spotify-genre-recommendation

February 7, 2024

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.metrics import pairwise_distances_argmin_min
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report,␣
↪confusion_matrix
```

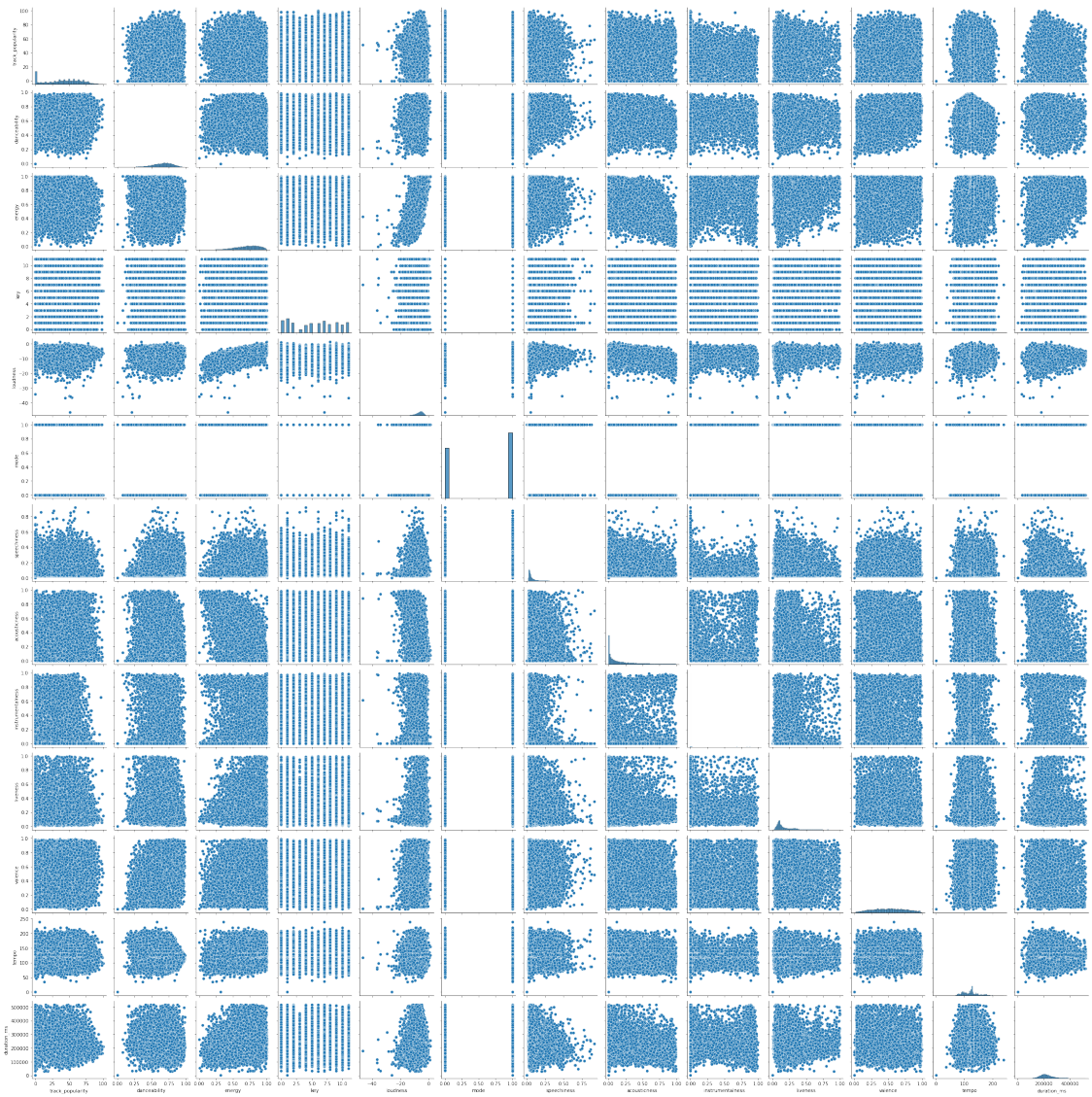
[2]:
```
spotify_data = pd.read_csv(r'/Users/akurisivanagendrareddy/Downloads/spotify␣
↪dataset.csv')
spotify_data.head(5)
```

[2]:

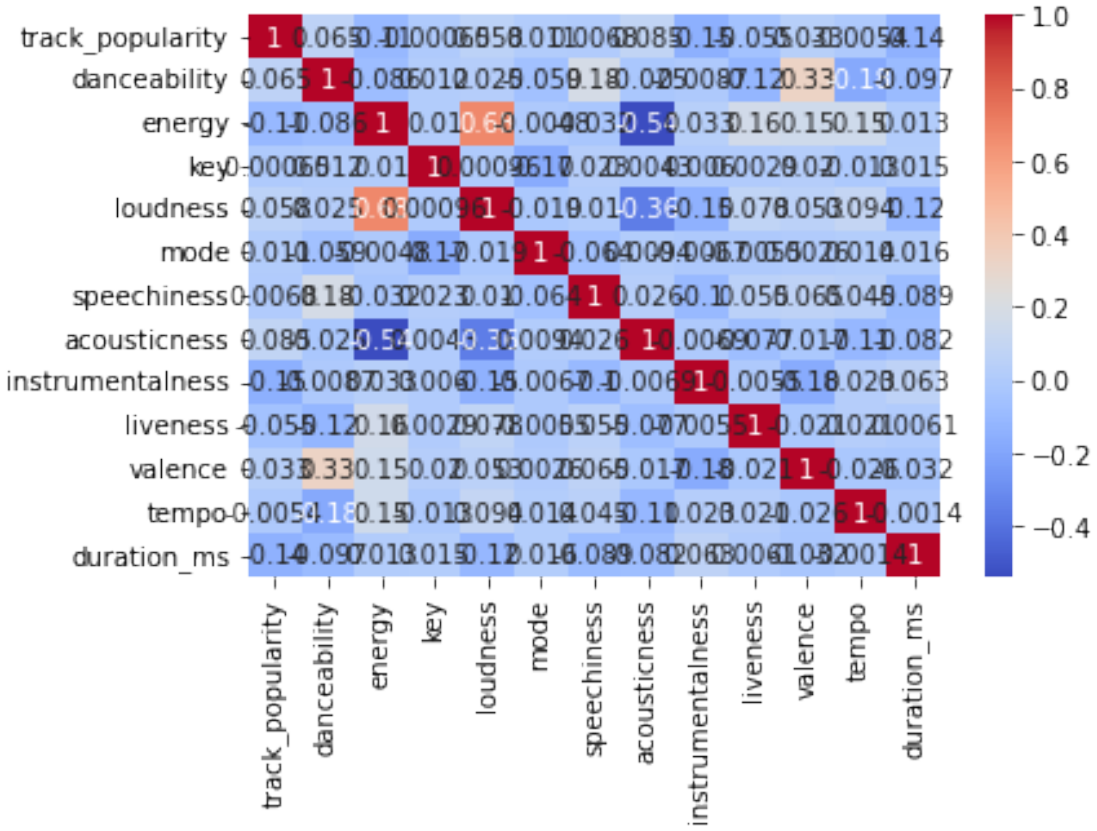
	track_id	track_name	track_artist	track_popularity	track_album_id	track_album_name	track_album_release_date
0	6f807x0ima9a1j3VPbc7VN	I Don't Care (with Justin Bieber) - Loud Luxur…	Ed Sheeran	66	2oCs0DGTsRO98Gh5ZSl2Cx	I Don't Care (with Justin Bieber) [Loud Luxury…	2019-06-14
1	0r7CVbZTWZgbTCYdfa2P31	Memories - Dillon Francis Remix	Maroon 5	67	63rPSO264uRjW1X5E6cWv6	Memories (Dillon Francis Remix)	2019-12-13
2	1z1Hg7Vb0AhHDiEmnDE79l	All the Time - Don Diablo Remix	Zara Larsson	70	1HoSmj2eLcsrR0vE9gThr4	All the Time (Don Diablo Remix)	2019-07-05
3	75FpbthrwQmzHlBJLuGdC7	Call You Mine - Keanu Silva Remix	The Chainsmokers	60	1nqYsOef1yKKuGOVchbsk6	Call You Mine - The Remixes	2019-07-19
4	1e8PAfcKUYoKkxPhrHqw4x	Someone You Loved - Future Humans Remix	Lewis Capaldi	69	7m7vv9wlQ4i0LFuJiE2zsQ	Someone You Loved (Future Humans Remix)	2019-03-05

