In [1]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
```

In [2]:
```
df = pd.read_csv('spotify1.csv')
print(df.head(2))  # Display first 2 rows
print(df.info())  # Get data info
print(df.isnull().sum())
```

```
                       id                                name  duration  \
0  4ZtFanR9U6ndgddUvNcjcG             Good 4 U Olivia Rodrigo      2.97   
1  5fxyZf6m2xHeSrOzUfcJrq  Stay The Kid LAROI & Justin Bieber      2.30   

   energy  key  loudness  mode  speechiness  acousticness  instrumentalness  \
0   0.664    9    -5.044     1       0.1540         0.335             0.000   
1   0.506    8   -11.275     1       0.0589         0.379             0.868   

   liveness  valence    tempo  danceability timestamp  
0    0.0849    0.688  166.928         0.563  08:00:00  
1    0.1100    0.454  170.054         0.564  08:00:00  
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 15 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   id                100 non-null    object 
 1   name              100 non-null    object 
 2   duration          100 non-null    float64
 3   energy            100 non-null    float64
 4   key               100 non-null    int64  
 5   loudness          100 non-null    float64
 6   mode              100 non-null    int64  
 7   speechiness       100 non-null    float64
 8   acousticness      100 non-null    float64
 9   instrumentalness  100 non-null    float64
 10  liveness          100 non-null    float64
 11  valence           100 non-null    float64
 12  tempo             100 non-null    float64
 13  danceability      100 non-null    float64
 14  timestamp         100 non-null    object 
dtypes: float64(10), int64(2), object(3)
memory usage: 11.8+ KB
None
id                  0
name                0
duration            0
energy              0
key                 0
loudness            0
mode                0
speechiness         0
acousticness        0
instrumentalness    0
liveness            0
valence             0
tempo               0
danceability        0
timestamp           0
dtype: int64
```

In [3]:
```
l_id = LabelEncoder()
df['n_id'] = l_id.fit_transform(df['id'])
```

In [4]:
```
l_name = LabelEncoder()
df['n_name'] = l_name.fit_transform(df['name'])
```

In [5]:
```
data = df.drop(['id', 'name'], axis='columns')
```

In [9]:
```
data['timestamp'] = pd.to_datetime(data['timestamp'])
data['hour'] = data['timestamp'].dt.hour
data['day_of_week'] = data['timestamp'].dt.dayofweek
data['month'] = data['timestamp'].dt.month
```

In [10]:
```
data['play_count'] = data.groupby(['n_id', 'n_name'])['timestamp'].transform('count')
```

In [12]:
```
data['repeated_plays'] = (data['play_count'] > 1).astype(int)
```

In [13]:
```
x = data[['n_id', 'n_name', 'duration', 'energy', 'key', 'loudness', 'mode', 'speechiness',
          'instrumentalness', 'liveness', 'valence', 'tempo', 'danceability', 'hour', 'day_of_week',
          'month', 'play_count']]
y = data['repeated_plays']
```

In [14]:
```
scaler = StandardScaler()
```

In [14]:

```
scaler = StandardScaler()
x_scaled = scaler.fit_transform(x)
```

In [15]:

```
x_train, x_test, y_train, y_test = train_test_split(x_scaled, y, test_size=0.2, random_state=42)
```

In [16]:

```
model = LogisticRegression()
model.fit(x_train, y_train)
```

Out[16]:

```
LogisticRegression()
```

In [17]:

```
y_pred = model.predict(x_test)
print("Model predictions:", y_pred)
```

```
Model predictions: [0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 0 0]
```

In [18]:

```
model.score(x_train,y_train)
```

Out[18]:

```
1.0
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js