Spotify Genre Prediction

Source

```
import pandas as pd
import matplotlib.pyplot as plt
df_raw = pd.read_csv('SpotifyFeatures.csv')
df_raw.head()
```

```
   genre        artist_name                       track_name  \
0  Movie     Henri Salvador       C'est beau de faire un Show
1  Movie  Martin & les fées  Perdu d'avance (par Gad Elmaleh)
2  Movie    Joseph Williams    Don't Let Me Be Lonely Tonight
3  Movie     Henri Salvador    Dis-moi Monsieur Gordon Cooper
4  Movie       Fabien Nataf                         Ouverture

                 track_id  popularity  acousticness  danceability  \
0  0BRjO6ga9RKCKjfDqeFgWV           0         0.611         0.389
1  0BjC1NfoEOOusryehmNudP           1         0.246         0.590
2  0CoSDzoNIKCRs124s9uTVy           3         0.952         0.663
3  0Gc6TVm52BwZD07Ki6tIvf           0         0.703         0.240
4  0IuslXpMROHdEPvSl1fTQK           4         0.950         0.331

   duration_ms  energy  instrumentalness key  liveness  loudness
mode  \
0        99373   0.910             0.000  C#    0.3460    -1.828
Major
1       137373   0.737             0.000  F#    0.1510    -5.559
Minor
2       170267   0.131             0.000   C    0.1030   -13.879
Minor
3       152427   0.326             0.000  C#    0.0985   -12.178
Major
4        82625   0.225             0.123   F    0.2020   -21.150
Major

   speechiness    tempo time_signature  valence
0       0.0525  166.969            4/4    0.814
1       0.0868  174.003            4/4    0.816
