Course-End Project: Creating Cohorts of Songs

Problem Scenario: The customer always looks forward to specialized treatment, whether shopping over an e-commerce website or watching Netflix. They want what they might like to see. To keep the customers engaged, it is also crucial for companies to always present the most relevant information. Spotify is a Swedish audio streaming and media service provider. The company has over 456 million active monthly users, including over 195 million paying subscribers, as of September 2022. The company intends to create cohorts of different songs that will aid in the recommendation of songs to users based on various relevant features. Each cohort would contain similar types of songs.

https://lms.simplilearn.com
Practice Labs | Data Science with Python

jupyterlab jupyter

End Lab Reset

File Edit View Run Kernel Tabs Settings Help

cohorts of songs.ipynb

Code

Python 3 [3.10]

[2]:
```
#Import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
```

[3]:
```
file_path='rolling_stones_spotify.csv'

#Load the Excel file into DataFrame
data=pd.read_csv(file_path)

print(data.head())
```

```
   Unnamed: 0                       name               album release_date  \
0           0   Concert Intro Music - Live  Licked Live In NYC   2022-06-10
1           1   Street Fighting Man - Live  Licked Live In NYC   2022-06-10
2           2          Start Me Up - Live  Licked Live In NYC   2022-06-10
3           3  If You Can't Rock Me - Live  Licked Live In NYC   2022-06-10
4           4           Don't Stop - Live  Licked Live In NYC   2022-06-10

   track_number                      id                                uri  \
0             1  2IEkywLJ4ykbhi1yRQvmsT  spotify:track:2IEkywLJ4ykbhi1yRQvmsT
1             2  6GVgVJBKkGJoRfarYRvGTU  spotify:track:6GVgVJBKkGJoRfarYRvGTU
2             3  1Lu761pZ0dBTGpzxaQoZNW  spotify:track:1Lu761pZ0dBTGpzxaQoZNW
```

Simple 0 2 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 cohorts of songs.ipynb

https://lms.simplilearn.com
Practice Labs | Data Science with Python

jupyterlab jupyter

End Lab Reset

File Edit View Run Kernel Tabs Settings Help

cohorts of songs.ipynb

Code

Python 3 [3.10]

```
1             2  6GVgVJBKkGJoRfarYRvGTU  spotify:track:6GVgVJBKkGJoRfarYRvGTU
2             3  1Lu761pZ0dBTGpzxaQoZNW  spotify:track:1Lu761pZ0dBTGpzxaQoZNW
3             4  1agTQzOTUnGNggyckEqiDH  spotify:track:1agTQzOTUnGNggyckEqiDH
4             5  7piGJR8YndQBQWVXv6KtQw  spotify:track:7piGJR8YndQBQWVXv6KtQw

   acousticness  danceability  energy  instrumentalness  liveness  loudness  \
0        0.0824         0.463   0.993          0.996000    0.9320   -12.913
1        0.4370         0.326   0.965          0.233000    0.0961    -4.803
2        0.4160         0.386   0.969          0.400000    0.0956    -4.936
3        0.5670         0.369   0.985          0.000107    0.0895    -5.535
4        0.4000         0.303   0.969          0.055900    0.0966    -5.098

   speechiness    tempo  valence  popularity  duration_ms
0       0.1100  118.001   0.0302          33        48640
1       0.0759  131.455   0.3180          34       253173
2       0.1150  130.066   0.3130          34       263160
3       0.1930  132.994   0.1470          32       305880
4       0.0930  130.533   0.2060          32       305106
```

[4]:
```
#Intial data inspection
print("Data Info:")
print(data.info())

#check for duplicates
data=data.drop_duplicates()

#check for missing values
print("\nMissing Values:")
```