```
  playlist_name             playlist_id playlist_genre  … key  loudness  \
0     Pop Remix  37i9dQZF1DXcZDD7cfEKhW            pop  …   6    -2.634
1     Pop Remix  37i9dQZF1DXcZDD7cfEKhW            pop  …  11    -4.969
2     Pop Remix  37i9dQZF1DXcZDD7cfEKhW            pop  …   1    -3.432
3     Pop Remix  37i9dQZF1DXcZDD7cfEKhW            pop  …   7    -3.778
4     Pop Remix  37i9dQZF1DXcZDD7cfEKhW            pop  …   1    -4.672

   mode  speechiness  acousticness  instrumentalness  liveness  valence  \
0     1       0.0583        0.1020          0.000000    0.0653    0.518
1     1       0.0373        0.0724          0.004210    0.3570    0.693
2     0       0.0742        0.0794          0.000023    0.1100    0.613
3     1       0.1020        0.0287          0.000009    0.2040    0.277
4     1       0.0359        0.0803          0.000000    0.0833    0.725

     tempo  duration_ms
0  122.036       194754
1   99.972       162600
2  124.008       176616
3  121.956       169093
4  123.976       189052

[5 rows x 23 columns]
```

[3]:
```
sns.pairplot(spotify_data)
plt.show()
```

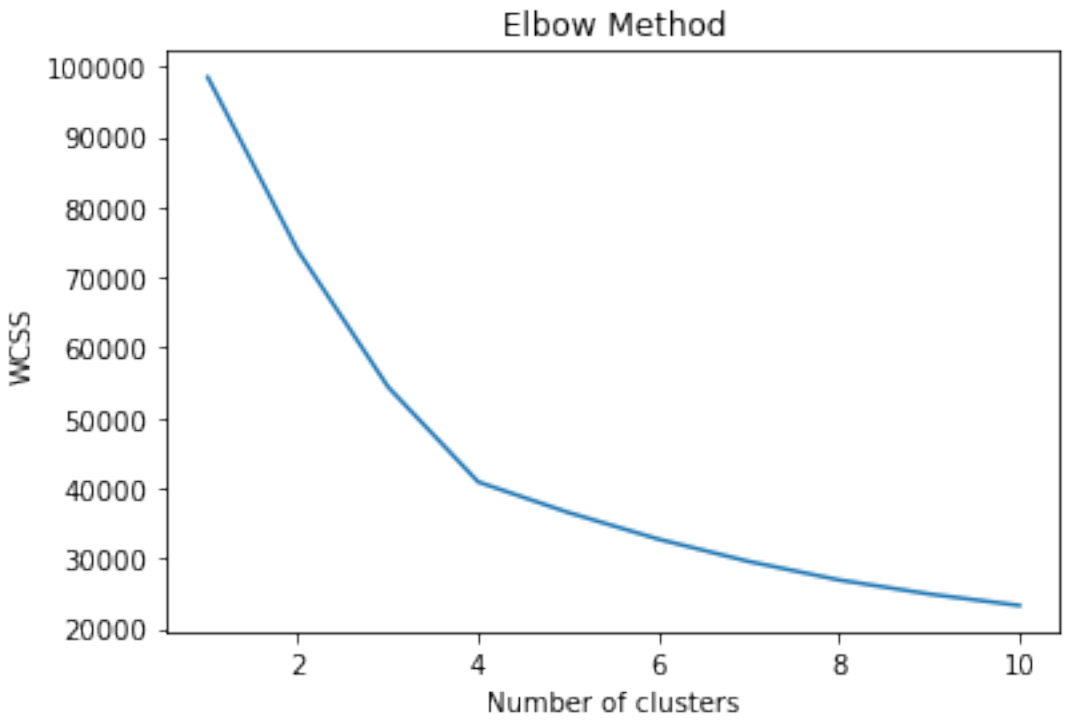


[4]:
```
correlation_matrix = spotify_data.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.show()
```

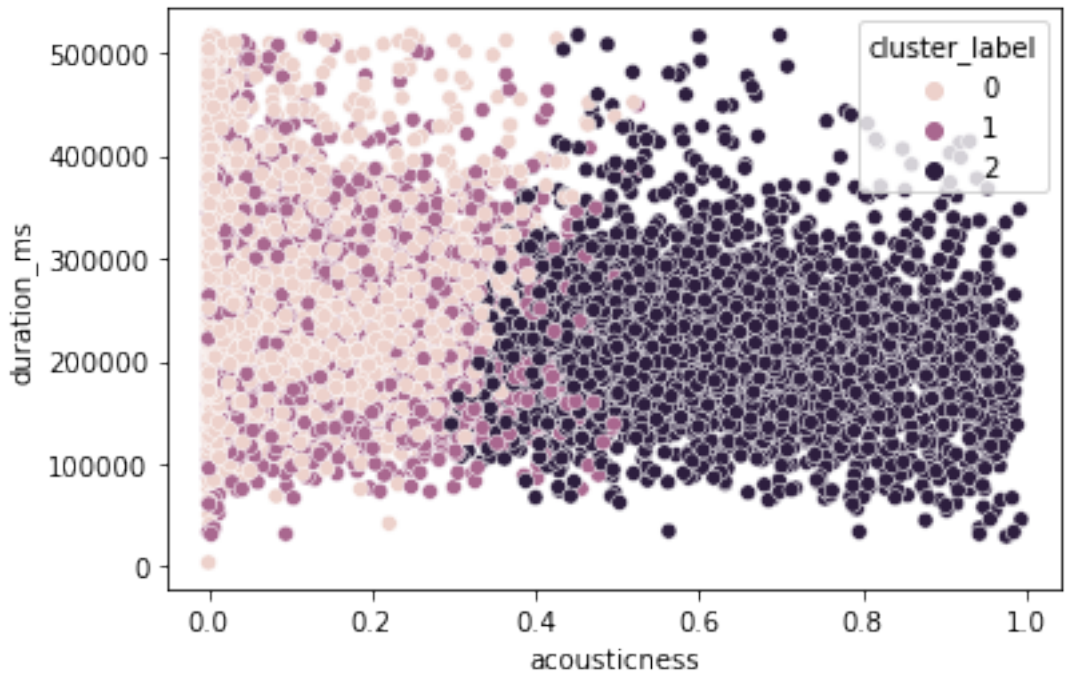


[5]:
```
features_for_clustering = spotify_data[['acousticness', 'duration_ms',␣
 ↪'valence']]
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features_for_clustering)
```

[6]:
```
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10,␣