2       0.0362   99.488            5/4    0.368
3       0.0395  171.758            4/4    0.227
4       0.0456  140.576            4/4    0.390
```

```
df_raw.shape
```

```
(232725, 18)
```

```
genre_list = df_raw['genre'].unique()
genre_list
```

```
array(['Movie', 'R&B', 'A Capella', 'Alternative', 'Country', 'Dance',
       'Electronic', 'Anime', 'Folk', 'Blues', 'Opera', 'Hip-Hop',
       "Children's Music", 'Children’s Music', 'Rap', 'Indie',
       'Classical', 'Pop', 'Reggae', 'Reggaeton', 'Jazz', 'Rock',
'Ska',
       'Comedy', 'Soul', 'Soundtrack', 'World'], dtype=object)
```

```
#Preprocessing
df = df_raw[['genre', 'popularity',
'acousticness','danceability','duration_ms','energy', 'instrumentalnes
s',  'key', 'liveness',   'loudness', 'mode', 'speechiness',
     'tempo','time_signature','valence']]
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 232725 entries, 0 to 232724
Data columns (total 15 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   genre             232725 non-null  object
 1   popularity        232725 non-null  int64
 2   acousticness      232725 non-null  float64
 3   danceability      232725 non-null  float64
 4   duration_ms       232725 non-null  int64
 5   energy            232725 non-null  float64
 6   instrumentalness  232725 non-null  float64
 7   key               232725 non-null  object
 8   liveness          232725 non-null  float64
 9   loudness          232725 non-null  float64
 10  mode              232725 non-null  object
 11  speechiness       232725 non-null  float64
