Spotify.Description.for.Students

November 5, 2024

1 Project 4: Music Popularity Prediction

This project will take data features collected for songs that have been on the Top 200 Weekly (Global) charts of Spotify in 2020 & 2021. The popularity of the song will be predicted using a tree-based regression model trained on these features.

The goals for the project are:

- Minimize the cross-validated ***root mean squared error (RMSE)*** when predicting the popularity of a new song.
- Determine the importance of the features in driving the regression result. The project will be done using tree-based regression techniques as covered in class. The parameters of the trees should be carefully selected to avoid over-fitting.

There are three main challenges for this project:

1. Determining the outcome (i.e. target). There is a "popularity" column. But other columns may or may not be more appropriate indicators of popularity.
2. Choosing appropriate predictors (i.e. features). When building a machine learning model, we want to make sure that we consider how the model will be ultimately used. For this project, we are predicting the popularity of a new song. Therefore, we should only include the predictors we would have for a new song.
3. Data cleaning and feature engineering. Some creative cleaning and/or feature engineering may be needed to extract useful information for prediction.

Once again, be sure to go through the whole data science process and document as such in your Jupyter notebook.

The data is available AWS at https://ddc-datascience.s3.amazonaws.com/Projects/Project.4-Spotify/Data/Spotify.csv .

2 Imports

[591]:
```
import sys
print(sys.executable)
```

```
/usr/local/bin/python
```

[592]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
import xgboost as xgb

from sklearn.metrics import mean_squared_error, root_mean_squared_error,r2_score
```

[ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
#n_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
import xgboost as xgb

from sklearn.metrics import mean_squared_error, root_mean_squared_error,r2_score
```

[594]:
```
%%capture
url = "https://ddc-datascience.s3.amazonaws.com/Projects/Project.4-Spotify/Data/
 ↪Spotify.csv"
!curl -s -I {url}
```

3 Data Exploration

[595]:
```
df_1 = pd.read_csv(url).copy()
```

3.1 Head

[596]:
```
df_1.head()
```

[596]:

	Index	Highest Charting Position	Number of Times Charted	\
0	1	1	8	
1	2	2	3	
2	3	1	11	
3	4	3	5	

```
4      5                          5                          1

  Week of Highest Charting                             Song Name     Streams  \
0   2021-07-23--2021-07-30                               Beggin'  48,633,449
1   2021-07-23--2021-07-30                 STAY (with Justin Bieber)  47,248,719
2   2021-06-25--2021-07-02                                 good 4 u  40,162,559
3   2021-07-02--2021-07-09                               Bad Habits  37,799,456
4   2021-07-23--2021-07-30  INDUSTRY BABY (feat. Jack Harlow)  33,948,454

           Artist Artist Followers                 Song ID  \
0        Måneskin          3377762  3Wrjm47oTz2sjIgck11l5e
1   The Kid LAROI          2230022  5HCyWlXZPP0y6Gqq8TgA20
2  Olivia Rodrigo          6266514  4ZtFanR9U6ndgddUvNcjcG
3      Ed Sheeran         83293380  6PQ88X9TkUIAUIZJHW2upE
4       Lil Nas X          5473565  27NovPIUIRrOZoCHxABJwK

                                    Genre  … Danceability Energy Loudness  \
0  ['indie rock italiano', 'italian pop']  …        0.714    0.8   -4.808
1                  ['australian hip hop']  …        0.591  0.764   -5.484
2                                 ['pop']  …        0.563  0.664   -5.044
3                     ['pop', 'uk pop']  …        0.808  0.897   -3.712
4           ['lgbtq+ hip hop', 'pop rap']  …        0.736  0.704   -7.409

  Speechiness Acousticness Liveness    Tempo Duration (ms) Valence  Chord
0      0.0504        0.127    0.359  134.002        211560   0.589      B
1      0.0483       0.0383    0.103  169.928        141806   0.478  C#/Db
2       0.154        0.335   0.0849  166.928        178147   0.688      A
3      0.0348       0.0469    0.364  126.026        231041   0.591      B
4      0.0615       0.0203   0.0501  149.995        212000   0.894  D#/Eb

[5 rows x 23 columns]
```

3.2 Tail

[ ]:
```
df_1.tail()dt_model.fit(X_train_std, y_train_1)

feature_importances = dt_model.feature_importances_
feature_names = X_train_1.columns
feature_importance_df = pd.DataFrame({'feature': feature_names, 'importance':␣
 ↪feature_importances})
feature_importance_df = feature_importance_df.sort_values('importance',␣
 ↪ascending=False)
print(feature_importance_df)
```

[ ]:
```
      Index  Highest Charting Position  Number of Times Charted  \
1551   1552                        195                        1
1552   1553                        196                        1
```

```
1553   1554                        197                         1
1554   1555                        198                         1
1555   1556                        199                         1

     Week of Highest Charting                            Song Name    Streams  \
1551   2019-12-27--2020-01-03                            New Rules  4,630,675
1552   2019-12-27--2020-01-03                    Cheirosa - Ao Vivo  4,623,030
1553   2019-12-27--2020-01-03              Havana (feat. Young Thug)  4,620,876
1554   2019-12-27--2020-01-03             Surtada - Remix Brega Funk  4,607,385
1555   2019-12-27--2020-01-03  Lover (Remix) [feat. Shawn Mendes]  4,595,450

                             Artist Artist Followers                 Song ID  \
1551                       Dua Lipa         27167675  2ekn2ttSfGqwhhate0LSR0
1552                  Jorge & Mateus         15019109  2PWjKmjyTZeDpmOUa3a5da
1553                  Camila Cabello         22698747  1rfofaqEpACxVEHIZBJe6W
1554  Dadá Boladão, Tati Zaqui, OIK           208630  5F8ffc8KWKNawllr5WsW0r
1555                     Taylor Swift         42227614  3i9UVldZOE0aD0JnyfAZZ0

                                              Genre  … Danceability  \
1551                    ['dance pop', 'pop', 'uk pop']  …        0.762
1552           ['sertanejo', 'sertanejo universitario']  …        0.528
1553  ['dance pop', 'electropop', 'pop', 'post-teen …  …        0.765
1554                    ['brega funk', 'funk carioca']  …        0.832
1555                         ['pop', 'post-teen pop']  …        0.448

     Energy Loudness Speechiness Acousticness Liveness    Tempo Duration (ms)  \
1551    0.7   -6.021      0.0694      0.00261    0.153  116.073        209320
1552   0.87   -3.123      0.0851         0.24    0.333   152.37        181930
1553  0.523   -4.333        0.03        0.184    0.132  104.988        217307
1554   0.55   -7.026      0.0587        0.249    0.182  154.064        152784
1555  0.603   -7.176       0.064        0.433   0.0862  205.272        221307

     Valence Chord
1551   0.608     A
1552   0.714     B
1553   0.394     D
1554   0.881     F
1555   0.422     G

[5 rows x 23 columns]
```

3.3 Shape

[598]:
```
df_1.shape
```

[598]:
```
(1556, 23)
```

3.4 columns

[599]:
```
df_1.columns
```

[599]:
```
Index(['Index', 'Highest Charting Position', 'Number of Times Charted',
       'Week of Highest Charting', 'Song Name', 'Streams', 'Artist',
       'Artist Followers', 'Song ID', 'Genre', 'Release Date', 'Weeks Charted',
       'Popularity', 'Danceability', 'Energy', 'Loudness', 'Speechiness',
       'Acousticness', 'Liveness', 'Tempo', 'Duration (ms)', 'Valence',
       'Chord'],
      dtype='object')
```

3.5 Dtypes

[600]:
```
df_1.dtypes
```

[600]:
```
Index                          int64
Highest Charting Position      int64
Number of Times Charted        int64
Week of Highest Charting      object
Song Name                     object
Streams                       object
Artist                        object
Artist Followers              object
Song ID                       object
Genre                         object
Release Date                  object
Weeks Charted                 object
Popularity                    object
Danceability                  object
Energy                        object
Loudness                      object
Speechiness                   object
Acousticness                  object
Liveness                      object
Tempo                         object
Duration (ms)                 object
Valence                       object
Chord                         object
dtype: object
```

3.6 Describe

[601]:
```
df_1.describe()
```

[601]:

