```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# Load the dataset
df = pd.read_csv("dataset.csv")  # Replace with actual dataset path

# Define mood categories based on valence, energy, and danceability
def classify_mood(row):
    if row['valence'] > 0.6 and row['energy'] > 0.6:
        return "Happy"
    elif row['valence'] < 0.4 and row['energy'] < 0.4:
        return "Sad"
    elif row['energy'] > 0.7 and row['danceability'] > 0.6:
        return "Energetic"
    else:
        return "Calm"

# Apply the function to create a 'mood' column
df['mood'] = df.apply(classify_mood, axis=1)

# Encode mood labels into numerical values
label_encoder = LabelEncoder()
df['mood'] = label_encoder.fit_transform(df['mood'])

# Select features and target variable
features = ['danceability', 'energy', 'valence', 'tempo',
'acousticness', 'instrumentalness', 'liveness', 'speechiness']
X = df[features]
y = df['mood']

# Split 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)

# Normalize features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train a Random Forest Classifier
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predictions
```

```
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
print(classification_report(y_test, y_pred,
target_names=label_encoder.classes_))

# Function to classify a new song
def classify_song(features):
    features = np.array(features).reshape(1, -1)
    features = scaler.transform(features)
    mood_index = model.predict(features)[0]
    return label_encoder.inverse_transform([mood_index])[0]

# Example usage
new_song_features = [0.8, 0.7, 0.6, 120, 0.1, 0.0, 0.2, 0.05]
print("Predicted Mood:", classify_song(new_song_features))
```

```
Accuracy: 1.00
              precision    recall  f1-score   support

        Calm       1.00      1.00      1.00     12228
   Energetic       1.00      1.00      1.00      1998
       Happy       1.00      1.00      1.00      5773
         Sad       1.00      1.00      1.00      2801

    accuracy                           1.00     22800
   macro avg       1.00      1.00      1.00     22800
weighted avg       1.00      1.00      1.00     22800

Predicted Mood: Calm

C:\Users\manas\anaconda3\Lib\site-packages\sklearn\base.py:493:
UserWarning: X does not have valid feature names, but StandardScaler
was fitted with feature names
  warnings.warn(
```