 12  tempo             232725 non-null  float64
 13  time_signature    232725 non-null  object
 14  valence           232725 non-null  float64
dtypes: float64(9), int64(2), object(4)
memory usage: 26.6+ MB
```

```
#Preprocessing
df['genre'] = df['genre'].str.replace("Children's Music", "Childrens
Music")
df['genre'].value_counts()
```

```
C:\Users\mithu\AppData\Local\Temp\ipykernel_6260\121275794.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df['genre'] = df['genre'].str.replace("Children's Music", "Childrens
Music")
```

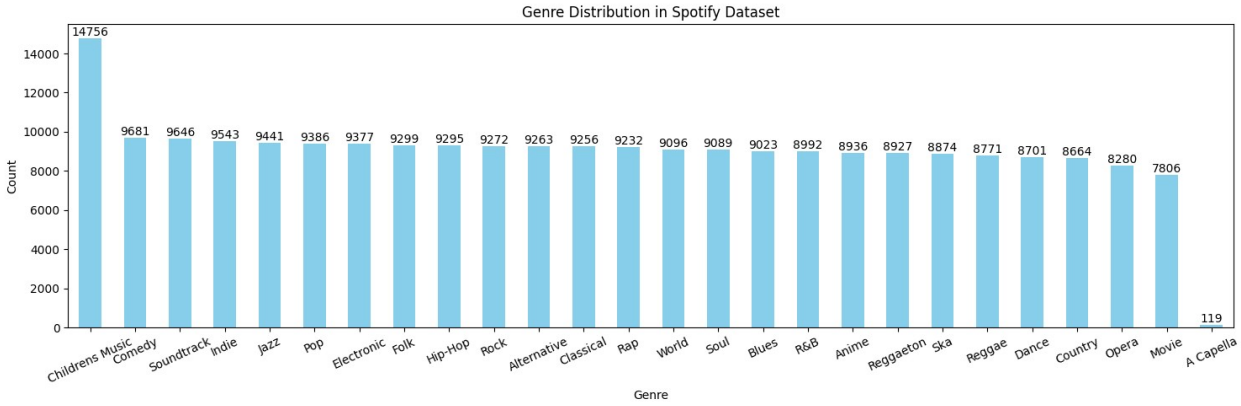
```
genre
Comedy              9681
Soundtrack          9646
Indie               9543
Jazz                9441
Pop                 9386
Electronic          9377
Children’s Music    9353
Folk                9299
Hip-Hop             9295
Rock                9272
Alternative         9263
Classical           9256
Rap                 9232
World               9096
Soul                9089
Blues               9023
R&B                 8992
Anime               8936
Reggaeton           8927
Ska                 8874
Reggae              8771
Dance               8701
Country             8664
Opera               8280
Movie               7806
Childrens Music     5403
A Capella            119
Name: count, dtype: int64
```

```
df['genre'] = df['genre'].str.replace("Children’s Music", "Childrens
Music")
df['genre'].value_counts()
```

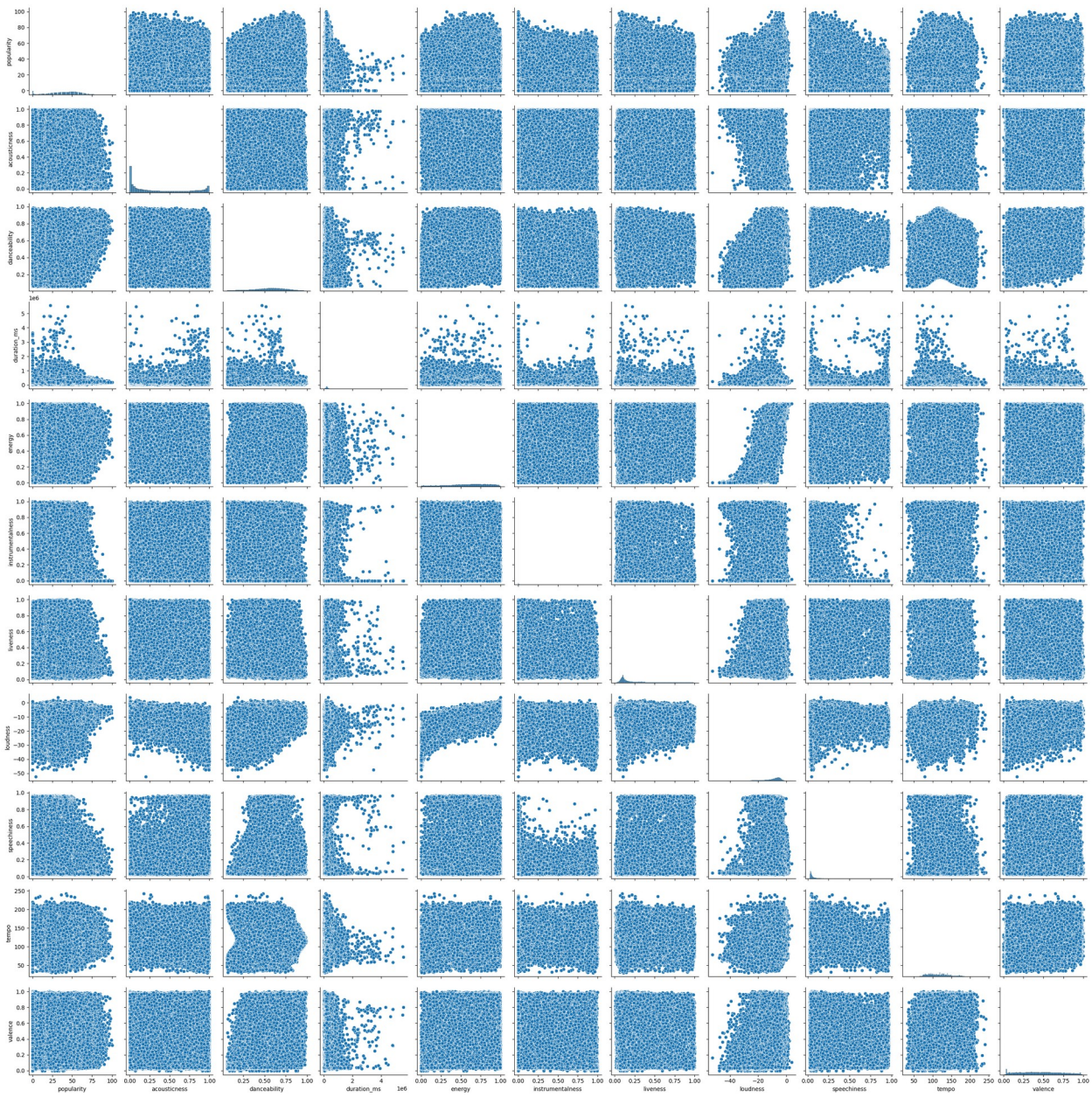
```