	Index	Highest Charting Position	Number of Times Charted
count	1556.000000	1556.000000	1556.000000
mean	778.500000	87.744216	10.668380

```
std     449.322824                58.147225                16.360546
min       1.000000                 1.000000                 1.000000
25%     389.750000                37.000000                 1.000000
50%     778.500000                80.000000                 4.000000
75%    1167.250000               137.000000                12.000000
max    1556.000000               200.000000               142.000000
```

3.7 Isnull Sum

[602]:
```
df_1.isnull().sum()
```

[602]:
```
Index                         0
Highest Charting Position     0
Number of Times Charted       0
Week of Highest Charting      0
Song Name                     0
Streams                       0
Artist                        0
Artist Followers              0
Song ID                       0
Genre                         0
Release Date                  0
Weeks Charted                 0
Popularity                    0
Danceability                  0
Energy                        0
Loudness                      0
Speechiness                   0
Acousticness                  0
Liveness                      0
Tempo                         0
Duration (ms)                 0
Valence                       0
Chord                         0
dtype: int64
```

3.8 Isna Sum

[603]:
```
df_1.isna().sum()
```

[603]:
```
Index                         0
Highest Charting Position     0
Number of Times Charted       0
Week of Highest Charting      0
Song Name                     0
Streams                       0
Artist                        0
```

```
Artist Followers            0
Song ID                     0
Genre                       0
Release Date                0
Weeks Charted               0
Popularity                  0
Danceability                0
Energy                      0
Loudness                    0
Speechiness                 0
Acousticness                0
Liveness                    0
Tempo                       0
Duration (ms)               0
Valence                     0
Chord                       0
dtype: int64
```

3.9 unique values

[604]:
```
df_1.count('rows').unique().sum()
```

[604]:
```
np.int64(1556)
```

[605]:
```
df_1.count('columns')
```

[605]:
```
0       23
1       23
2       23
3       23
4       23
        ..
1551    23
1552    23
1553    23
1554    23
1555    23
Length: 1556, dtype: int64
```

3.10 Sort_values

[606]:
```
df_1.sort_values(by = ['Popularity'], ascending = False).head(10)
```

[606]:
```
     Index  Highest Charting Position  Number of Times Charted  \
1        2                          2                        3
2        3                          1                       11
3        4                          3                        5
```

```
5       6                          1                       18
4       5                          5                        1
8       9                          3                        8
14     15                          2                       10
7       8                          2                       10
9      10                          8                       10
11     12                          9                        9
```

```
   Week of Highest Charting                         Song Name     Streams  \
1    2021-07-23--2021-07-30          STAY (with Justin Bieber)  47,248,719
2    2021-06-25--2021-07-02                            good 4 u  40,162,559
3    2021-07-02--2021-07-09                         Bad Habits  37,799,456
5    2021-05-07--2021-05-14     MONTERO (Call Me By Your Name)  30,071,134
4    2021-07-23--2021-07-30  INDUSTRY BABY (feat. Jack Harlow)  33,948,454
8    2021-06-18--2021-06-25                           Yonaguni  25,030,128
14   2021-05-21--2021-05-28                             Butter  19,985,713
7    2021-06-18--2021-06-25                         Todo De Ti  26,951,613
9    2021-07-02--2021-07-09              I WANNA BE YOUR SLAVE  24,551,591
11   2021-07-02--2021-07-09                    Qué Más Pues?  22,405,111

                    Artist Artist Followers                Song ID  \
1             The Kid LAROI          2230022  5HCyWlXZPP0y6Gqq8TgA20
2           Olivia Rodrigo          6266514  4ZtFanR9U6ndgddUvNcjcG
3                Ed Sheeran         83293380  6PQ88X9TkUIAUIZJHW2upE
5                 Lil Nas X          5473565  67BtfxlNbhBmCDR2L2l8qd
4                 Lil Nas X          5473565  27NovPIUIRrOZoCHxABJwK
8                 Bad Bunny         36142273  2JPLbjOn0wPCngEot2STUS
14                      BTS         37106176  2bgTY4UwhfBYhGT4HUYStN
7            Rauw Alejandro          6080597  4fSIb4hdOQ151TILNsSEaF
9                 Måneskin          3377762  4pt5fDVTg5GhEvEtlz9dKk
11  J Balvin, Maria Becerra         29051363  6hf0RpxTbOprT5nnwzkk8e

                                        Genre  … Danceability Energy  \
1                  ['australian hip hop']  …        0.591  0.764
2                                  ['pop']  …        0.563  0.664
3                      ['pop', 'uk pop']  …        0.808  0.897
5          ['lgbtq+ hip hop', 'pop rap']  …         0.61  0.508
4          ['lgbtq+ hip hop', 'pop rap']  …        0.736  0.704
8    ['latin', 'reggaeton', 'trap latino']  …        0.644  0.648
14           ['k-pop', 'k-pop boy group']  …        0.759  0.459
7      ['puerto rican pop', 'trap latino']  …         0.78  0.718
9    ['indie rock italiano', 'italian pop']  …         0.75  0.608
11  ['latin', 'reggaeton', 'reggaeton colombiano']  …        0.891  0.819

   Loudness Speechiness Acousticness Liveness    Tempo Duration (ms) Valence  \
1    -5.484      0.0483       0.0383    0.103  169.928        141806   0.478
2    -5.044       0.154        0.335   0.0849  166.928        178147   0.688
```

```
3    -3.712      0.0348       0.0469    0.364  126.026        231041   0.591
5    -6.682       0.152        0.297    0.384  178.818        137876   0.758
4    -7.409      0.0615       0.0203   0.0501  149.995        212000   0.894
8    -4.601       0.118        0.276    0.135  179.951        206710    0.44
14   -5.187      0.0948      0.00323   0.0906  109.997        164442   0.695
7    -3.605      0.0506         0.31   0.0932  127.949        199604   0.342
9    -4.008      0.0387      0.00165    0.178  132.507        173347   0.958
11   -3.964       0.106       0.0261    0.173  101.968        217773   0.768

    Chord
1   C#/Db
2       A
3       B
5   G#/Ab
4   D#/Eb
8   C#/Db
14  G#/Ab
7   D#/Eb
9   C#/Db
11  G#/Ab

[10 rows x 23 columns]
```