Simple 0 2 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 cohorts of songs.ipynb

https://lms.simplilearn.com
Practice Labs | Data Science with Python

jupyterlab jupyter

End Lab Reset

File Edit View Run Kernel Tabs Settings Help

cohorts of songs.ipynb

Code

Python 3 [3.10]

```
[4]: #Intial data inspection
print("Data Info:")
print(data.info())

#check for duplicates
data=data.drop_duplicates()

#check for missing values
print("\nMissing Values:")
print(data.isnull().sum())

#handle missing values
data=data.dropna()
```

```
Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1610 entries, 0 to 1609
Data columns (total 18 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Unnamed: 0        1610 non-null   int64
 1   name              1610 non-null   object
 2   album             1610 non-null   object
 3   release_date      1610 non-null   object
 4   track_number      1610 non-null   int64
 5   id                1610 non-null   object
 6   uri               1610 non-null   object
 7   acousticness      1610 non-null   float64
```

Simple 0 2 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 cohorts of songs.ipynb

https://lms.simplilearn.com
Practice Labs | Data Science with Python

jupyterlab jupyter

End Lab Reset

File Edit View Run Kernel Tabs Settings Help

cohorts of songs.ipynb

Code

Python 3 [3.10]

```
 3   release_date      1610 non-null   object
 4   track_number      1610 non-null   int64
 5   id                1610 non-null   object
 6   uri               1610 non-null   object
 7   acousticness      1610 non-null   float64
 8   danceability      1610 non-null   float64
 9   energy            1610 non-null   float64
 10  instrumentalness  1610 non-null   float64
 11  liveness          1610 non-null   float64
 12  loudness          1610 non-null   float64
 13  speechiness       1610 non-null   float64
 14  tempo             1610 non-null   float64
 15  valence           1610 non-null   float64
 16  popularity        1610 non-null   int64
 17  duration_ms       1610 non-null   int64
dtypes: float64(9), int64(4), object(5)
memory usage: 226.5+ KB
None

Missing Values:
Unnamed: 0          0
name                0
album               0
release_date        0
track_number        0
id                  0
uri                 0
acousticness        0
```

Simple 0 2 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 cohorts of songs.ipynb

File Edit View Run Kernel Tabs Settings Help

cohorts of songs.ipynb

Code Python 3 [3.10]

```
Unnamed: 0          0
name                0
album               0
release_date        0
track_number        0
id                  0
uri                 0
acousticness        0
danceability        0
energy              0
instrumentalness    0
liveness            0
loudness            0
speechiness         0
tempo               0
valence             0
popularity          0
duration_ms         0
dtype: int64
```

[5]:
```
#Recommendation based on popular songs
avg_popularity_per_album=data.groupby('album')['popularity'].mean()
top_albums=avg_popularity_per_album.nlargest(2)
print("\nTop 2 Albums Based on Popular Songs:")
print(top_albums)
```

```
Top 2 Albums Based on Popular Songs:
album
```

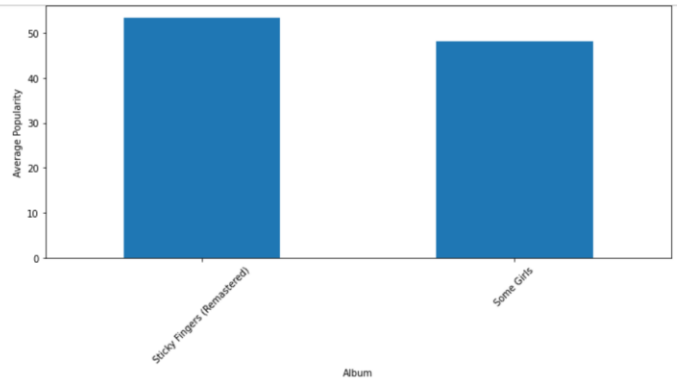
Simple 0 2 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 cohorts of songs.ipynb

https://lms.simplilearn.com
Practice Labs | Data Science with Python
jupyterlab jupyter
End Lab Reset

File Edit View Run Kernel Tabs Settings Help

cohorts of songs.ipynb

Code Python 3 [3.10]



[8]:
```
#EDA
#distribution of danceability
plt.figure(figsize=(8, 6))
```

Simple 0 2 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 cohorts of songs.ipynb