C:\Users\mithu\AppData\Local\Temp\ipykernel_6260\497271231.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df['genre'] = df['genre'].str.replace("Children’s Music", "Childrens
Music")
```

```
genre
Childrens Music    14756
Comedy              9681
Soundtrack          9646
Indie               9543
Jazz                9441
Pop                 9386
Electronic          9377
Folk                9299
Hip-Hop             9295
Rock                9272
Alternative         9263
Classical           9256
Rap                 9232
World               9096
Soul                9089
Blues               9023
R&B                 8992
Anime               8936
Reggaeton           8927
Ska                 8874
Reggae              8771
Dance               8701
Country             8664
Opera               8280
Movie               7806
A Capella            119
Name: count, dtype: int64
```

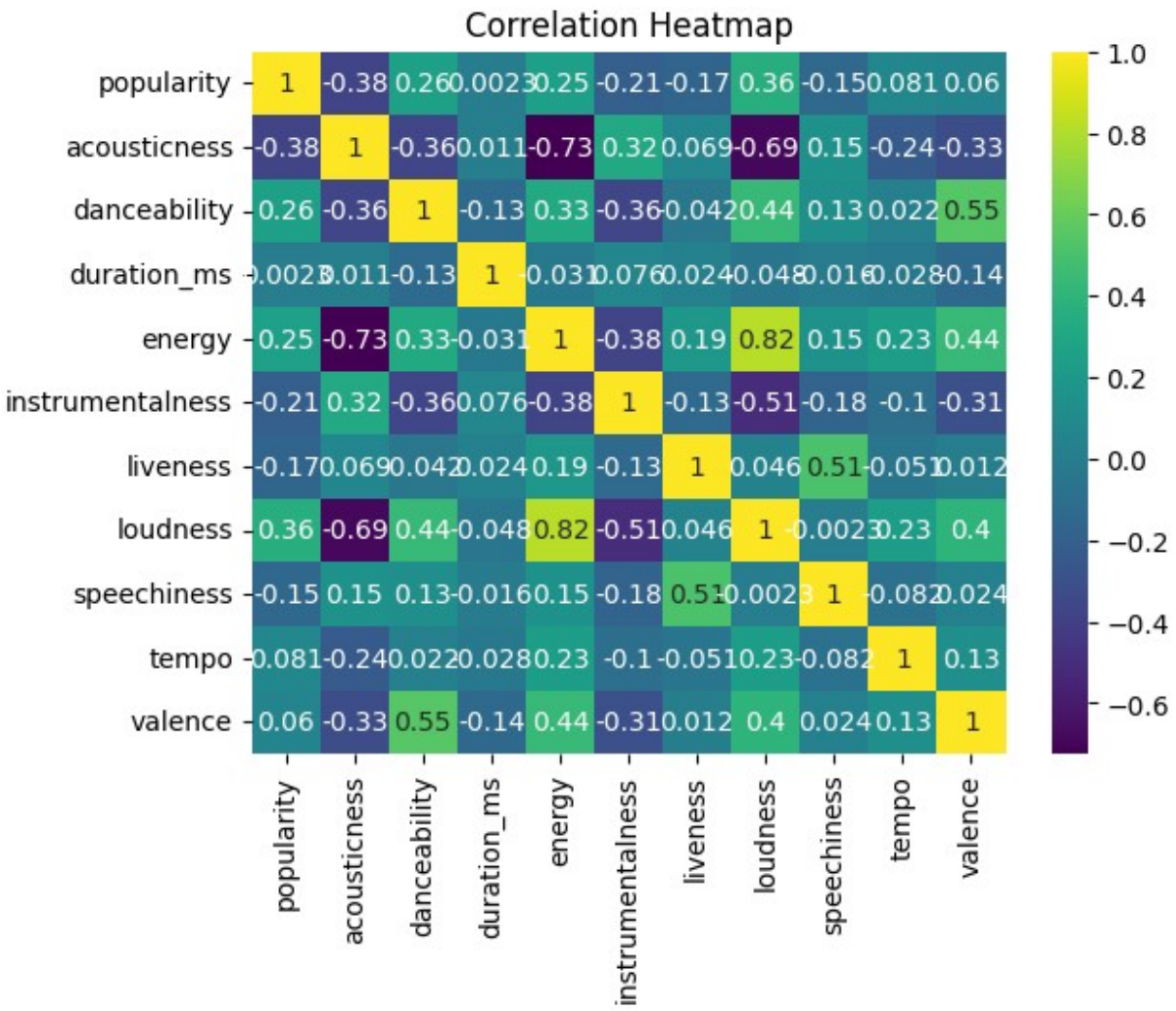
```
genre_counts = df['genre'].value_counts()
plt.figure(figsize=(15, 5))
genre_counts.plot(kind='bar', color='skyblue')
plt.title('Genre Distribution in Spotify Dataset')
plt.xlabel('Genre')
plt.ylabel('Count')
plt.xticks(rotation=25)
plt.tight_layout()
for bar in plt.gca().containers[0]:
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(),
int(bar.get_height()), ha='center', va='bottom')
plt.show()
```




```
#Checking out relationships of variables in this dataset
import seaborn as sns
sns.pairplot(df[['popularity',
'acousticness','danceability','duration_ms','energy','instrumentalness
', 'liveness',  'loudness', 'speechiness', 'tempo','valence']])
plt.show()