4 Data Cleaning and Feature Engineering

4.1 New copy of dataframe

[607]:
```
df_cleaning = df_1.copy()
df_cleaning
```

[607]:
```
      Index  Highest Charting Position  Number of Times Charted  \
0         1                          1                        8
1         2                          2                        3
2         3                          1                       11
3         4                          3                        5
4         5                          5                        1
…     …                          …                        …
1551   1552                        195                        1
1552   1553                        196                        1
1553   1554                        197                        1
1554   1555                        198                        1
1555   1556                        199                        1

     Week of Highest Charting                      Song Name     Streams  \
0      2021-07-23--2021-07-30                        Beggin'  48,633,449
1      2021-07-23--2021-07-30  STAY (with Justin Bieber)  47,248,719
2      2021-06-25--2021-07-02                      good 4 u  40,162,559
```

```
3      2021-07-02--2021-07-09                          Bad Habits  37,799,456
4      2021-07-23--2021-07-30   INDUSTRY BABY (feat. Jack Harlow)  33,948,454
…                         …                                   …           …
1551   2019-12-27--2020-01-03                          New Rules   4,630,675
1552   2019-12-27--2020-01-03                 Cheirosa - Ao Vivo   4,623,030
1553   2019-12-27--2020-01-03          Havana (feat. Young Thug)   4,620,876
1554   2019-12-27--2020-01-03         Surtada - Remix Brega Funk   4,607,385
1555   2019-12-27--2020-01-03  Lover (Remix) [feat. Shawn Mendes]   4,595,450

                           Artist Artist Followers                Song ID  \
0                        Måneskin          3377762  3Wrjm47oTz2sjIgck11l5e
1                     The Kid LAROI          2230022  5HCyWlXZPP0y6Gqq8TgA20
2                    Olivia Rodrigo          6266514  4ZtFanR9U6ndgddUvNcjcG
3                        Ed Sheeran         83293380  6PQ88X9TkUIAUIZJHW2upE
4                         Lil Nas X          5473565  27NovPIUIRrOZoCHxABJwK
…                               …                …                       …
1551                      Dua Lipa         27167675  2ekn2ttSfGqwhhate0LSR0
1552                Jorge & Mateus         15019109  2PWjKmjyTZeDpmOUa3a5da
1553                Camila Cabello         22698747  1rfofaqEpACxVEHIZBJe6W
1554  Dadá Boladão, Tati Zaqui, OIK           208630  5F8ffc8KWKNawllr5WsW0r
1555                  Taylor Swift         42227614  3i9UVldZOE0aD0JnyfAZZ0

                                          Genre  … Danceability  \
0     ['indie rock italiano', 'italian pop']  …          0.714
1                       ['australian hip hop']  …          0.591
2                                      ['pop']  …          0.563
3                           ['pop', 'uk pop']  …          0.808
4               ['lgbtq+ hip hop', 'pop rap']  …          0.736
…                                         …  …              …
1551          ['dance pop', 'pop', 'uk pop']  …          0.762
1552   ['sertanejo', 'sertanejo universitario']  …          0.528
1553  ['dance pop', 'electropop', 'pop', 'post-teen …  …          0.765
1554          ['brega funk', 'funk carioca']  …          0.832
1555                ['pop', 'post-teen pop']  …          0.448

     Energy Loudness Speechiness Acousticness Liveness    Tempo Duration (ms)  \
0       0.8   -4.808      0.0504        0.127    0.359  134.002        211560
1     0.764   -5.484      0.0483       0.0383    0.103  169.928        141806
2     0.664   -5.044       0.154        0.335   0.0849  166.928        178147
3     0.897   -3.712      0.0348       0.0469    0.364  126.026        231041
4     0.704   -7.409      0.0615       0.0203   0.0501  149.995        212000
…     …       …           …            …        …       …              …
1551    0.7   -6.021      0.0694      0.00261    0.153  116.073        209320
1552   0.87   -3.123      0.0851         0.24    0.333   152.37        181930
1553  0.523   -4.333        0.03        0.184    0.132  104.988        217307
1554   0.55   -7.026      0.0587        0.249    0.182  154.064        152784
1555  0.603   -7.176       0.064        0.433   0.0862  205.272        221307
```

```
     Valence  Chord
0      0.589      B
1      0.478  C#/Db
2      0.688      A
3      0.591      B
4      0.894  D#/Eb
…      …    …
1551   0.608      A
1552   0.714      B
1553   0.394      D
1554   0.881      F
1555   0.422      G

[1556 rows x 23 columns]
```

4.2 drop Index

[ ]:
```
df_cleaning.drop('Index', axis = 1, inplace = True)
#
```

[609]:
```
df_cleaning.transpose()
```

[609]:
```
                                                                    0     \
Highest Charting Position                                             1
Number of Times Charted                                               8
Week of Highest Charting                          2021-07-23--2021-07-30
Song Name                                                      Beggin'
Streams                                                      48,633,449
Artist                                                         Måneskin
Artist Followers                                                3377762
Song ID                                           3Wrjm47oTz2sjIgck11l5e
Genre                          ['indie rock italiano', 'italian pop']
Release Date                                                2017-12-08
Weeks Charted             2021-07-23--2021-07-30\n2021-07-16--2021-07-23…
Popularity                                                          100
Danceability                                                      0.714
Energy                                                              0.8
Loudness                                                         -4.808
Speechiness                                                      0.0504
Acousticness                                                      0.127
Liveness                                                          0.359
Tempo                                                           134.002
Duration (ms)                                                    211560
Valence                                                           0.589
Chord                                                                B
```

	1	\
Highest Charting Position	2	
Number of Times Charted	3	
Week of Highest Charting	2021-07-23--2021-07-30	
Song Name	STAY (with Justin Bieber)	
Streams	47,248,719	
Artist	The Kid LAROI	
Artist Followers	2230022	
Song ID	5HCyWlXZPP0y6Gqq8TgA20	
Genre	['australian hip hop']	
Release Date	2021-07-09	
Weeks Charted	2021-07-23--2021-07-30\n2021-07-16--2021-07-23…	
Popularity	99	
Danceability	0.591	
Energy	0.764	
Loudness	-5.484	
Speechiness	0.0483	
Acousticness	0.0383	
Liveness	0.103	
Tempo	169.928	
Duration (ms)	141806	
Valence	0.478	
Chord	C#/Db	

	2	\
Highest Charting Position	1	
Number of Times Charted	11	
Week of Highest Charting	2021-06-25--2021-07-02	
Song Name	good 4 u	
Streams	40,162,559	
Artist	Olivia Rodrigo	
Artist Followers	6266514	
Song ID	4ZtFanR9U6ndgddUvNcjcG	
Genre	['pop']	
Release Date	2021-05-21	
Weeks Charted	2021-07-23--2021-07-30\n2021-07-16--2021-07-23…	
Popularity	99	
Danceability	0.563	
Energy	0.664	
Loudness	-5.044	
Speechiness	0.154	
Acousticness	0.335	
Liveness	0.0849	
Tempo	166.928	
Duration (ms)	178147	
Valence	0.688	
Chord	A	

	3 \
Highest Charting Position	3
Number of Times Charted	5
Week of Highest Charting	2021-07-02--2021-07-09
Song Name	Bad Habits
Streams	37,799,456
Artist	Ed Sheeran
Artist Followers	83293380
Song ID	6PQ88X9TkUIAUIZJHW2upE
Genre	['pop', 'uk pop']
Release Date	2021-06-25
Weeks Charted	2021-07-23--2021-07-30\n2021-07-16--2021-07-23…
Popularity	98
Danceability	0.808
Energy	0.897
Loudness	-3.712
Speechiness	0.0348
Acousticness	0.0469
Liveness	0.364
Tempo	126.026
Duration (ms)	231041
Valence	0.591
Chord	B

	4 \
Highest Charting Position	5
Number of Times Charted	1
Week of Highest Charting	2021-07-23--2021-07-30
Song Name	INDUSTRY BABY (feat. Jack Harlow)
Streams	33,948,454
Artist	Lil Nas X
Artist Followers	5473565
Song ID	27NovPIUIRrOZoCHxABJwK
Genre	['lgbtq+ hip hop', 'pop rap']
Release Date	2021-07-23
Weeks Charted	2021-07-23--2021-07-30
Popularity	96
Danceability	0.736
Energy	0.704
Loudness	-7.409
Speechiness	0.0615
Acousticness	0.0203
Liveness	0.0501
Tempo	149.995
Duration (ms)	212000
Valence	0.894

Chord	D#/Eb

	5 \
Highest Charting Position	1
Number of Times Charted	18
Week of Highest Charting	2021-05-07--2021-05-14
Song Name	MONTERO (Call Me By Your Name)
Streams	30,071,134
Artist	Lil Nas X
Artist Followers	5473565
Song ID	67BtfxlNbhBmCDR2L2l8qd
Genre	['lgbtq+ hip hop', 'pop rap']
Release Date	2021-03-31
Weeks Charted	2021-07-23--2021-07-30\n2021-07-16--2021-07-23…
Popularity	97
Danceability	0.61
Energy	0.508
Loudness	-6.682
Speechiness	0.152
Acousticness	0.297
Liveness	0.384
Tempo	178.818
Duration (ms)	137876
Valence	0.758
Chord	G#/Ab

	6 \
Highest Charting Position	3
Number of Times Charted	16
Week of Highest Charting	2021-05-14--2021-05-21
Song Name	Kiss Me More (feat. SZA)
Streams	29,356,736
Artist	Doja Cat
Artist Followers	8640063
Song ID	748mdHapucXQri7IAO8yFK
Genre	['dance pop', 'pop']
Release Date	2021-04-09
Weeks Charted	2021-07-23--2021-07-30\n2021-07-16--2021-07-23…
Popularity	94
Danceability	0.762
Energy	0.701
Loudness	-3.541
Speechiness	0.0286
Acousticness	0.235
Liveness	0.123
Tempo	110.968
Duration (ms)	208867

Valence	0.742
Chord	G#/Ab

	7 \
Highest Charting Position	2
Number of Times Charted	10
Week of Highest Charting	2021-06-18--2021-06-25
Song Name	Todo De Ti
Streams	26,951,613
Artist	Rauw Alejandro
Artist Followers	6080597
Song ID	4fSIb4hdOQ151TILNsSEaF
Genre	['puerto rican pop', 'trap latino']
Release Date	2021-05-20
Weeks Charted	2021-07-23--2021-07-30\n2021-07-16--2021-07-23…
Popularity	95
Danceability	0.78
Energy	0.718
Loudness	-3.605
Speechiness	0.0506
Acousticness	0.31
Liveness	0.0932
Tempo	127.949
Duration (ms)	199604
Valence	0.342
Chord	D#/Eb

