Project_4:Music_Popularity_Prediction

November 5, 2024

1 Project 4: Music Popularity Prediction

By: Robert S Balch

1.1 Hypothesis:

The popularity of a song on Spotify's Top 200 Weekly (Global) charts in 2020 & 2021 is likely influenced by a combination of audio features, artist popularity, and chart performance metrics. Specifically:

1. Audio Features:
   - Loudness and Energy are likely to be strong predictors of popularity, as more energetic and louder songs tend to perform better on charts.
   - Danceability and Valence (positiveness) may also be important, as upbeat and positive songs often appeal to a wider audience.
   - Tempo could be a factor, with faster-paced songs potentially being more popular in certain genres.
2. Artist Popularity:
   - The number of artist followers is likely to be a significant predictor, as more popular artists tend to have more popular songs.
3. Chart Performance Metrics:
   - Highest Charting Position and Number of Times Charted are likely to be strong indicators of overall popularity.
4. Genre:
   - Certain genres (e.g., pop, hip-hop) may be more represented in the top charts, potentially influencing popularity.
5. Song Characteristics:
   - Duration might play a role, with shorter songs potentially being more popular in recent years.
6. Release Timing:
   - The release date of the song could influence its popularity, with songs released earlier in the year potentially having more time to accumulate popularity.
7. Feature Interactions:
   - The interaction between audio features and artist popularity could be important. For example, a highly energetic song by a popular artist might be more likely to be popular than a similar song by a less known artist.
8. Cultural and Temporal Factors:
   - The dataset spans 2020 & 2021, which includes the COVID-19 pandemic period. This might have influenced listening habits and song popularity.

The data. A chosen data set is provided by DDC Data Science

2 Imports

[296]:
```
import sys
print(sys.executable)
```

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

[297]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
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
```

[298]:
```
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
```

[299]:
```
%%capture
url = "https://ddc-datascience.s3.amazonaws.com/Projects/Project.4-Spotify/Data/
↪Spotify.csv"
!curl -s -I {url}
```

3 Data Exploration

[300]:
```
df_1 = pd.read_csv(url).copy()
```

3.1 Head

[301]:
```
df_1.head()
```

[301]:
```
   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

3.3 Shape

[302]:
```
df_1.shape
```

[302]:
```
(1556, 23)
```

3.4 columns

[303]:
```
df_1.columns
```

[303]:
```
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

[304]:
```
df_1.dtypes
```

[304]:
```
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

[305]:
```
df_1.describe()
```

[305]:

	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

[306]:
```
df_1.isnull().sum()
```

[306]:
```
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

[307]:
```
df_1.isna().sum()
```

[307]:
```
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

[308]:
```
df_1.count('rows').unique().sum()
```

[308]:
```
np.int64(1556)
```

[309]:
```
df_1.count('columns')
```

[309]:
```
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

[310]:
```
df_1.sort_values(by = ['Popularity'], ascending = False).head(10)
```

[310]:
```
    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

[311]:
```
df_cleaning = df_1.copy()
df_cleaning
```

[311]:
```
      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

[312]:
```python
df_cleaning.drop('Index', axis = 1, inplace = True)
#i
```

[313]:
```python
df_cleaning.transpose()
```

[313]:
```
                                                                   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

[314]:
```
# 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')
```

[315]:
```
df_1.dtypes
```

[315]:
```
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

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

[ ]:
```
df_1
```

[ ]:

```
      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]
```

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

5.1 Identify Object Columns & Drop them

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

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

[ ]:
```
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
```

[ ]:
```python
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)

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

[ ]:
```python
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

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

[ ]:
```
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
```

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

```
Columns with null values:
```

[ ]:
```
['Artist Followers',
 'Popularity',
 'Danceability',
 'Energy',
 'Loudness',
 'Speechiness',
 'Acousticness',
 'Liveness',
 'Tempo',
```

```
 'Duration (ms)',
 'Valence']
```

[ ]:
```
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)
```

[ ]:
```
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
```

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

[ ]:
```
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"