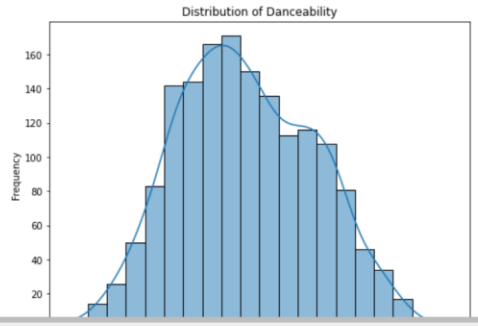
File Edit View Run Kernel Tabs Settings Help

cohorts of songs.ipynb

Code Python 3 [3.10]

```
plt.figure(figsize=(8, 6))
sns.histplot(data['danceability'], bins=20, kde=True)
plt.xlabel('Danceability')
plt.ylabel('Frequency')
plt.title('Distribution of Danceability')
plt.show
```

[8]: <function matplotlib.pyplot.show(close=None, block=None)>



Simple 0 2 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 cohorts of songs.ipynb

https://lms.simplilearn.com
Practice Labs | Data Science with Python
jupyterlab jupyter
End Lab Reset

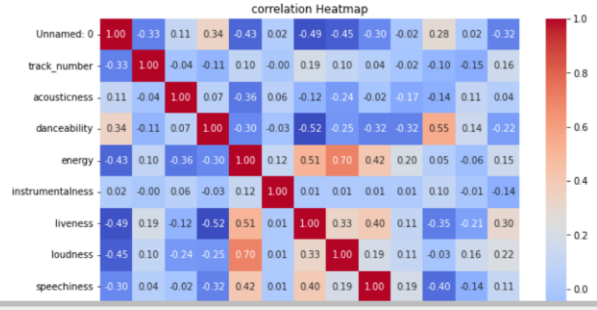
File Edit View Run Kernel Tabs Settings Help

cohorts of songs.ipynb

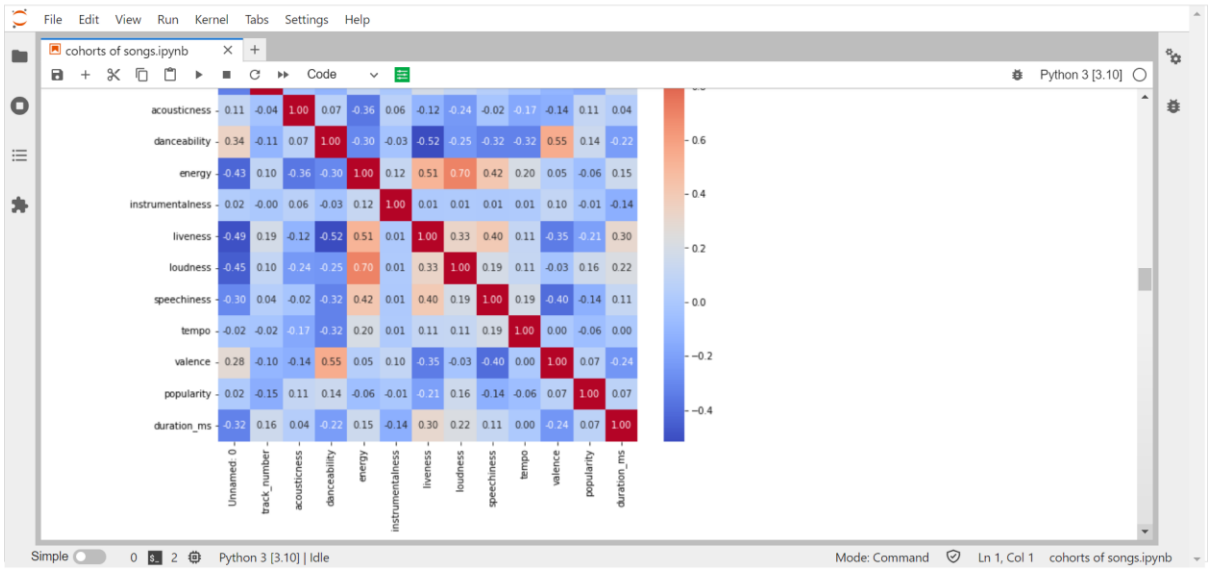
Code Python 3 [3.10]

```
plt.figure(figsize=(10, 8))
correlation_matrix=data.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('correlation Heatmap')
plt.show()
```