	8 \
Highest Charting Position	3
Number of Times Charted	8
Week of Highest Charting	2021-06-18--2021-06-25
Song Name	Yonaguni
Streams	25,030,128
Artist	Bad Bunny
Artist Followers	36142273
Song ID	2JPLbjOn0wPCngEot2STUS
Genre	['latin', 'reggaeton', 'trap latino']
Release Date	2021-06-04
Weeks Charted	2021-07-23--2021-07-30\n2021-07-16--2021-07-23…
Popularity	96
Danceability	0.644
Energy	0.648
Loudness	-4.601
Speechiness	0.118
Acousticness	0.276
Liveness	0.135
Tempo	179.951

Duration (ms)			206710
Valence			0.44
Chord			C#/Db

			\
		9	
Highest Charting Position		8	
Number of Times Charted		10	
Week of Highest Charting		2021-07-02--2021-07-09	
Song Name		I WANNA BE YOUR SLAVE	
Streams		24,551,591	
Artist		Måneskin	
Artist Followers		3377762	
Song ID		4pt5fDVTg5GhEvEtlz9dKk	
Genre		['indie rock italiano', 'italian pop']	
Release Date		2021-03-19	
Weeks Charted		2021-07-23--2021-07-30\n2021-07-16--2021-07-23…	
Popularity		95	
Danceability		0.75	
Energy		0.608	
Loudness		-4.008	
Speechiness		0.0387	
Acousticness		0.00165	
Liveness		0.178	
Tempo		132.507	
Duration (ms)		173347	
Valence		0.958	
Chord		C#/Db	

			\
	…	1546	
Highest Charting Position	…	143	
Number of Times Charted	…	1	
Week of Highest Charting	…	2019-12-27--2020-01-03	
Song Name	…	JACKBOYS	
Streams	…	5,363,493	
Artist	…	JACKBOYS	
Artist Followers	…	437907	
Song ID	…	62zKJrpbLxz6InR3tGyr7o	
Genre	…	['rap', 'trap']	
Release Date	…	2019-12-27	
Weeks Charted	…	2019-12-27--2020-01-03	
Popularity	…	56	
Danceability	…	0.413	
Energy	…	0.13	
Loudness	…	-25.166	
Speechiness	…	0.0336	
Acousticness	…	0.9	
Liveness	…	0.111	

Tempo	…	123.342
Duration (ms)	…	46837
Valence	…	0.0676
Chord	…	C

	1547 \
Highest Charting Position	156
Number of Times Charted	1
Week of Highest Charting	2019-12-27--2020-01-03
Song Name	Combatchy (feat. MC Rebecca)
Streams	5,149,797
Artist	Anitta, Lexa, Luísa Sonza
Artist Followers	10741972
Song ID	2bPtwnrpFNEe8N7Q85kLHw
Genre	['funk carioca', 'funk pop', 'pagode baiano', …
Release Date	2019-11-20
Weeks Charted	2019-12-27--2020-01-03
Popularity	64
Danceability	0.826
Energy	0.73
Loudness	-3.032
Speechiness	0.0809
Acousticness	0.383
Liveness	0.0197
Tempo	150.134
Duration (ms)	157600
Valence	0.605
Chord	C#/Db

	1548 \
Highest Charting Position	178
Number of Times Charted	1
Week of Highest Charting	2019-12-27--2020-01-03
Song Name	Old Town Road
Streams	4,852,004
Artist	Lil Nas X
Artist Followers	5488666
Song ID	2YpeDb67231RjR0MgVLzsG
Genre	['lgbtq+ hip hop', 'pop rap']
Release Date	2019-06-21
Weeks Charted	2019-12-27--2020-01-03
Popularity	81
Danceability	0.878
Energy	0.619
Loudness	-5.56
Speechiness	0.102
Acousticness	0.0533

Liveness	0.113	
Tempo	136.041	
Duration (ms)	157067	
Valence	0.639	
Chord	F#/Gb	

	1549	\
Highest Charting Position	187	
Number of Times Charted	1	
Week of Highest Charting	2019-12-27--2020-01-03	
Song Name	Let Me Know (I Wonder Why Freestyle)	
Streams	4,701,532	
Artist	Juice WRLD	
Artist Followers	19102888	
Song ID	3wwo0bJvDSorOpNfzEkfXx	
Genre	['chicago rap', 'melodic rap']	
Release Date	2019-12-07	
Weeks Charted	2019-12-27--2020-01-03	
Popularity	76	
Danceability	0.635	
Energy	0.537	
Loudness	-7.895	
Speechiness	0.0832	
Acousticness	0.172	
Liveness	0.418	
Tempo	125.028	
Duration (ms)	215381	
Valence	0.383	
Chord	G	

	1550	\
Highest Charting Position	190	
Number of Times Charted	1	
Week of Highest Charting	2019-12-27--2020-01-03	
Song Name	Ne reviens pas	
Streams	4,676,857	
Artist	Gradur, Heuss L'enfoiré	
Artist Followers	1390813	
Song ID	4TnFANpjVwVKWzkxNzIyFH	
Genre	['francoton', 'french hip hop', 'pop urbaine',…	
Release Date	2019-11-29	
Weeks Charted	2019-12-27--2020-01-03	
Popularity	62	
Danceability	0.932	
Energy	0.778	
Loudness	-3.384	
Speechiness	0.0638	

Acousticness	0.212
Liveness	0.168
Tempo	124.996
Duration (ms)	188613
Valence	0.933
Chord	A#/Bb

	1551 \
Highest Charting Position	195
Number of Times Charted	1
Week of Highest Charting	2019-12-27--2020-01-03
Song Name	New Rules
Streams	4,630,675
Artist	Dua Lipa
Artist Followers	27167675
Song ID	2ekn2ttSfGqwhhate0LSR0
Genre	['dance pop', 'pop', 'uk pop']
Release Date	2017-06-02
Weeks Charted	2019-12-27--2020-01-03
Popularity	79
Danceability	0.762
Energy	0.7
Loudness	-6.021
Speechiness	0.0694
Acousticness	0.00261
Liveness	0.153
Tempo	116.073
Duration (ms)	209320
Valence	0.608
Chord	A

	1552 \
Highest Charting Position	196
Number of Times Charted	1
Week of Highest Charting	2019-12-27--2020-01-03
Song Name	Cheirosa - Ao Vivo
Streams	4,623,030
Artist	Jorge & Mateus
Artist Followers	15019109
Song ID	2PWjKmjyTZeDpmOUa3a5da
Genre	['sertanejo', 'sertanejo universitario']
Release Date	2019-10-11
Weeks Charted	2019-12-27--2020-01-03
Popularity	66
Danceability	0.528
Energy	0.87
Loudness	-3.123

Speechiness	0.0851
Acousticness	0.24
Liveness	0.333
Tempo	152.37
Duration (ms)	181930
Valence	0.714
Chord	B

	1553 \
Highest Charting Position	197
Number of Times Charted	1
Week of Highest Charting	2019-12-27--2020-01-03
Song Name	Havana (feat. Young Thug)
Streams	4,620,876
Artist	Camila Cabello
Artist Followers	22698747
Song ID	1rfofaqEpACxVEHIZBJe6W
Genre	['dance pop', 'electropop', 'pop', 'post-teen …
Release Date	2018-01-12
Weeks Charted	2019-12-27--2020-01-03
Popularity	81
Danceability	0.765
Energy	0.523
Loudness	-4.333
Speechiness	0.03
Acousticness	0.184
Liveness	0.132
Tempo	104.988
Duration (ms)	217307
Valence	0.394
Chord	D