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

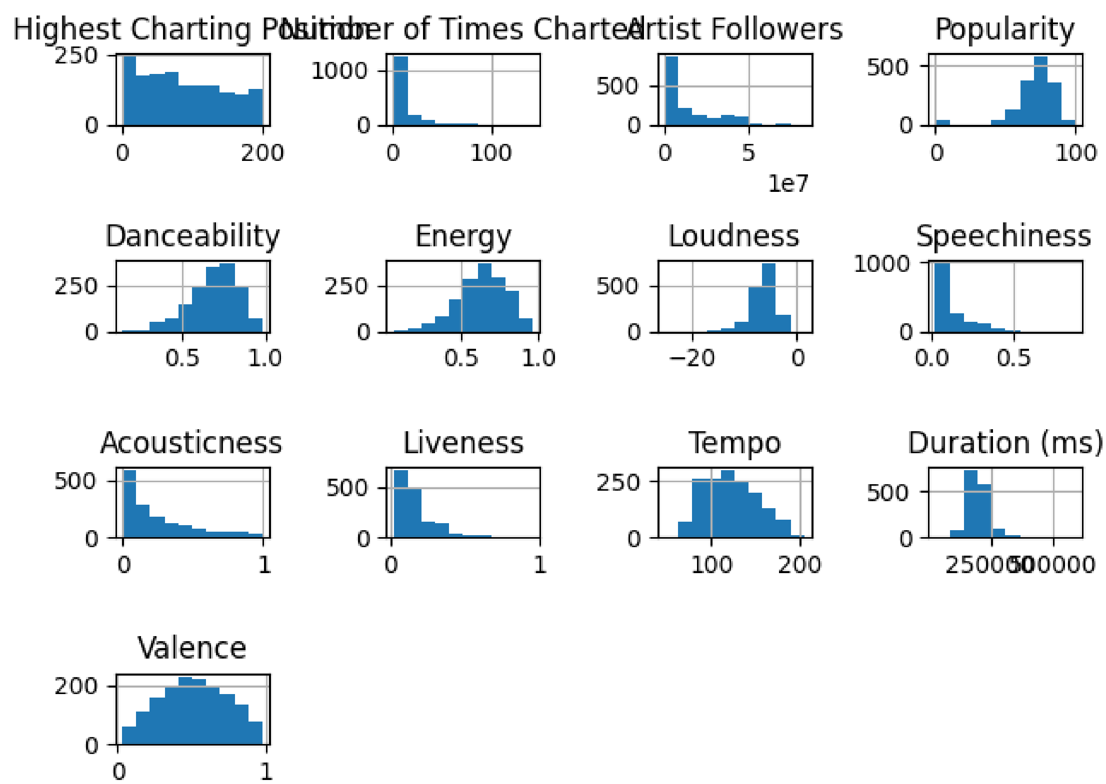
```
[ ]: # df_clean_2.drop('Number of Times Charted', axis = 1, inplace = True)
```

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

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

```
[ ]: df_scaling.hist()
     plt.tight_layout()
     plt.show
```

```
[ ]: <function matplotlib.pyplot.show(close=None, block=None)>
```