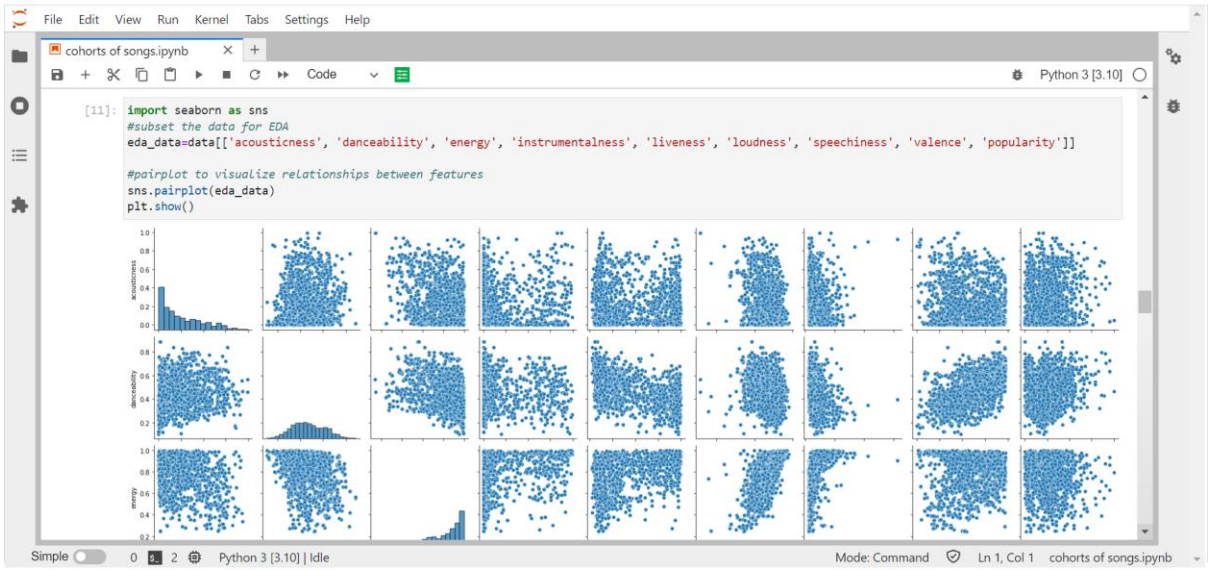
```
/tmp/ipykernel_188/4021597552.py:3: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will d
efault to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  correlation_matrix=data.corr()
```



Simple 0 2 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 cohorts of songs.ipynb



https://lms.simplilearn.com
Practice Labs | Data Science with Python
jupyterlab jupyter
End Lab Reset