	1554 \
Highest Charting Position	198
Number of Times Charted	1
Week of Highest Charting	2019-12-27--2020-01-03
Song Name	Surtada - Remix Brega Funk
Streams	4,607,385
Artist	Dadá Boladão, Tati Zaqui, OIK
Artist Followers	208630
Song ID	5F8ffc8KWKNawllr5WsW0r
Genre	['brega funk', 'funk carioca']
Release Date	2019-09-25
Weeks Charted	2019-12-27--2020-01-03
Popularity	60
Danceability	0.832
Energy	0.55

```
Loudness                                                -7.026
Speechiness                                             0.0587
Acousticness                                             0.249
Liveness                                                 0.182
Tempo                                                  154.064
Duration (ms)                                           152784
Valence                                                  0.881
Chord                                                        F

                                                          1555
Highest Charting Position                                  199
Number of Times Charted                                      1
Week of Highest Charting                  2019-12-27--2020-01-03
Song Name                 Lover (Remix) [feat. Shawn Mendes]
Streams                                               4,595,450
Artist                                              Taylor Swift
Artist Followers                                       42227614
Song ID                                  3i9UVldZOE0aD0JnyfAZZ0
Genre                                 ['pop', 'post-teen pop']
Release Date                                        2019-11-13
Weeks Charted                             2019-12-27--2020-01-03
Popularity                                                   70
Danceability                                              0.448
Energy                                                    0.603
Loudness                                                 -7.176
Speechiness                                               0.064
Acousticness                                              0.433
Liveness                                                 0.0862
Tempo                                                   205.272
Duration (ms)                                            221307
Valence                                                   0.422
Chord                                                         G

[22 rows x 1556 columns]
```

4.3 Convert object columns with numbers to float64

[610]:
```
# List of columns to convert
columns_to_convert = ['Artist Followers', 'Streams','Popularity',␣
 ↪'Danceability', 'Energy', 'Loudness',
                      'Speechiness', 'Acousticness', 'Liveness', 'Tempo',␣
 ↪'Duration (ms)', 'Valence']

# Convert columns to numeric
for column in columns_to_convert:
    df_1[column] = pd.to_numeric(df_1[column], errors='coerce')
```

[611]:
```
df_1.dtypes
```

[611]:
```
Index                          int64
Highest Charting Position      int64
Number of Times Charted        int64
Week of Highest Charting      object
Song Name                     object
Streams                      float64
Artist                        object
Artist Followers             float64
Song ID                       object
Genre                         object
Release Date                  object
Weeks Charted                 object
Popularity                   float64
Danceability                 float64
Energy                       float64
Loudness                     float64
Speechiness                  float64
Acousticness                 float64
Liveness                     float64
Tempo                        float64
Duration (ms)                float64
Valence                      float64
Chord                         object
dtype: object
```

5 Data Cleaning Continued: Prepare DataFrame for Modeling and Training

[612]:
```
df_1 = df_1.drop("Index", axis = 1)
```

[613]:
```
df_1
```

[613]:
```
      Highest Charting Position  Number of Times Charted  \
0                             1                        8
1                             2                        3
2                             1                       11
3                             3                        5
4                             5                        1
…                           …                        …
1551                        195                        1
1552                        196                        1
1553                        197                        1
1554                        198                        1
1555                        199                        1
```

```
     Week of Highest Charting                          Song Name  Streams  \
0      2021-07-23--2021-07-30                            Beggin'      NaN
1      2021-07-23--2021-07-30          STAY (with Justin Bieber)      NaN
2      2021-06-25--2021-07-02                           good 4 u      NaN
3      2021-07-02--2021-07-09                         Bad Habits      NaN
4      2021-07-23--2021-07-30  INDUSTRY BABY (feat. Jack Harlow)      NaN
…                         …                                  …      …
1551   2019-12-27--2020-01-03                        New Rules      NaN
1552   2019-12-27--2020-01-03               Cheirosa - Ao Vivo      NaN
1553   2019-12-27--2020-01-03         Havana (feat. Young Thug)      NaN
1554   2019-12-27--2020-01-03        Surtada - Remix Brega Funk      NaN
1555   2019-12-27--2020-01-03  Lover (Remix) [feat. Shawn Mendes]      NaN

                            Artist  Artist Followers                Song ID  \
0                          Måneskin         3377762.0  3Wrjm47oTz2sjIgck11l5e
1                      The Kid LAROI         2230022.0  5HCyWlXZPP0y6Gqq8TgA20
2                    Olivia Rodrigo         6266514.0  4ZtFanR9U6ndgddUvNcjcG
3                        Ed Sheeran        83293380.0  6PQ88X9TkUIAUIZJHW2upE
4                       Lil Nas X          5473565.0  27NovPIUIRrOZoCHxABJwK
…                               …                …                       …
1551                     Dua Lipa        27167675.0  2ekn2ttSfGqwhhate0LSR0
1552                Jorge & Mateus        15019109.0  2PWjKmjyTZeDpmOUa3a5da
1553                Camila Cabello        22698747.0  1rfofaqEpACxVEHIZBJe6W
1554  Dadá Boladão, Tati Zaqui, OIK          208630.0  5F8ffc8KWKNawllr5WsW0r
1555                   Taylor Swift        42227614.0  3i9UVldZOE0aD0JnyfAZZ0

                                          Genre Release Date  …  \
0               ['indie rock italiano', 'italian pop']   2017-12-08  …
1                                ['australian hip hop']   2021-07-09  …
2                                            ['pop']   2021-05-21  …
3                                  ['pop', 'uk pop']   2021-06-25  …
4                       ['lgbtq+ hip hop', 'pop rap']   2021-07-23  …
…                                                 …          …  …
1551                    ['dance pop', 'pop', 'uk pop']   2017-06-02  …
1552          ['sertanejo', 'sertanejo universitario']   2019-10-11  …
1553  ['dance pop', 'electropop', 'pop', 'post-teen …   2018-01-12  …
1554                    ['brega funk', 'funk carioca']   2019-09-25  …
1555                         ['pop', 'post-teen pop']   2019-11-13  …

     Danceability  Energy  Loudness  Speechiness  Acousticness  Liveness  \
0           0.714   0.800    -4.808       0.0504       0.12700    0.3590
1           0.591   0.764    -5.484       0.0483       0.03830    0.1030
2           0.563   0.664    -5.044       0.1540       0.33500    0.0849
3           0.808   0.897    -3.712       0.0348       0.04690    0.3640
4           0.736   0.704    -7.409       0.0615       0.02030    0.0501
…               …       …         …            …             …         …
```

```
1551        0.762   0.700    -6.021       0.0694       0.00261    0.1530
1552        0.528   0.870    -3.123       0.0851       0.24000    0.3330
1553        0.765   0.523    -4.333       0.0300       0.18400    0.1320
1554        0.832   0.550    -7.026       0.0587       0.24900    0.1820
1555        0.448   0.603    -7.176       0.0640       0.43300    0.0862
```

	Tempo	Duration (ms)	Valence	Chord
0	134.002	211560.0	0.589	B
1	169.928	141806.0	0.478	C#/Db
2	166.928	178147.0	0.688	A
3	126.026	231041.0	0.591	B
4	149.995	212000.0	0.894	D#/Eb
…	…	…	…	…
1551	116.073	209320.0	0.608	A
1552	152.370	181930.0	0.714	B
1553	104.988	217307.0	0.394	D
1554	154.064	152784.0	0.881	F
1555	205.272	221307.0	0.422	G

```
[1556 rows x 22 columns]
```

[614]:
```
df_clean_2 = df_1.copy()
```

5.1 Identify Object Columns & Drop them

[615]:
```
object_columns = df_clean_2.select_dtypes(include=['object']).columns
df_clean_2 = df_clean_2.drop(columns=object_columns)
```

[616]:
```
df_clean_2.isnull().sum()
```

[616]:
```
Highest Charting Position      0
Number of Times Charted        0
Streams                     1556
Artist Followers              11
Popularity                    11
Danceability                  11
Energy                        11
Loudness                      11
Speechiness                   11
Acousticness                  11
Liveness                      11
Tempo                         11
Duration (ms)                 11
Valence                       11
dtype: int64
```

[617]:
```
df_clean_2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1556 entries, 0 to 1555
Data columns (total 14 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Highest Charting Position  1556 non-null   int64
 1   Number of Times Charted    1556 non-null   int64
 2   Streams                    0 non-null      float64
 3   Artist Followers           1545 non-null   float64
 4   Popularity                 1545 non-null   float64
 5   Danceability               1545 non-null   float64
 6   Energy                     1545 non-null   float64
 7   Loudness                   1545 non-null   float64
 8   Speechiness                1545 non-null   float64
 9   Acousticness               1545 non-null   float64
 10  Liveness                   1545 non-null   float64
 11  Tempo                      1545 non-null   float64
 12  Duration (ms)              1545 non-null   float64
 13  Valence                    1545 non-null   float64
dtypes: float64(12), int64(2)
memory usage: 170.3 KB
```