 ↪random_state=0)
    kmeans.fit(scaled_features)
    wcss.append(kmeans.inertia_)

plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



[7]:
```
kmeans = KMeans(n_clusters=3, init='k-means++', max_iter=300, n_init=10,␣
↪random_state=0)
cluster_labels = kmeans.fit_predict(scaled_features)


spotify_data['cluster_label'] = cluster_labels


sns.scatterplot(x='acousticness', y='duration_ms', hue='cluster_label',␣
↪data=spotify_data)
plt.show()
```



[8]:
```
X = spotify_data[['acousticness', 'duration_ms', 'valence']]
y = spotify_data['playlist_genre']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)


y_pred = rf_model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

```
Accuracy: 0.360590832952642
Classification Report:
              precision    recall  f1-score   support

         edm       0.44      0.49      0.47      1218
       latin       0.35      0.34      0.34      1033
         pop       0.25      0.24      0.25      1081
         r&b       0.30      0.33      0.31      1031
         rap       0.33      0.30      0.31      1168
        rock       0.49      0.45      0.47      1036
```

```
    accuracy                           0.36      6567
   macro avg       0.36      0.36      0.36      6567
weighted avg       0.36      0.36      0.36      6567

Confusion Matrix:
 [[601 102 191  88 138  98]
 [135 348 158 172 152  68]
 [211 168 261 187 136 118]
 [ 99 139 152 340 199 102]
 [191 162 162 209 347  97]
 [124  88 108 151  94 471]]
```

[ ]: