Music Mood Classification

```
import numpy as np
import pandas as pd
import seaborn as sns

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

```

```
data = pd.read_csv('~/Downloads/EmotionBasedMusicRecommendationSystem/EmotionBasedMusicRecommendationSystem/dataset/data_moods.csv')
data
```

	name	album	artist	id	release_date	popularity	length	danceability	acousticness	energy	i
0	1999	1999	Prince	2H7PHVdQ3mXqEHXcvclTB0	1982-10-27	68	379266	0.866	0.13700	0.7300	
1	23	23	Blonde Redhead	4HlwL9ii9CcXpTOTzMq0MP	2007-04-16	43	318800	0.381	0.01890	0.8320	
2	9 Crimes	9	Damien Rice	5GZEeowhvSieFDiR8fQ2im	2006-11-06	60	217946	0.346	0.91300	0.1390	
3	99 Luftballons	99 Luftballons	Nena	6HA97v4wEGQ5TUClRM0XLc	1984-08-21	2	233000	0.466	0.08900	0.4380	
4	A Boy Brushed Red Living In Black And White	They're Only Chasing Safety	Underoath	47lWLflKOKhFnz1FUEUlkE	2004-01-01	60	268000	0.419	0.00171	0.9320	
...	...	...	...	...	...	...	...	...	...	...	...
681	windcatcher	windcatcher	Leo Nocta	59VApBbrS2lADQk4ml5mdo	2020-06-19	36	123066	0.402	0.96100	0.2360	
682	yellow is the color of her eyes	yellow is the color of her eyes	Soccer Mommy	4D3nttJPU6L0M2epr7sld6	2019-11-19	5	435080	0.452	0.75700	0.5150	
683	you broke me first	you broke me first	Tate McRae	45bE4HXI0AwGZXfZtMp8JR	2020-04-17	87	169265	0.642	0.78600	0.3740	
684	you were good to me	brent	Jeremy Zucker	4CxFN5zON70B3VOPBYbd6P	2019-05-03	76	219146	0.561	0.91300	0.0848	
685	æfre	æfre	praam	2irbT1BSYaIEF44PlyKaoM	2020-07-17	41	186331	0.377	0.99400	0.0156	

686 rows × 19 columns

```
# Check for missing values
missing_values = data.isnull().sum()
missing_values
```

```
name               0
album              0
artist             0
id                 0
release_date       0
popularity         0
length             0
danceability       0
acousticness       0
energy             0
instrumentalness   0
liveness           0
valence            0
loudness           0
speechiness        0
tempo              0
key                0
time_signature     0
mood               0
dtype: int64
```

```
# Select relevant features
selected_features = ['danceability', 'acousticness', 'energy', 'instrumentalness', 'liveness', 'valence', 'loudness', 'speechiness', 'tempo']
X = data[selected_features]
y = data['mood']
```

```
# Normalization
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
# Categorical Encoding
from sklearn.preprocessing import LabelEncoder

label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
```

```
# Split the Data
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y_encoded, test_size=0.2, random_state=42)

```

```
# Train a Mode
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

```

```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
# Evaluate the Model
from sklearn.metrics import classification_report

y_pred = model.predict(X_test)
report = classification_report(y_test, y_pred, target_names=label_encoder.classes_)
print(report)

```

```
              precision    recall  f1-score   support

        Calm       0.97      0.95      0.96        41
   Energetic       0.71      0.75      0.73        32
       Happy       0.50      0.48      0.49        25
         Sad       0.88      0.88      0.88        40

    accuracy                           0.80       138
   macro avg       0.76      0.76      0.76       138
weighted avg       0.80      0.80      0.80       138
```

```
# Add the predicted moods to the dataset
data['encoded_mood'] = label_encoder.transform(data['mood'])
data['predicted_mood'] = label_encoder.inverse_transform(model.predict(X_scaled))
```

```
# Save the updated dataset to a new CSV file
output_file_path = 'path_to_save_the_new_file/updated_data_moods.csv'
data.to_csv(output_file_path, index=False)


```

```
# Display the first few rows of the updated dataset
print(data[['name', 'album', 'artist', 'mood', 'predicted_mood']].head())
```

```
                                       name                       album  \
0                                      1999                        1999
1                                        23                          23
2                                 9 Crimes                           9
3                           99 Luftballons              99 Luftballons
4  A Boy Brushed Red Living In Black And White  They're Only Chasing Safety

           artist      mood predicted_mood
0          Prince     Happy          Happy
1  Blonde Redhead       Sad            Sad
2     Damien Rice       Sad            Sad
```

3	Nena	Happy	Happy
4	Underoath	Energetic	Energetic

```
Start coding or generate with AI.
```