5.2 Drop Streams Column (essentially empty)

[618]:
```
df_clean_2.drop('Streams', axis = 1, inplace = True)
```

[619]:
```
df_clean_2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1556 entries, 0 to 1555
Data columns (total 13 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Highest Charting Position  1556 non-null   int64
 1   Number of Times Charted    1556 non-null   int64
 2   Artist Followers           1545 non-null   float64
 3   Popularity                 1545 non-null   float64
 4   Danceability               1545 non-null   float64
 5   Energy                     1545 non-null   float64
 6   Loudness                   1545 non-null   float64
 7   Speechiness                1545 non-null   float64
 8   Acousticness               1545 non-null   float64
 9   Liveness                   1545 non-null   float64
 10  Tempo                      1545 non-null   float64
 11  Duration (ms)              1545 non-null   float64
 12  Valence                    1545 non-null   float64
dtypes: float64(11), int64(2)
memory usage: 158.2 KB
```

5.3 Get means and replace null values with mean per column

[620]:
```
df_clean_2.isna().sum()
```

[620]:
```
Highest Charting Position     0
Number of Times Charted       0
Artist Followers             11
Popularity                   11
Danceability                 11
Energy                       11
Loudness                     11
Speechiness                  11
Acousticness                 11
Liveness                     11
Tempo                        11
Duration (ms)                11
Valence                      11
dtype: int64
```

[621]:
```
null_columns = df_clean_2.columns[df_clean_2.isnull().any()].tolist()
print("Columns with null values:")
null_columns
```

```
Columns with null values:
```

[621]:
```
['Artist Followers',
 'Popularity',
 'Danceability',
 'Energy',
 'Loudness',
 'Speechiness',
 'Acousticness',
 'Liveness',
 'Tempo',
 'Duration (ms)',
 'Valence']
```

[622]:
```
for col in null_columns:
    #Calculate the mean, exluding NaN values
    mean= df_clean_2[col].mean(skipna=True)

    #replace NaNs with the mean per column
    df_clean_2[col] = df_clean_2[col].fillna(mean)
```

[623]:
```
print("\nNull value count after replacement:")
print(df_clean_2.isnull().sum())
```

```
Null value count after replacement:
```

```
Highest Charting Position    0
Number of Times Charted      0
Artist Followers             0
Popularity                   0
Danceability                 0
Energy                       0
Loudness                     0
Speechiness                  0
Acousticness                 0
Liveness                     0
Tempo                        0
Duration (ms)                0
Valence                      0
dtype: int64
```

[624]:
```
df_clean_2.dtypes
```

[624]:
```
Highest Charting Position      int64
Number of Times Charted        int64
Artist Followers             float64
Popularity                   float64
Danceability                 float64
Energy                       float64
Loudness                     float64
Speechiness                  float64
Acousticness                 float64
Liveness                     float64
Tempo                        float64
Duration (ms)                float64
Valence                      float64
dtype: object
```

5.4 Drop columns that have no relation to target = "Popularity"

[625]:
```
df_clean_2.drop('Highest Charting Position', axis = 1, inplace = True)
```

[626]:
```
df_clean_2.drop('Number of Times Charted', axis = 1, inplace = True)
```

[627]:
```
df_clean_2.drop('Artist Followers', axis = 1, inplace = True)
```

[628]:
```
df_scaling = df_clean_2.copy()
```

6 Data Scaling

6.1 Data Scaling (standard scaler)

6.1.1 Setup standard scaled training and testing data

[629]:
```
df_3_std = df_scaling.copy()
```

[630]:
```
x1 = df_3_std.drop(['Popularity'], axis=1)
y1 = df_3_std['Popularity']

X_train_1, X_test_1, y_train_1, y_test_1 = train_test_split(x1, y1, test_size=0.
↪2)
```

[631]:
```
scaler = StandardScaler()
X_train_std = scaler.fit_transform(X_train_1)
X_test_std = scaler.transform(X_test_1)
```

[632]:
```
print("Before scaling:")
print(X_train_1.describe())

print("\nAfter scaling:")
print(pd.DataFrame(X_train_std).describe())
```

```
Before scaling:
       Danceability       Energy     Loudness  Speechiness  Acousticness  \
count   1244.000000  1244.000000  1244.000000  1244.000000   1244.000000
mean       0.693330     0.634184    -6.350928     0.125646      0.247434
std        0.141247     0.160718     2.517666     0.112861      0.249600
min        0.150000     0.054000   -25.166000     0.023200      0.000025
25%        0.611750     0.534750    -7.470500     0.046075      0.048000
50%        0.710000     0.642000    -6.022000     0.077700      0.161000
75%        0.798000     0.749000    -4.732500     0.166250      0.374000
max        0.980000     0.970000    -0.515000     0.884000      0.994000

          Liveness        Tempo  Duration (ms)      Valence
count  1244.000000  1244.000000    1244.000000  1244.000000
mean      0.182821   122.890179  198631.054650     0.516792
std       0.147315    29.589980   48720.311502     0.225130
min       0.019700    46.718000   30133.000000     0.032000
25%       0.096225    97.083000  169354.750000     0.345750
50%       0.124000   122.801512  193544.000000     0.514852
75%       0.217000   142.980000  219971.000000     0.685500
max       0.962000   205.272000  588139.000000     0.979000

After scaling:
                  0             1             2             3             4  \
count  1.244000e+03  1.244000e+03  1.244000e+03  1.244000e+03  1.244000e+03
mean   3.541290e-16  1.342263e-16  7.425286e-17 -1.699248e-16  1.599292e-16
```

```
std    1.000402e+00  1.000402e+00  1.000402e+00  1.000402e+00  1.000402e+00
min   -3.848223e+00 -3.611403e+00 -7.476225e+00 -9.080810e-01 -9.916188e-01
25%   -5.778009e-01 -6.189333e-01 -4.448654e-01 -7.053169e-01 -7.993359e-01
50%    1.180712e-01  4.865362e-02  1.307005e-01 -4.249928e-01 -3.464293e-01
75%    7.413460e-01  7.146844e-01  6.430872e-01  3.599148e-01  5.072797e-01
max    2.030392e+00  2.090318e+00  2.318923e+00  6.722054e+00  2.992254e+00

                5             6             7             8
count  1.244000e+03  1.244000e+03  1.244000e+03  1.244000e+03
mean   7.425286e-17 -3.427055e-16  2.855879e-16 -7.996462e-17
std    1.000402e+00  1.000402e+00  1.000402e+00  1.000402e+00
min   -1.107733e+00 -2.575291e+00 -3.459867e+00 -2.154258e+00
25%   -5.880606e-01 -8.725102e-01 -6.011472e-01 -7.600559e-01
50%   -3.994438e-01 -2.997754e-03 -1.044554e-01 -8.622533e-03
75%    2.321083e-01  6.792130e-01  4.381854e-01  7.496816e-01
max    5.291316e+00  2.785232e+00  7.997991e+00  2.053899e+00
```

[633]:
```
print("Mean:", X_train_std.mean(axis=0))
print("Std:", X_train_std.std(axis=0))
```

```
Mean: [ 3.54129016e-16  1.34226321e-16  7.42528582e-17 -1.69924810e-16
  1.59929233e-16  7.42528582e-17 -3.42705500e-16  2.85587916e-16
 -7.99646166e-17]
Std: [1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

6.2 Data Scaling Continued (min-max scaler)

[634]:
```
df_3_mm = df_scaling.copy()
```

[635]:
```
x2 = df_3_mm.drop(['Popularity'], axis=1)
y2 = df_3_mm['Popularity']

X_train_2, X_test_2, y_train_2, y_test_2 = train_test_split(x2, y2, test_size=0.
 ↪2)
```

6.2.1 Setup mm scaled training and testing data

[636]:
```
scaler = MinMaxScaler()
X_train_mm = scaler.fit_transform(X_train_2)
X_test_mm = scaler.transform(X_test_2)
```

[637]:
```
print("Before scaling:")
print(X_train_2.describe())

print("\nAfter scaling:")
print(pd.DataFrame(X_train_mm).describe())
```

```
Before scaling:
       Danceability       Energy     Loudness  Speechiness  Acousticness  \
```

