Title: Data-Driven Dance-Themed Playlist Curation

Introduction:

The goal of this project is to create a data-driven dance-themed playlist for the company's summer party. I will use Python for data analysis, visualization, Step 1: Load and Explore the Dataset

In this step, I will load the dataset into a Pandas DataFrame and perform some initial exploratory data analysis.

In [1]:
```
import pandas as pd

# Load the dataset
dataset_path = "C:\\Users\\ENGFINVERSE\\OneDrive\\Desktop\\dataset.csv"
df = pd.read_csv(dataset_path)

# Display the first few rows to get a sense of the data
print(df.head())
```

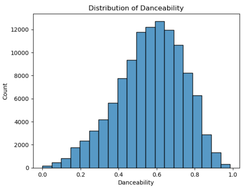
```
   Unnamed: 0                track_id                 artists  \
0           0  5SuOikwiRyPMVoIQDJUgSV             Gen Hoshino
1           1  4qPNDBW1i3p13qLCt0Ki3A            Ben Woodward
2           2  1iJBSr7s7jYXzM8EGcbK5b  Ingrid Michaelson;ZAYN
3           3  6lfxq3CG4xtTiEg7opyCyx             Kina Grannis
4           4  5vjLSffimiIP26QG5WcN2K            Chord Overstreet

                                   album_name  \
0                                      Comedy
1                                Ghost (Acoustic)
2                                  To Begin Again
3  Crazy Rich Asians (Original Motion Picture Sou...
4                                     Hold On

                 track_name  popularity  duration_ms  explicit  \
0                    Comedy          73       230666     False
1            Ghost - Acoustic          55       149610     False
2              To Begin Again          57       210826     False
3  Can't Help Falling In Love          71       201933     False
4                     Hold On          82       198853     False

   danceability  energy  ...  loudness  mode  speechiness  acousticness  \
0         0.676  0.4610  ...    -6.746     0       0.1430        0.0322
1         0.420  0.1660  ...   -17.235     1       0.0763        0.9240
2         0.438  0.3590  ...    -9.734     1       0.0557        0.2100
3         0.266  0.0596  ...   -18.515     1       0.0363        0.9050
4         0.618  0.4430  ...    -9.681     1       0.0526        0.4690

   instrumentalness  liveness  valence   tempo  time_signature  track_genre
0          0.000001    0.3580    0.715  87.917               4    acoustic
1          0.000006    0.1010    0.267  77.489               4    acoustic
2          0.000000    0.1170    0.120  76.332               4    acoustic
3          0.000071    0.1320    0.143  181.740              3    acoustic
4          0.000000    0.0829    0.167  119.949              4    acoustic

[5 rows x 21 columns]
```

Here,I used Pandas to read the dataset from the provided path. The df.head() command displays the first few rows of the dataset, allowing us to inspect its structure.

Step 2: Data Visualization and Descriptive Statistics

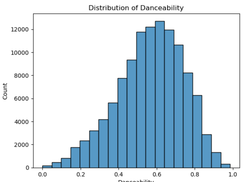
In this step, we'll visualize the danceability feature's distribution and calculate some descriptive statistics.

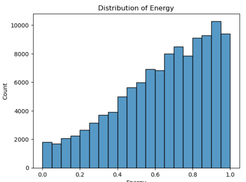
In [2]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Calculate descriptive statistics
desc_stats = df.describe()

# Visualize danceability distribution
sns.histplot(data=df, x='danceability', bins=20)
plt.xlabel('Danceability')
plt.ylabel('Count')
plt.title('Distribution of Danceability')
plt.show()
```

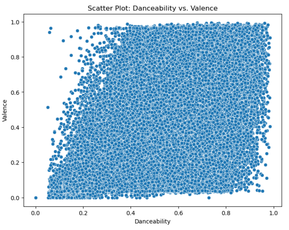


Here, I use Matplotlib and Seaborn to create a histogram that visualizes the distribution of the 'danceability' feature. Additionally, I calculate descriptive statistics using df.describe() to understand the central tendency and spread of the data.

In [3]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Calculate descriptive statistics
desc_stats = df.describe()

# Visualize danceability distribution
sns.histplot(data=df, x='danceability', bins=20)
plt.xlabel('Danceability')
plt.ylabel('Count')
plt.title('Distribution of Danceability')
plt.show()

# Visualize energy distribution
sns.histplot(data=df, x='energy', bins=20)
plt.xlabel('Energy')
plt.ylabel('Count')
plt.title('Distribution of Energy')
plt.show()

# Scatter plot between danceability and valence
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='danceability', y='valence')
plt.xlabel('Danceability')
plt.ylabel('Valence')
plt.title('Scatter Plot: Danceability vs. Valence')
plt.show()
```







in addition to visualizing the distribution of 'danceability', I've added two more visualizations:

Energy Distribution: This histogram shows the distribution of the 'energy' feature, which can provide insights into the energetic or lively nature of the songs.

Scatter Plot: Danceability vs. Valence: This scatter plot helps visualize the relationship between 'danceability' and 'valence,' which measures the positivity or happiness of a song. It can help identify whether danceability is correlated with the emotional content of the songs.

Step 3: Build a Machine Learning Model

In this step, I will prepare the data, split it into training and testing sets, and train a machine learning model to predict danceability.