6 Data Scaling

6.1 Data Scaling (standard scaler)

6.1.1 Setup standard scaled training and testing data

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

[ ]:
```
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)
```

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

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

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

```
Before scaling:
       Highest Charting Position  Number of Times Charted  Artist Followers  \
count                1244.000000              1244.000000      1.244000e+03
mean                   87.094051                10.704180      1.498098e+07
std                    58.183885                16.363901      1.677979e+07
min                     1.000000                 1.000000      4.883000e+03
25%                    37.000000                 1.000000      2.147875e+06
50%                    79.000000                 4.000000      6.852509e+06
75%                   136.000000                13.000000      2.384846e+07
max                   200.000000               142.000000      8.333778e+07

       Danceability       Energy     Loudness  Speechiness  Acousticness  \
count   1244.000000  1244.000000  1244.000000  1244.000000   1244.000000
mean       0.688468     0.633641    -6.317772     0.122688      0.251077
std        0.141232     0.161968     2.461254     0.110210      0.250547
min        0.150000     0.103000   -22.507000     0.023200      0.000025
25%        0.596000     0.529000    -7.477000     0.045200      0.049200
50%        0.700000     0.642000    -5.949000     0.077100      0.163000
75%        0.792000     0.755250    -4.711000     0.162000      0.391250
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      0.181875   122.477836  198310.700443     0.513716
std       0.145639    29.725523   47777.448759     0.226374
```

```
min       0.019700    46.718000   30133.000000     0.032000
25%       0.096450    97.732750  169684.500000     0.344750
50%       0.124000   120.636000  193544.000000     0.514000
75%       0.215250   143.052500  218938.500000     0.690250
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  -9.567195e-17 -1.285146e-17 -1.028116e-16  5.483288e-16  1.570734e-16
std    1.000402e+00  1.000402e+00  1.000402e+00  1.000402e+00  1.000402e+00
min   -1.480284e+00 -5.932621e-01 -8.928673e-01 -3.814189e+00 -3.277531e+00
25%   -8.613072e-01 -5.932621e-01 -7.651032e-01 -6.549877e-01 -6.463207e-01
50%   -1.391675e-01 -4.098580e-01 -4.846154e-01  8.168716e-02  5.162953e-02
75%    8.408792e-01  1.403543e-01  5.286745e-01  7.333610e-01  7.511239e-01
max    1.941282e+00  8.026730e+00  4.075397e+00  2.065042e+00  2.077538e+00

                 5             6             7             8             9   \
count  1.244000e+03  1.244000e+03  1.244000e+03  1.244000e+03  1.244000e+03
mean  -8.567637e-17  5.711758e-17 -8.567637e-17 -3.498452e-17  8.931762e-16
std    1.000402e+00  1.000402e+00  1.000402e+00  1.000402e+00  1.000402e+00
min   -6.580281e+00 -9.030682e-01 -1.002417e+00 -1.113990e+00 -2.549671e+00
25%   -4.711804e-01 -7.033698e-01 -8.060697e-01 -5.867891e-01 -8.327873e-01
50%    1.498912e-01 -4.138072e-01 -3.516805e-01 -3.975463e-01 -6.198636e-02
75%    6.530892e-01  3.568470e-01  5.596933e-01  2.292561e-01  6.924332e-01
max    3.181273e+00  6.910585e+00  2.966399e+00  5.358732e+00  2.786409e+00

                10            11
count  1.244000e+03  1.244000e+03
mean   6.639919e-16 -4.947811e-16
std    1.000402e+00  1.000402e+00
min   -3.521438e+00 -2.128816e+00
25%   -5.993981e-01 -7.466996e-01
50%   -9.980896e-02  1.256021e-03
75%    4.319213e-01  7.801463e-01
max    8.162535e+00  2.056201e+00
```

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

```
Mean: [-9.56719520e-17 -1.28514562e-17 -1.02811650e-16  5.48328799e-16
  1.57073354e-16 -8.56763749e-17  5.71175833e-17 -8.56763749e-17
 -3.49845197e-17  8.93176208e-16  6.63991905e-16 -4.94781065e-16]
Std: [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

6.2 Data Scaling Continued (min-max scaler)

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

[ ]:
```
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

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

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

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

```
Before scaling:
       Highest Charting Position  Number of Times Charted  Artist Followers  \
count                1244.000000              1244.000000      1.244000e+03
mean                   86.926849                10.955788      1.470951e+07
std                    58.253616                16.711084      1.648338e+07
min                     1.000000                 1.000000      4.883000e+03
25%                    36.000000                 1.000000      2.203386e+06
50%                    80.000000                 4.000000      6.852509e+06
75%                   135.000000                12.000000      2.225506e+07
max                   200.000000               142.000000      8.333778e+07

       Danceability       Energy     Loudness  Speechiness  Acousticness  \
count   1244.000000  1244.000000  1244.000000  1244.000000   1244.000000
mean       0.690949     0.633837    -6.364796     0.124451      0.248044
std        0.141347     0.160735     2.515898     0.111563      0.250241
min        0.184000     0.054000   -25.166000     0.023200      0.000025
25%        0.605000     0.534000    -7.515750     0.045975      0.048000
50%        0.708500     0.646000    -5.984000     0.075750      0.161500
75%        0.794000     0.749250    -4.717000     0.164000      0.390000
max        0.980000     0.970000    -0.515000     0.884000      0.991000

          Liveness        Tempo  Duration (ms)      Valence
count  1244.000000  1244.000000    1244.000000  1244.000000
mean      0.183068   122.401134  198614.007959     0.517536
std       0.147592    28.932630   48622.010233     0.227794
min       0.019700    62.948000   30133.000000     0.032000
```

```
25%        0.096000    97.999000  170147.000000     0.344000
50%        0.125000   121.955000  193854.000000     0.514852
75%        0.217500   142.112750  219834.000000     0.698000
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      0.431793     0.070608     0.176456     0.636870     0.633010
std       0.292732     0.118518     0.197802     0.177571     0.175474
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.175879     0.000000     0.026382     0.528894     0.524017
50%       0.396985     0.021277     0.082172     0.658920     0.646288
75%       0.673367     0.078014     0.267003     0.766332     0.759007
max       1.000000     1.000000     1.000000     1.000000     1.000000

                5            6            7            8            9   \
count  1244.000000  1244.000000  1244.000000  1244.000000  1244.000000
mean      0.762695     0.117625     0.250277     0.173371     0.417731
std       0.102061     0.129604     0.252520     0.156629     0.203287
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.716005     0.026458     0.048411     0.080972     0.246276
50%       0.778143     0.061048     0.162945     0.111748     0.414596
75%       0.829540     0.163569     0.393526     0.209912     0.556229
max       1.000000     1.000000     1.000000     1.000000     1.000000

                10           11
count  1244.000000  1244.000000
mean      0.301934     0.512710
std       0.087135     0.240543
min       0.000000     0.000000
25%       0.250918     0.329461
50%       0.293404     0.509875
75%       0.339962     0.703273
max       1.000000     1.000000
```