```
count   1244.000000  1244.000000  1244.000000  1244.000000   1244.000000
mean       0.691402     0.631728    -6.358311     0.124415      0.250728
std        0.140560     0.161583     2.545385     0.110221      0.245910
min        0.184000     0.103000   -25.166000     0.023200      0.000178
25%        0.600000     0.528750    -7.493000     0.046000      0.051075
50%        0.708500     0.641000    -6.013500     0.078200      0.168000
75%        0.796000     0.748000    -4.725000     0.164000      0.391000
max        0.980000     0.970000     1.509000     0.884000      0.994000

          Liveness        Tempo  Duration (ms)      Valence
count  1244.000000  1244.000000    1244.000000  1244.000000
mean      0.182661   123.158875  197730.761027     0.517858
std       0.148033    29.536084   46750.187382     0.226690
min       0.019700    46.718000   30133.000000     0.032000
25%       0.096450    97.987000  169117.500000     0.346000
50%       0.123000   122.811023  193573.000000     0.514704
75%       0.217000   142.994250  217773.000000     0.694000
max       0.962000   205.272000  588139.000000     0.979000

After scaling:
                0            1            2            3            4  \
count  1244.000000  1244.000000  1244.000000  1244.000000  1244.000000
mean      0.637440     0.609836     0.705068     0.117583     0.252108
std       0.176583     0.186370     0.095422     0.128045     0.247438
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.522613     0.491061     0.662530     0.026487     0.051213
50%       0.658920     0.620531     0.717994     0.063894     0.168865
75%       0.768844     0.743945     0.766298     0.163569     0.393252
max       1.000000     1.000000     1.000000     1.000000     1.000000

                5            6            7            8
count  1244.000000  1244.000000  1244.000000  1244.000000
mean      0.172940     0.482113     0.300351     0.513050
std       0.157098     0.186284     0.083781     0.239377
min       0.000000     0.000000     0.000000     0.000000
25%       0.081450     0.323354     0.249073     0.331573
50%       0.109625     0.479919     0.292900     0.509719
75%       0.209381     0.607214     0.336269     0.699050
max       1.000000     1.000000     1.000000     1.000000
```

[638]:
```
print("Mean:", X_train_mm.mean(axis=0))
print("Std:", X_train_mm.std(axis=0))
```

```
Mean: [0.63743959 0.60983606 0.70506799 0.1175827  0.25210771 0.1729399
 0.48211256 0.30035118 0.51304952]
Std: [0.17651218 0.18629487 0.09538376 0.12799348 0.24733896 0.15703468
 0.18620918 0.08374712 0.23928097]
```

7 Model Selection and Training

7.1 Models: STD Scaler

7.1.1 Linear Regression std scaler

[639]:
```
lr_model = LinearRegression()
lr_model.fit(X_train_std, y_train_1)
y_pred_lr = lr_model.predict(X_test_std)
print('Linear Regression:')
print(f"RMSE: {np.sqrt(mean_squared_error(y_test_1,y_pred_lr)) :.2f}%")
print(f"R2 Score: {r2_score(y_test_1,y_pred_lr):.2f}")
```

```
Linear Regression:
RMSE: 15.72%
R2 Score: -0.02
```

Cross Validation Score for Linear Regression

[656]:
```
lr_model = LinearRegression()
cv_scores = cross_val_score(lr_model, X_train_1, y_train_1, cv=5,␣
 ↪scoring='neg_mean_squared_error')
rmse = np.sqrt(-cv_scores.mean())
print(f"Cross-validated RMSE: {rmse:.2f}")
```

```
Cross-validated RMSE: 15.54
```

7.1.2 Decision Tree Model std scaler

[670]:
```
dt_model = DecisionTreeRegressor()
dt_model.fit(X_train_std, y_train_1)
y_pred_dt = dt_model.predict(X_test_std)

print("\nDecision Tree:")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test_1, y_pred_dt)) :.2f}%")
print(f"R2 Score: {r2_score(y_test_1, y_pred_dt):.2f}")
```

```
Decision Tree:
RMSE: 23.84%
R2 Score: -1.34
```

Cross Validation Score for Decision Tree

[669]:
```
dt_model = DecisionTreeRegressor()
cv_scores = cross_val_score(dt_model, X_train_std, y_train_1, cv=5,␣
 ↪scoring='neg_mean_squared_error')
rmse = np.sqrt(-cv_scores.mean())
print(f"Cross-validated RMSE: {rmse:.2f}")
```

```
Cross-validated RMSE: 22.90
```

Feature Importance for Decision Tree

[658]:
```
dt_model.fit(X_train_std, y_train_1)

feature_importances = dt_model.feature_importances_
feature_names = X_train_1.columns
feature_importance_df = pd.DataFrame({'feature': feature_names, 'importance':␣
 ↪feature_importances})
feature_importance_df = feature_importance_df.sort_values('importance',␣
 ↪ascending=False)
print(feature_importance_df)
```

```
         feature  importance
7  Duration (ms)    0.159252
2       Loudness    0.129797
5       Liveness    0.125686
3    Speechiness    0.119065
0   Danceability    0.103464
8        Valence    0.102468
4   Acousticness    0.101686
6          Tempo    0.085153
1         Energy    0.073431
```

7.1.3 Random Forest Model std scaler

[641]:
```
rf_model = RandomForestRegressor(n_estimators=100)
rf_model.fit(X_train_std, y_train_1)
y_pred_rf = rf_model.predict(X_test_std)

print("\nRandom Forest:")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test_1, y_pred_rf)) :.2f}%")
print(f"R2 Score: {r2_score(y_test_1, y_pred_rf):.2f}")
```

```
Random Forest:
RMSE: 15.97%
R2 Score: -0.05
```

Cross Validation Score for Random Forest

[649]:
```
rf_model = RandomForestRegressor(n_estimators=100)
cv_scores = cross_val_score(rf_model, X_train_1, y_train_1, cv=5,␣
 ↪scoring='neg_mean_squared_error')
rmse = np.sqrt(-cv_scores.mean())
print(f"Cross-validated RMSE: {rmse:.2f}")
```

```
Cross-validated RMSE: 16.16
```

Feature Importance for Random Forest

[ ]:
```
rf_model.fit(X_train_std, y_train_1)
```

```
3    Speechiness    0.112036
7  Duration (ms)    0.110194
8        Valence    0.102458
1         Energy    0.102449
0   Danceability    0.093869
4   Acousticness    0.093788
Cross-validated RMSE: 22.90
Cross-validated RMSE: 15.54
         feature  importance
7  Duration (ms)    0.159252
2       Loudness    0.129797
5       Liveness    0.125686
3    Speechiness    0.119065
0   Danceability    0.103464
8        Valence    0.102468
4   Acousticness    0.101686
6          Tempo    0.085153
1         Energy    0.073431
Feature Importance for XGBoost
         feature  importance
2       Loudness    0.154894
5       Liveness    0.123070
7  Duration (ms)    0.111880
3    Speechiness    0.109152
1         Energy    0.108955
6          Tempo    0.108737
8        Valence    0.100562
4   Acousticness    0.092776
0   Danceability    0.089975
Cross Validation Score for XGBoost
Cross-validated RMSE: 16.10
Cross-validated RMSE: 15.92
Cross Validation Score for Decision Tree mm
Cross-validated RMSE: 23.36
Feature Importance for Random Forest mm
Feature Importance for Decision Tree mm
         feature  importance
2       Loudness    0.200263
5       Liveness    0.146557
4   Acousticness    0.115964
6          Tempo    0.110931
3    Speechiness    0.109839
7  Duration (ms)    0.093054
8        Valence    0.089971
0   Danceability    0.069099
```

```
1         Energy    0.064322
          feature  importance
2       Loudness    0.151351
5       Liveness    0.127799
7  Duration (ms)    0.110652
1         Energy    0.105845
3    Speechiness    0.104500
4   Acousticness    0.103343
8        Valence    0.101876
6          Tempo    0.101761
0   Danceability    0.092874
Cross Validation Score for XGBoost mm
Cross-validated RMSE: 16.39
Feature Importance for XGBoost mm
          feature  importance
2       Loudness    0.155671
5       Liveness    0.139164
6          Tempo    0.107859
3    Speechiness    0.107299
7  Duration (ms)    0.106620
8        Valence    0.102282
4   Acousticness    0.098379
1         Energy    0.092395
feature_importances = rf_model.feature_importances_
feature_names = X_train_1.columns
feature_importance_df = pd.DataFrame({'feature': feature_names, 'importance':␣
 ↪feature_importances})
feature_importance_df = feature_importance_df.sort_values('importance',␣
 ↪ascending=False)
print(feature_importance_df)
```