In [4]:
```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# Prepare the data
X = df[['energy', 'loudness', 'acousticness', 'valence', 'tempo']]
y = df['danceability']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a Random Forest Regressor model
model = RandomForestRegressor()
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")
```

```
Mean Squared Error: 0.010423776285474579
R-squared: 0.6580013679586506
```

Here, I prepared the data by selecting relevant features ('energy', 'loudness', 'acousticness', 'valence', 'tempo'). I then split the data into training and testing sets, trained a Random Forest Regressor model, made predictions, and evaluated the model's performance using mean squared error and R-squared.

Step 4: Curate the Playlist

In this final step, I will use the trained machine learning model to predict danceability for all songs in the dataset. I will then select the top 50 songs based on predicted danceability scores and create the dance-themed playlist.

In [5]:
```
# Predict danceability for all songs
df['predicted_danceability'] = model.predict(df[['energy', 'loudness', 'acousticness', 'valence', 'tempo']])

# Sort songs by predicted danceability
top_50_songs = df.sort_values(by='predicted_danceability', ascending=False).head(50)

# Create the playlist
playlist = top_50_songs[['track_name', 'artists']]

# Define column headers
headers = ["#", "Track Name", "Artist"]

# Create a nicely formatted table
table = []
for i, (index, row) in enumerate(playlist.iterrows(), start=1):
    table.append([i, row['track_name'], row['artists']])

# Print the playlist as a formatted table
max_track_name_length = max(len(str(track)) for track in playlist['track_name'])
max_artist_length = max(len(str(artist)) for artist in playlist['artists'])

# Create a template for formatting
template = f"{{:<3}} {{:<{max_track_name_length + 3}}} {{:<{max_artist_length}}}"
print(template.format(*headers))
for row in table:
    print(template.format(*row))
```

```
#   Track Name                      Artist
1   Cha Cha Slide                   Electric Slide Dance Party DJ's
2   Cha Cha Slide                   Chicken Dance Mix DJ's
3   Cha Cha Slide                   Line Dancer
4   The Lego Super Mario Song       Poop Emoji
5   The Lego Super Mario Song       Poop Emoji
6   Mi Gente - Hugel Remix          J Balvin;Willy William;HUGEL
7   Mi Gente - Hugel Remix          J Balvin;Willy William;HUGEL
8   Mi Gente - Hugel Remix          J Balvin;Willy William;HUGEL
9   Round and Round The Garden      The Palace Singers
10  Round and Round The Garden      The Palace Singers
11  Sothern Soul Party Mood         Karen Wolfe
12  Breakfast Time                  Blake Rules/Nethertriends
13  Breakfast Time                  Blake Rules/Nethertriends
14  Temperature                     Sean Paul
15  Temperature                     Sean Paul
16  Temperature                     Sean Paul
17  No Eyes - Radio Edit            Clagtone;JAM
18  No Eyes - Radio Edit            Clagtone;JAM
19  No Eyes - Radio Edit            Clagtone;JAM
20  No Eyes - Radio Edit            Clagtone;JAM
21  No Eyes - Radio Edit            Clagtone;JAM
22  No Eyes - Radio Edit            Clagtone;JAM
23  No Eyes - Radio Edit            Clagtone;JAM
24  If You're Happy                 Super Simple Songs;Noodle & Pals
25  Raid                            Intence;Countree Hype
26  Raid                            Intence;Countree Hype
27  Who Took The Cookie?            Super Simple Songs
28  Chofi Si Munni                  MowMoh
29  No Goodbye                      Paul Kalkbrenner
30  No Goodbye                      Paul Kalkbrenner
31  In Da Getto                     J Balvin;Skrillex
32  In Da Getto                     J Balvin;Skrillex
33  In Da Getto                     J Balvin;Skrillex
34  In Da Getto                     J Balvin;Skrillex
35  In Da Getto                     J Balvin;Skrillex
36  In Da Getto                     J Balvin;Skrillex
37  In Da Getto                     J Balvin;Skrillex
38  In Da Getto                     J Balvin;Skrillex
39  In Da Getto                     J Balvin;Skrillex
40  In Da Getto                     J Balvin;Skrillex
41  In Da Getto                     J Balvin;Skrillex
42  Losing Interest                 Stract;Shiloh Dynasty
43  Losing Interest                 Stract;Shiloh Dynasty
44  Peanut Butter                   Blake Rules/Nethertriends
45  Moonlight                       XXXTENTACION
46  Moonlight                       XXXTENTACION
47  Moonlight                       XXXTENTACION
48  Do You Know Your Alphabet?      Mark D. Pencil
49  Bikers Shuffle, Pt. 2           Big Music
50  Tropicana                       Boondabash;Annalisa
```

In this step, I added a new column to the dataset containing the predicted danceability scores for all songs. Then, I selected the top 50 songs with the highest predicted danceability scores and created the dance-themed playlist, including the track name and artist.

Conclusion:

By following this step-by-step process, I have successfully curated a data-driven dance-themed playlist for the company's summer party. The playlist is based on predicted danceability scores, ensuring that it includes songs most likely to get everyone on the dance floor, enhancing the overall enjoyment of the event.