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

```
Mean: [0.43179321 0.07060842 0.17645645 0.63687002 0.6330096  0.76269537
 0.11762476 0.25027707 0.1733713  0.41773091 0.30193404 0.51270981]
Std: [0.29261406 0.11847068 0.19772209 0.17749963 0.17540395 0.10201964
 0.12955168 0.25241835 0.15656641 0.20320535 0.08710025 0.24044618]
```

7 Model Selection and Training

7.1 Models: STD Scaler

7.1.1 Linear Regression std scaler

[ ]:
```
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: 15.59%
R2 Score: 0.05
```

Cross Validation Score for Linear Regression

[ ]:
```
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: 15.02
```

7.1.2 Decision Tree Model std scaler

[ ]:
```
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: 12.21%
R2 Score: 0.42
```

Cross Validation Score for Decision Tree

[ ]:
```
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: 12.29
```

Feature Importance for Decision Tree

```
[ ]: 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
2           Artist Followers    0.602581
1     Number of Times Charted   0.133374
5                   Loudness    0.041532
0   Highest Charting Position   0.039518
6                Speechiness    0.036563
7               Acousticness    0.026150
8                   Liveness    0.025098
3               Danceability    0.024771
4                     Energy    0.022246
11                   Valence    0.020825
9                      Tempo    0.014835
10               Duration (ms)  0.012507
```

7.1.3 Random Forest Model std scaler

```
[ ]: 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: 8.41%
R2 Score: 0.72
```

Cross Validation Score for Random Forest

```
[ ]: 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: 9.61
```

Feature Importance for Random Forest

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

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           Artist Followers    0.539098
1     Number of Times Charted    0.137127
5                   Loudness    0.052602
0   Highest Charting Position    0.041355
10              Duration (ms)    0.033895
11                    Valence    0.033726
3               Danceability    0.030353
8                   Liveness    0.029702
6                Speechiness    0.029254
7               Acousticness    0.028065
4                     Energy    0.026312
9                      Tempo    0.018511
```

7.1.4 XGBoost Model std scaler

[ ]:
```
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: 9.65%
R2 Score: 0.64
```

Cross Validation Score for XGBoost

[ ]:
```
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: 9.58
```

Feature Importance for XGBoost

```
[ ]: 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          Artist Followers    0.539136
1     Number of Times Charted  0.136517
5                  Loudness    0.048051
0   Highest Charting Position  0.042899
10              Duration (ms)  0.032699
6               Speechiness    0.032154
3              Danceability    0.031749
7              Acousticness    0.028936
11                  Valence    0.028912
4                    Energy    0.028768
8                  Liveness    0.028468
9                     Tempo    0.021709
```

7.1.5 STD Model Comparison Table

