```



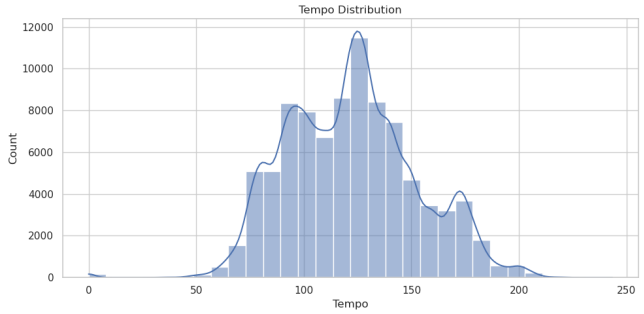
Songs volume is normally distributed around -8 dB.

```
sns.histplot(df["speechiness"], bins=30, kde=True)
plt.title("Speechiness Distribution")
plt.xlabel("Speechiness")
plt.tight_layout()
plt.savefig('figures/speech_distribution.png', bbox_inches='tight')
plt.show()
```



Most songs have mostly low speechiness→ *notspokenword*.

```
sns.histplot(df["tempo"], bins=30, kde=True)
plt.title("Tempo Distribution")
plt.xlabel("Tempo")
plt.tight_layout()
plt.savefig('figures/tempo_distribution.png', bbox_inches='tight')
plt.show()
```



The majority of tempos range from 100–140 BPM.

5 Calculate correlation of all numeric features with popularity, sorted by strength (positive to negative).

```
# Calculate correlation only for numeric columns
corr = df.select_dtypes(include=[np.number]).corr()
corr["popularity"].sort_values(ascending=False)
corr.to_csv("results/full_correlation_matrix.csv")
```

Interpretation

- No audio feature strongly predicts popularity
- Loudness has the highest positive correlation (still small)

- Instrumentalness has the highest negative correlation → $instrumentalsongstendtobelesspopularPopularityisli$