```
          feature  importance
2       Loudness    0.157219
5       Liveness    0.114515
6          Tempo    0.113472
3    Speechiness    0.112036
7  Duration (ms)    0.110194
8        Valence    0.102458
1         Energy    0.102449
0   Danceability    0.093869
4   Acousticness    0.093788
```

7.1.4 XGBoost Model std scaler

[642]:
```
xgb_model = xgb.XGBRegressor(n_estimators=100)
xgb_model.fit(X_train_std, y_train_1)
y_pred_xgb = xgb_model.predict(X_test_std)
```

```
print("\nXGBoost:")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test_1, y_pred_xgb)) :.2f}%")
print(f"R2 Score: {r2_score(y_test_1, y_pred_xgb):.2f}")
```

```
XGBoost:
RMSE: 18.02%
R2 Score: -0.33
```

Cross Validation Score for XGBoost

[668]:
```
xgb_model = RandomForestRegressor(n_estimators=100)
cv_scores = cross_val_score(rf_model, X_train_std, y_train_1, cv=5,␣
 ↪scoring='neg_mean_squared_error')
rmse = np.sqrt(-cv_scores.mean())
print(f"Cross-validated RMSE: {rmse:.2f}")
```

```
Cross-validated RMSE: 16.10
```

Feature Importance for XGBoost

[667]:
```
xgb_model.fit(X_train_std, y_train_1)

feature_importances = xgb_model.feature_importances_
feature_importance_df = pd.DataFrame({'feature': feature_names, 'importance':␣
 ↪feature_importances})
feature_importance_df = feature_importance_df.sort_values('importance',␣
 ↪ascending=False)
print(feature_importance_df)
```

```
         feature  importance
2       Loudness    0.154894
5       Liveness    0.123070
7  Duration (ms)    0.111880
3    Speechiness    0.109152
1         Energy    0.108955
6          Tempo    0.108737
8        Valence    0.100562
4   Acousticness    0.092776
0   Danceability    0.089975
```

7.2 Models: MM Scaler

7.2.1 Linear Regression mm scaler

[643]:
```
lr_model = LinearRegression()
lr_model.fit(X_train_mm, y_train_2)
y_pred_lr = lr_model.predict(X_test_mm)
print('Linear Regression:')
```

```
print(f"RMSE: {np.sqrt(mean_squared_error(y_test_2,y_pred_lr)) :.2f}%")
print(f"R2 Score: {r2_score(y_test_2,y_pred_lr):.2f}")
```

```
Linear Regression:
RMSE: 14.28%
R2 Score: 0.01
```

Cross Validation Score for Linear Regression mm

[ ]:
```
lr_model = LinearRegression()
cv_scores = cross_val_score(lr_model, X_train_mm, y_train_2, cv=5,␣
 ↪scoring='neg_mean_squared_error')
rmse = np.sqrt(-cv_scores.mean())
print(f"Cross-validated RMSE: {rmse:.2f}")re': feature_n
```

```
Cross-validated RMSE: 15.92
```

7.2.2 Decision Tree mm scaler

[644]:
```
dt_model = DecisionTreeRegressor()
dt_model.fit(X_train_mm, y_train_2)
y_pred_dt = dt_model.predict(X_test_mm)

print("\nDecision Tree:")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test_2, y_pred_dt)) :.2f}%")
print(f"R2 Score: {r2_score(y_test_2, y_pred_dt):.2f}")
```

```
Decision Tree:
RMSE: 22.82%
R2 Score: -1.52
```

Cross Validation Score for Decision Tree mm

[674]:
```
cv_scores = cross_val_score(dt_model, X_train_mm, y_train_2, cv=5,␣
 ↪scoring='neg_mean_squared_error')
rmse = np.sqrt(-cv_scores.mean())
print(f"Cross-validated RMSE: {rmse:.2f}")
```

```
Cross-validated RMSE: 23.36
```

Feature Importance for Decision Tree mm

[681]:
```
dt_model.fit(X_train_mm, y_train_2)

feature_importances = dt_model.feature_importances_
feature_names = X_train_2.columns
feature_importance_df = pd.DataFrame({'feature': feature_names, 'importance':␣
 ↪feature_importances})
feature_importance_df = feature_importance_df.sort_values('importance',␣
 ↪ascending=False)
```

```
print(feature_importance_df)
```

```
         feature  importance
2       Loudness    0.200263
5       Liveness    0.146557
4   Acousticness    0.115964
6          Tempo    0.110931
3    Speechiness    0.109839
7  Duration (ms)    0.093054
8        Valence    0.089971
0   Danceability    0.069099
1         Energy    0.064322
```

7.2.3 Random Forest mm scaler

[ ]:
```
rf_model = RandomForestRegressor(n_estimators=100)re': feature_n
rf_model.fit(X_train_mm, y_train_2)
y_pred_rf = rf_model.predict(X_test_mm)

print("\nRandom Forest:")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test_2, y_pred_rf)) :.2f}%")
print(f"R2 Score: {r2_score(y_test_2, y_pred_rf):.2f}")
```

```
Random Forest:
RMSE: 15.40%
R2 Score: -0.15
```

Cross Validation Score Random Forest mm

[648]:
```
rf_model = RandomForestRegressor(n_estimators=100)
cv_scores = cross_val_score(rf_model, X_train_2, y_train_2, cv=5,␣
 ↪scoring='neg_mean_squared_error')
rmse = np.sqrt(-cv_scores.mean())
print(f"Cross-validated RMSE: {rmse:.2f}")
```

```
Cross-validated RMSE: 16.38
```

Feature Importance for Random Forest mm

[677]:
```
rf_model.fit(X_train_mm, y_train_2)

feature_importances = rf_model.feature_importances_
feature_names = X_train_2.columns
feature_importance_df = pd.DataFrame({'feature': feature_names, 'importance':␣
 ↪feature_importances})
feature_importance_df = feature_importance_df.sort_values('importance',␣
 ↪ascending=False)
print(feature_importance_df)
```

```
          feature  importance
2        Loudness    0.151351
5        Liveness    0.127799
7  Duration (ms)     0.110652
1         Energy     0.105845
3     Speechiness    0.104500
4    Acousticness    0.103343
8         Valence    0.101876
6           Tempo    0.101761
0    Danceability    0.092874
```

7.2.4 XGBoost mm scaler

[646]:
```
xgb_model = xgb.XGBRegressor(n_estimators=100)
xgb_model.fit(X_train_mm, y_train_2)
y_pred_xgb = xgb_model.predict(X_test_mm)

print("\nXGBoost:")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test_2, y_pred_xgb)) :.2f}%")
print(f"R2 Score: {r2_score(y_test_2, y_pred_xgb):.2f}")
```

```
XGBoost:
RMSE: 17.16%
R2 Score: -0.43
```

Cross Validation Score for XGBoost mm

[678]:
```
xgb_model = RandomForestRegressor(n_estimators=100)
cv_scores = cross_val_score(rf_model, X_train_2, y_train_2, cv=5,␣
 ↪scoring='neg_mean_squared_error')
rmse = np.sqrt(-cv_scores.mean())
print(f"Cross-validated RMSE: {rmse:.2f}")
```

```
Cross-validated RMSE: 16.39
```

Feature Importance for XGBoost mm

[679]:
```
xgb_model.fit(X_train_mm, y_train_2)

feature_importances = xgb_model.feature_importances_
feature_names = X_train_2.columns
feature_importance_df = pd.DataFrame({'feature': feature_names, 'importance':␣
 ↪feature_importances})
feature_importance_df = feature_importance_df.sort_values('importance',␣
 ↪ascending=False)
print(feature_importance_df)
```

```
          feature  importance
2        Loudness    0.155671
5        Liveness    0.139164
```

```
6          Tempo    0.107859
3    Speechiness    0.107299
7  Duration (ms)    0.106620
8        Valence    0.102282
4   Acousticness    0.098379
1         Energy    0.092395
0   Danceability    0.090330
```