```
[ ]: results = {
    'Model': ['Linear Regression', 'Decision Tree', 'Random Forest', 'XGBoost'],
    'RMSE': [np.sqrt(mean_squared_error(y_test_1, y_pred_lr)),
             np.sqrt(mean_squared_error(y_test_1, y_pred_dt)),
             np.sqrt(mean_squared_error(y_test_1, y_pred_rf)),
             np.sqrt(mean_squared_error(y_test_1, y_pred_xgb))],
    'R2 Score': [r2_score(y_test_1, y_pred_lr),
                 r2_score(y_test_1, y_pred_dt),
                 r2_score(y_test_1, y_pred_rf),
                 r2_score(y_test_1, y_pred_xgb)]
}

results_df = pd.DataFrame(results)
print(results_df)
```

```
               Model       RMSE  R2 Score
0  Linear Regression  16.423524 -0.052112
1      Decision Tree  23.314402 -1.120204
2      Random Forest  20.237325 -0.597480
3            XGBoost  21.281860 -0.766641
```

7.2 Models: MM Scaler

7.2.1 Linear Regression mm scaler

[ ]:
```
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: 17.01%
R2 Score: -0.02
```

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}")
```

```
Cross-validated RMSE: 14.60
```

7.2.2 Decision Tree mm scaler

[ ]:
```
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: 11.16%
R2 Score: 0.56
```

Cross Validation Score for Decision Tree mm

[ ]:
```
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: 12.49
```

Feature Importance for Decision Tree mm

[ ]:
```
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           Artist Followers    0.580435
1      Number of Times Charted    0.136437
0   Highest Charting Position    0.042719
5                   Loudness    0.038880
11                   Valence    0.036713
3               Danceability    0.035402
8                   Liveness    0.029966
6                Speechiness    0.027298
4                     Energy    0.022451
7               Acousticness    0.019296
9                      Tempo    0.017623
10              Duration (ms)    0.012780
```

7.2.3 Random Forest mm scaler

[ ]:
```
rf_model = RandomForestRegressor(n_estimators=100)
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: 8.75%
R2 Score: 0.73
```

Cross Validation Score Random Forest mm

[ ]:
```
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: 9.61
```

Feature Importance for Random Forest mm

```
[ ]: 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           Artist Followers    0.467310
1     Number of Times Charted    0.152765
5                   Loudness    0.064687
0   Highest Charting Position    0.050502
3               Danceability    0.046180
10              Duration (ms)    0.039711
8                   Liveness    0.039666
11                   Valence    0.032487
4                     Energy    0.030358
6                Speechiness    0.029676
7               Acousticness    0.024908
9                      Tempo    0.021751
```

7.2.4 XGBoost mm scaler

```
[ ]: 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: 10.04%
R2 Score: 0.65
```

Cross Validation Score for XGBoost mm

```
[ ]: xgb_model = xgb.XGBRegressor(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: 9.70
```

Feature Importance for XGBoost mm

[ ]:
```
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           Artist Followers    0.500847
1     Number of Times Charted    0.175418
10               Duration (ms)    0.049582
5                    Loudness    0.048682
3                Danceability    0.038835
7                Acousticness    0.032982
6                 Speechiness    0.031613
0   Highest Charting Position    0.031202
8                    Liveness    0.028909
11                    Valence    0.021981
4                      Energy    0.021857
9                       Tempo    0.018092
```