https://lms.simplilearn.com
Practice Labs | Data Science with Python

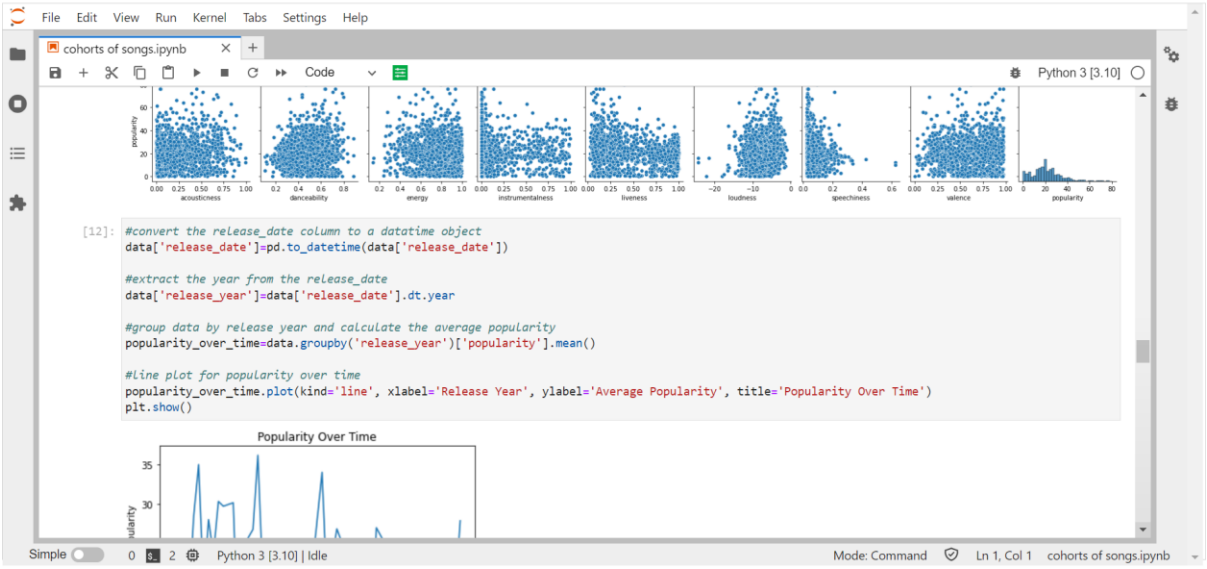
jupyterlab jupyter

End Lab Reset

https://lms.simplilearn.com
Practice Labs | Data Science with Python

jupyterlab jupyter

End Lab Reset



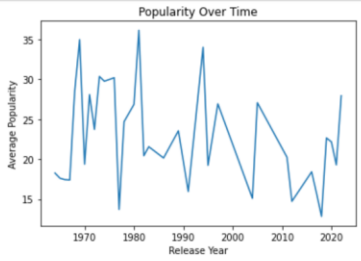
https://lms.simplilearn.com
Practice Labs | Data Science with Python

jupyterlab jupyter End Lab Reset

File Edit View Run Kernel Tabs Settings Help

cohorts of songs.ipynb

Code Python 3 [3.10]



[17]:
```
#cluster Analysis
selected_features=data[['acousticness', 'danceability']]

#standardize the features
scaler=StandardScaler()
X=scaler.fit_transform(selected_features)

k_values=range(1, 10)

wcss_values=[]
```

Simple 0 2 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 cohorts of songs.ipynb

https://lms.simplilearn.com
Practice Labs | Data Science with Python

jupyterlab jupyter End Lab Reset

File Edit View Run Kernel Tabs Settings Help

cohorts of songs.ipynb

Code Python 3 [3.10]

```
selected_features=data[['acousticness', 'danceability']]

#standardize the features
scaler=StandardScaler()
X=scaler.fit_transform(selected_features)

k_values=range(1, 10)

wcss_values=[]

max_iters= 100

for k in k_values:
    np.random.seed(0)
    centroids=X[np.random.choice(X.shape[0], k, replace=False)]

    for iteration in range(max_iters):
        distances=np.sqrt(np.sum((X[:, np.newaxis] - centroids) ** 2, axis=2))
        labels=np.argmin(distances, axis=1)
        new_centroids=np.array([X[labels == i].mean(axis=0) for i in range(k)])
        if np.all(centroids == new_centroids):
            break
        centroids=new_centroids

    #calculate WCSS
    wcss=np.sum(np.min(distances, axis=1))
    wcss_values.append(wcss)

#plot the WCSS values for each k
```

Simple 0 2 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 cohorts of songs.ipynb

File Edit View Run Kernel Tabs Settings Help

cohorts of songs.ipynb

Code Python 3 [3.10]

```
    wcss_values.append(wcss)

#plot the WCSS values for each k
plt.figure(figsize=(8, 6))
plt.plot(k_values, wcss_values, marker='o', linestyle='-', color='b')
plt.title('Elbow Method of Optimal k')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Within-Cluster Sum of Squares (WCSS)')
plt.grid(True)
plt.show()

#find the optimal value of k (elbow method)
optimal_k=np.argmin(np.diff(wcss_values))+2

k=optimal_k

np.random.seed(0)
centroids=X[np.random.choice(X.shape[0], k, replace=False)]

for iteration in range(max_iters):
    distances=np.sqrt(np.sum((X[:, np.newaxis] - centroids) ** 2, axis=2))
    labels=np.argmin(distances, axis=1)
    new_centroids=np.array([X[labels == i].mean(axis=0) for i in range(k)])
    if np.all(centroids == new_centroids):
        break
    centroids=new_centroids

#count and print the number of data points in each cluster
```