```

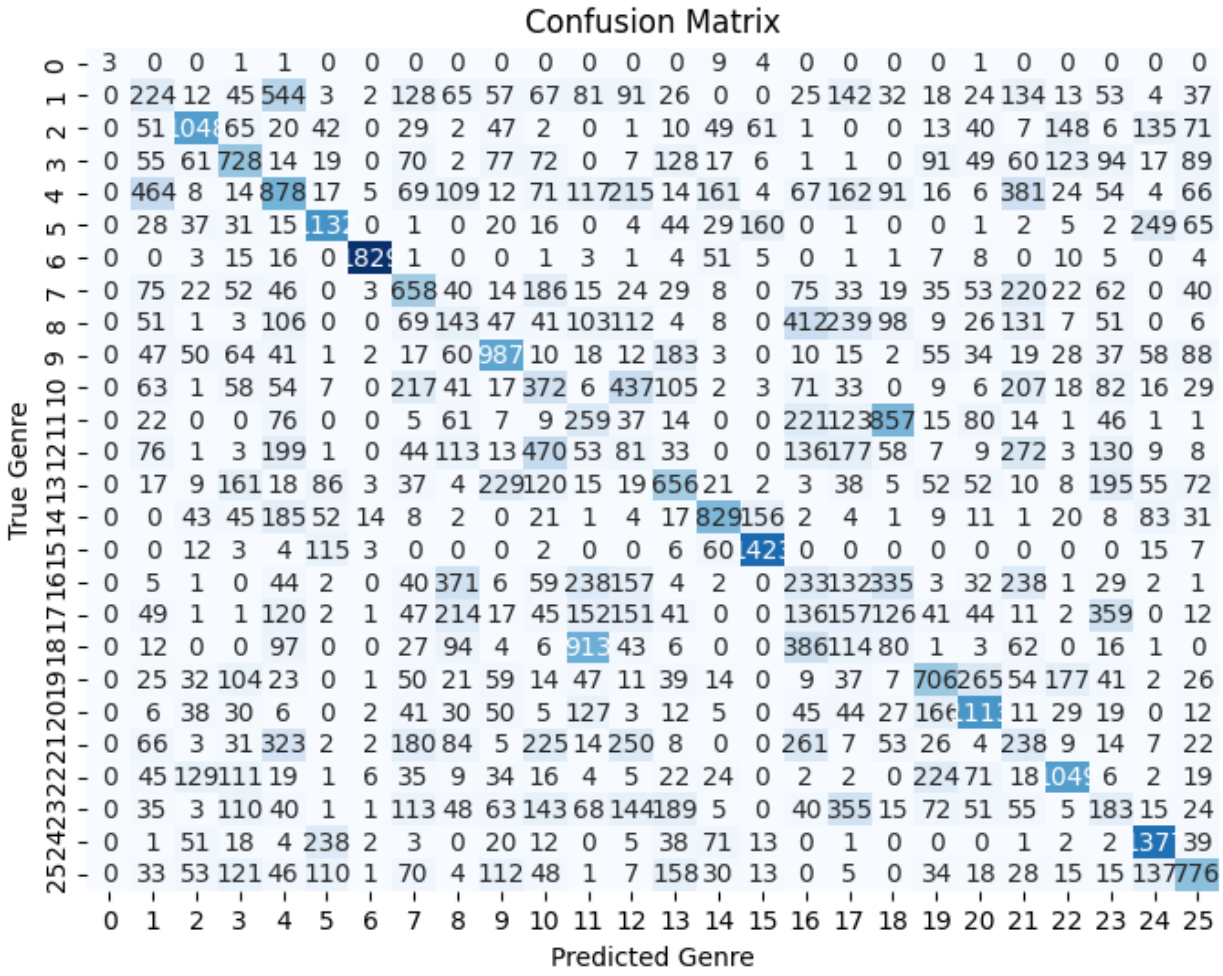



```
#Checking out the correlation matrix
correlation_matrix = df[['popularity',
'acousticness','danceability','duration_ms','energy', 'instrumentalnes
s', 'liveness', 'loudness', 'speechiness', 'tempo','valence']].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='viridis')
plt.title('Correlation Heatmap')
plt.show()
```




```
#Training a random forest classifier to predict genres
X= df[['popularity',
'acousticness','danceability','duration_ms','energy', 'instrumentalnes
s', 'liveness', 'loudness', 'speechiness', 'tempo','valence']].values
y = df['genre'].values
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score, confusion_matrix
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
rf_classifier = RandomForestClassifier(n_estimators=100,
random_state=42)
rf_classifier.fit(X_train, y_train)
y_pred = rf_classifier.predict(X_test)
cm_rf = confusion_matrix(y_test, y_pred)
```

```
plt.figure(figsize=(8, 6))
sns.heatmap(cm_rf, annot=True, fmt='g', cmap='Blues', cbar=False)
plt.xlabel('Predicted Genre')
plt.ylabel('True Genre')
plt.title('Confusion Matrix')
plt.show()
```




```
accuracy_rf = accuracy_score(y_test, y_pred)
accuracy_rf
```

```
0.36871844451605973
```

Closing thoughts

Due to the number of features as well as number of classes (genres) in this data, the model is struggling to find patterns and achieve a good accuracy. We can experiment with neural networks in the future for better performance.

Project by Mithun Meenakshi S.