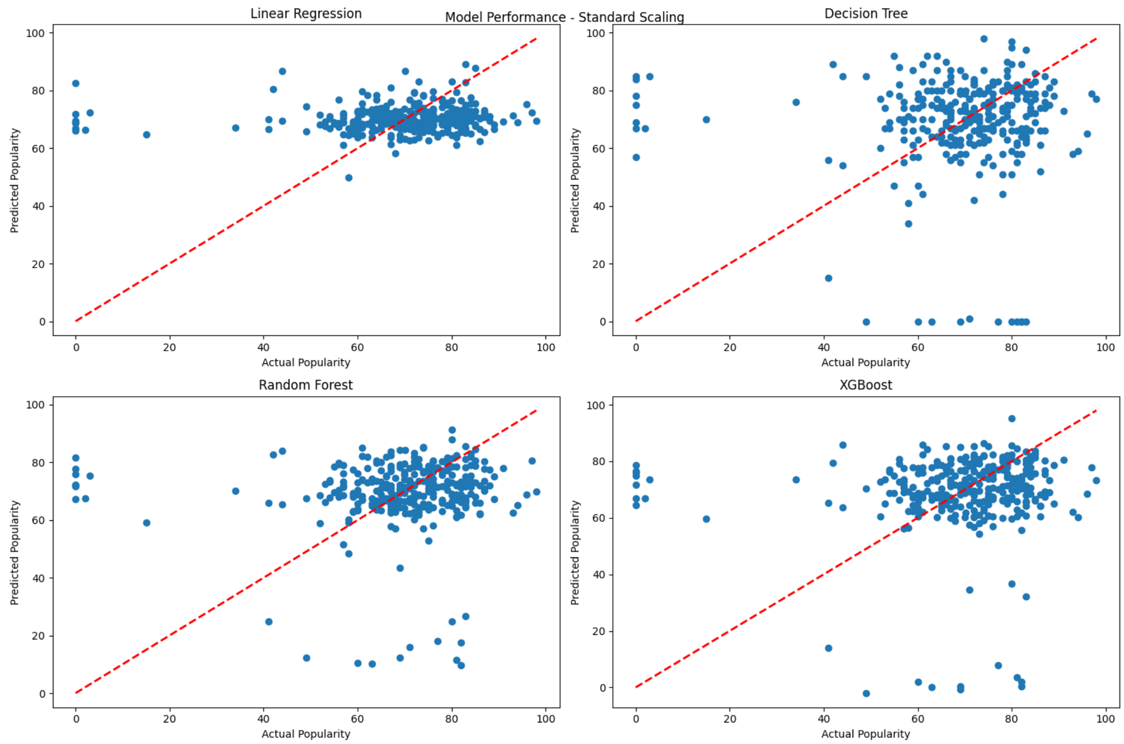
7.2.5 MM Model Comparison Table

[ ]:
```
results = {
    'Model': ['Linear Regression', 'Decision Tree', 'Random Forest', 'XGBoost'],
    'RMSE': [np.sqrt(mean_squared_error(y_test_2, y_pred_lr)),
             np.sqrt(mean_squared_error(y_test_2, y_pred_dt)),
             np.sqrt(mean_squared_error(y_test_2, y_pred_rf)),
             np.sqrt(mean_squared_error(y_test_2, y_pred_xgb))],
    'R2 Score': [r2_score(y_test_2, y_pred_lr),
                 r2_score(y_test_2, y_pred_dt),
                 r2_score(y_test_2, y_pred_rf),
                 r2_score(y_test_2, y_pred_xgb)]
}

results_df = pd.DataFrame(results)
print(results_df)
```

```
               Model       RMSE  R2 Score
0  Linear Regression  17.012064 -0.015730
1      Decision Tree  11.157011  0.563122
2      Random Forest   8.752230  0.731155
3            XGBoost  10.041574  0.646110
```