Simple 0 2 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 cohorts of songs.ipynb

https://lms.simplilearn.com
Practice Labs | Data Science with Python
jupyterlab jupyter
End Lab Reset

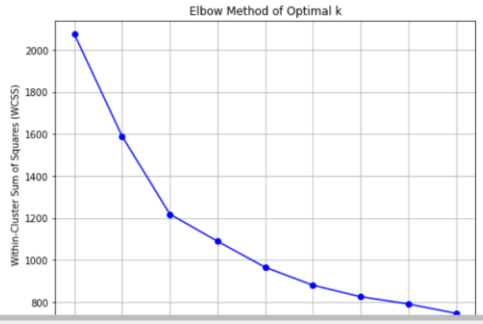
File Edit View Run Kernel Tabs Settings Help

cohorts of songs.ipynb

Code Python 3 [3.10]

```
#count and print the number of data points in each cluster
cluster_counts=np.bincount(labels)

for i in range(k):
    print(f"Cluster {i+1}")
    print(f"Centroid Used: {centroids[i]}, No. of Records: {cluster_counts[i]}")
```



Simple 0 2 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 cohorts of songs.ipynb

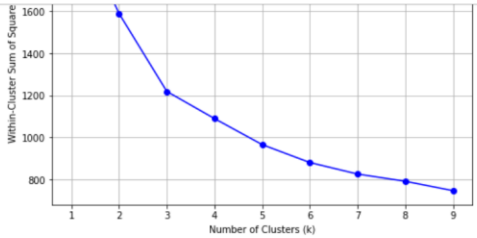
https://lms.simplilearn.com
Practice Labs | Data Science with Python

jupyterlab jupyter End Lab Reset

File Edit View Run Kernel Tabs Settings Help

cohorts of songs.ipynb

Code Python 3 [3.10]




```
Cluster 1
Centroid Used: [-0.60931236 -0.09898065], No. of Records: 1071
Cluster 2
Centroid Used: [1.21071157 0.19667584], No. of Records: 539
```

[21]:
```
#Add cluster labels to the original dataset
data['cluster']=labels

#visualize the clusters
plt.figure(figsize=(8, 6))
sns.scatterplot(x='acousticness', y='danceability', hue='cluster', data=data, palette='Set1')
plt.xlabel('Acousticness')
plt.ylabel('Danceability')
```

Simple 0 2 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 cohorts of songs.ipynb

https://lms.simplilearn.com
Practice Labs | Data Science with Python

jupyterlab jupyter End Lab Reset

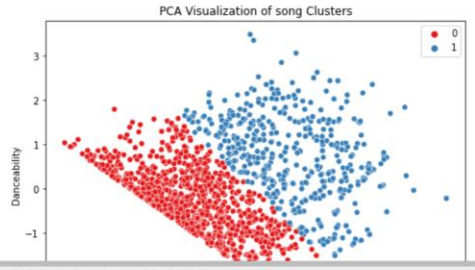
File Edit View Run Kernel Tabs Settings Help

cohorts of songs.ipynb

Code Python 3 [3.10]

[23]:
```
#Dimentionality Reduction (PCA)
pca=PCA(n_components=2)
reduced_features=pca.fit_transform(X)

#visualize the data after dimensionality reduction
plt.figure(figsize=(8, 6))
sns.scatterplot(x=reduced_features[:, 0], y=reduced_features[:, 1], hue=labels, palette='Set1')
plt.xlabel('Acousticness')
plt.ylabel('Danceability')
plt.title('PCA Visualization of song Clusters')
plt.show()
```



Simple 0 2 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 cohorts of songs.ipynb