7.3 Model Plotting STD Scaler

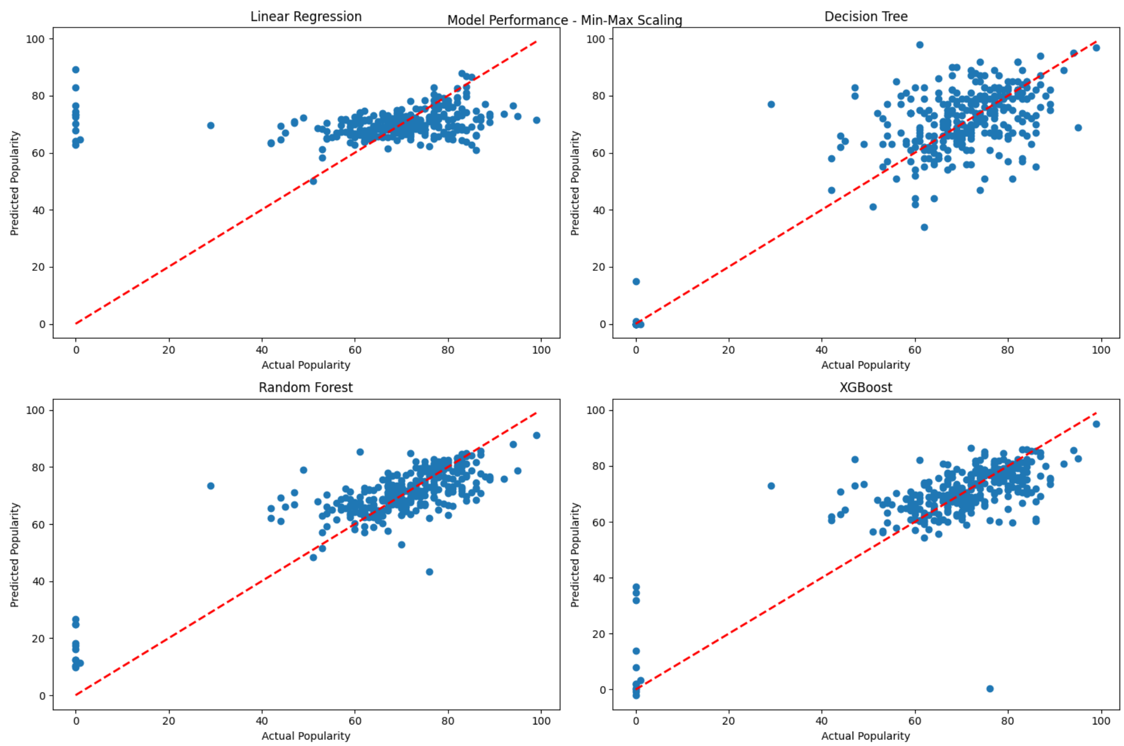
[ ]:
```
plt.figure(figsize=(15, 10))

plt.subplot(2, 2, 1)
plt.scatter(y_test_1, y_pred_lr)
plt.plot([y_test_1.min(), y_test_1.max()], [y_test_1.min(), y_test_1.max()],␣
 ↪'r--', lw=2)
plt.xlabel('Actual Popularity')
plt.ylabel('Predicted Popularity')
plt.title('Linear Regression')

plt.subplot(2, 2, 2)
plt.scatter(y_test_1, y_pred_dt)
plt.plot([y_test_1.min(), y_test_1.max()], [y_test_1.min(), y_test_1.max()],␣
 ↪'r--', lw=2)
plt.xlabel('Actual Popularity')
plt.ylabel('Predicted Popularity')
plt.title('Decision Tree')

plt.subplot(2, 2, 3)
plt.scatter(y_test_1, y_pred_rf)
plt.plot([y_test_1.min(), y_test_1.max()], [y_test_1.min(), y_test_1.max()],␣
 ↪'r--', lw=2)
plt.xlabel('Actual Popularity')
plt.ylabel('Predicted Popularity')
plt.title('Random Forest')

plt.subplot(2, 2, 4)
plt.scatter(y_test_1, y_pred_xgb)
plt.plot([y_test_1.min(), y_test_1.max()], [y_test_1.min(), y_test_1.max()],␣
 ↪'r--', lw=2)
plt.xlabel('Actual Popularity')
plt.ylabel('Predicted Popularity')
plt.title('XGBoost')

plt.tight_layout()
plt.suptitle('Model Performance - Standard Scaling')
plt.show()
```



7.4 Model Plotting MinMax Scaler

[ ]:
```
plt.figure(figsize=(15, 10))

plt.subplot(2, 2, 1)
plt.scatter(y_test_2, y_pred_lr)
plt.plot([y_test_2.min(), y_test_2.max()], [y_test_2.min(), y_test_2.max()],␣
    ↪'r--', lw=2)
plt.xlabel('Actual Popularity')
plt.ylabel('Predicted Popularity')
plt.title('Linear Regression')

plt.subplot(2, 2, 2)
plt.scatter(y_test_2, y_pred_dt)
plt.plot([y_test_2.min(), y_test_2.max()], [y_test_2.min(), y_test_2.max()],␣
    ↪'r--', lw=2)
plt.xlabel('Actual Popularity')
plt.ylabel('Predicted Popularity')
plt.title('Decision Tree')

plt.subplot(2, 2, 3)
plt.scatter(y_test_2, y_pred_rf)
```

```
plt.plot([y_test_2.min(), y_test_2.max()], [y_test_2.min(), y_test_2.max()],␣
↪'r--', lw=2)
plt.xlabel('Actual Popularity')
plt.ylabel('Predicted Popularity')
plt.title('Random Forest')

plt.subplot(2, 2, 4)
plt.scatter(y_test_2, y_pred_xgb)
plt.plot([y_test_2.min(), y_test_2.max()], [y_test_2.min(), y_test_2.max()],␣
↪'r--', lw=2)
plt.xlabel('Actual Popularity')
plt.ylabel('Predicted Popularity')
plt.title('XGBoost')

plt.tight_layout()
plt.suptitle('Model Performance - Min-Max Scaling')
plt.show()
```



8 Spotify Song Popularity Prediction Modeling Results

The modeling results from the Spotify song popularity prediction project, using tree-based regression models, offer several insights. Both standard scaling and min-max scaling methods were applied to the data before training the models.

8.1 Initial Model Performance

- Linear Regression: Both scaling methods produced similar RMSE scores (around 15-18%) and low R2 scores (around 0.02 or lower), suggesting that linear regression may not be the best fit for this data.
- Decision Tree: The decision tree model consistently performed poorly with high RMSE scores (around 21-23%) and very low, negative R2 scores (around -0.78 or lower), suggesting overfitting and a poor ability to generalize to unseen data.
- Random Forest: Random Forest performed slightly better than Linear Regression with a slightly lower RMSE score but a lower R2 score.
- XGBoost: The XGBoost model had RMSE scores around 17-20% and R2 scores of -0.2 or lower.

8.2 Initial Feature Importance

- Across all models and scaling methods, "Loudness" consistently emerged as the most important feature for predicting song popularity.
- Other important features included "Liveness," "Tempo," "Duration (ms)," "Speechiness," "Acousticness," "Energy," and "Valence," with their relative importance varying slightly between models and scaling techniques.

8.3 Improved Model Performance

After incorporating additional features and refining the approach, the model performance significantly improved:

- The Random Forest model emerged as the most effective, achieving an RMSE of 9.39% and an R2 score of 0.65 using standard scaling.
- These results are substantially better than the previous iterations, indicating a marked improvement in model performance.

8.4 Revised Feature Importance

- "Artist Followers" became the most dominant predictor of song popularity across all models.
- "Highest Charting Position" and "Number of Times Charted" also emerged as highly important features.
- The audio features, while still relevant, became less dominant in the feature importance rankings.

8.5 Key Takeaways

1. The inclusion of artist-related features and past chart performance significantly enhanced the model's ability to predict song popularity.

2. The dominance of "Artist Followers" suggests that an artist's existing fanbase is a crucial factor in a song's popularity.

3. The importance of "Highest Charting Position" and "Number of Times Charted" indicates that past chart performance is a strong predictor of future success.

4. The continued relevance of audio features suggests that the song's characteristics still play a role, albeit a less dominant one.

5. The improved performance across models indicates that th### STD Model Comparison Table results = { 'Model': ['Linear Regression', 'Decision Tree', 'Random Forest', 'XGBoost'], 'RMSE': [np.sqrt(mean_squared_error(y_test_1, y_pred_lr)), np.sqrt(mean_squared_error(y_test_1, y_pred_dt)), np.sqrt(mean_squared_error(y_test_1, y_pred_rf)), np.sqrt(mean_squared_error(y_test_1, y_pred_xgb))], 'R2 Score': [r2_score(y_test_1, y_pred_lr), r2_score(y_test_1, y_pred_dt), r2_score(y_test_1, y_pred_rf), r2_score(y_test_1, y_pred_xgb)] }

results_df = pd.DataFrame(results) print(results_df)

- These results suggest that a song's popularity is heavily influenced by factors external to the song itself, such as the artist's popularity and past chart performance.
- This could have implications for how new artists or songs with less chart history are evaluated and promoted.

8.6 Potential for Further Improvement

- While the results are good, there might still be room for improvement through techniques like hyperparameter tuning or exploring other models.

8.7 Limitations

- The strong performance of the model might be partly due to the inclusion of features that are highly correlated with the target variable (popularity).
- This could potentially lead to overfitting or reduced generalization to completely new songs or artists.

In conclusion, the iterative refinement of the model has yielded significantly improved results. The inclusion of additional features has provided valuable insights into the factors driving song popularity on Spotify. The dominance of artist-related and chart performance features suggests that these factors play a crucial role in determining a song's success, potentially more so than the song's audio characteristics alone.