IT350 - Data Analytics

Automatic Playlist Recommendation System

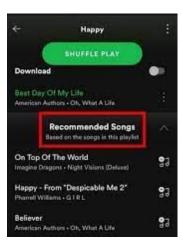
Saumya Karri	181ME273
Anshuman Sinha	181EE209
Sathvika B. Mahesh	181CV141

Introduction

- A playlist is simply a sequence of tracks intended to be listened to together
- Automatic music playlist continuation is a form of the more general task of sequential recommendation
- Danceability, Energy, Loudness, Instrumentalness, Liveliness, Tempo
- Broader usage in music softwares like Spotify and in any data-streaming platforms like YouTube.









Methodology

Data Extraction

- One slice of dataset chosen randomly playlist 20000 to 20999
- Attributes like album name, danceability, energy, acousticness etc.
- Creating a dataframe which gives trackwise features

Data cleaning

- Dropped non-quant and null values
- Normalized using z-score normalization
- Found correlation between features using heatmap
- Dropped heavily correlated attributes

Data modelling

- Used 6 classification models on the data
- Predicted the ratings of each song
- Calculated accuracy scores for each model

Implementation

Classification models

- 1. Linear regression
- 2. Logistic regression
- 3. Random forest regression

- 4. Random forest classification
- 5. Gradient Boosting regression
- 6. K nearest neighbour classification

Linear regression is a supervised Machine Learning approach for predicting continuous variables.

Logistic regression is a supervised Machine Learning algorithm that helps in binary classification.

Random Forest regression/classification is an ensemble approach that uses several decision trees and a technique called Bootstrap and Aggregation, also known as bagging, to solve both regression and classification problems. Classification transfers the input data item to a set of discrete labels, regression converts the input data item into continuous real values.

Gradient Boosting builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions.

K-Neighbors classification is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions).

Snapshots

```
print("Summary Statistics for duration\n")
  The 'number of tracks' characteristic of each playlist
                                                                                                                               describe = df['duration_ms'].describe()
                                                                                                                              print(str(describe)+'\n')
     喧斗及目… 🛍
      print("Summary Statistics for number of tracks\n")
                                                                                                                              print('Variance\t' + str(df['duration_ms'].var()))
                                                                                                                              print('Range\t\t' + str(max(df['duration_ms']) - min(df['duration_ms'])))
      describe = df['num_tracks'].describe()
      print(str(describe)+'\n')
                                                                                                                           Summary Statistics for duration
      print('Variance\t' + str(df['num_tracks'].var()))
                                                                                                                                   1.000000e+03
      print('Range\t\t' + str(max(df['num_tracks']) - min(df['num_tracks'])))
                                                                                                                                   1.604941e+07
28] 🗸 0.9s
                                                                                                                                   1.280911e+07
   Summary Statistics for number of tracks
                                                                                                                                   1.044377e+06
                                                                                                                           min
                                                                                                                                   6.118938e+06
            1000.000000
                                                                                                                                   1.228655e+07
   mean
             68.101000
                                                                                                                                   2.234913e+07
             53.532612
                                                                                                                                   8.309966e+07
              5.000000
                                                                                                                           Name: duration_ms, dtype: float64
              26.000000
   50%
             52.000000
                                                                                                                                          164073286096730.56
             95.000000
                                                NTracksHist = df['num tracks'].hist()
                                                                                                                                          82055288
             246.000000
                                                xlabel = NTracksHist.set xlabel("Number of tracks in playlist")
   Name: num_tracks, dtype: float64
                                                ylabel = NTracksHist.set ylabel("Frequency")
                                                title = NTracksHist.set title("Number of tracks")
   Variance
                  2865.740539539543
   Range
                                                                      Number of tracks
                                                250
                                                200
                                               · 150
                                                100
                                                  50
                                                                        100
                                                                                 150
                                                                    Number of tracks in playlist
                                           The 'duration in ms' characteristic of each playlist
```

The 'duration in ms' characteristic of each playlist

Duration = df['duration ms'].hist() xlabel = Duration.set xlabel("Duration in ms") ylabel = Duration.set_ylabel("Frequency") title = Duration.set_title("Duration") √ 0.5s Duration 400 350 300 > 250 200 100 50 Duration in ms 1e7

Snapshots

Track Feature

Correlation Heat Map

This graph shows the correlation among the 16 features per song of the chosen set of 1000 playlists.

track_duration_ms	1	0.028	-0.079	0.019	0.001	-0.018	-0.039	-0.082	0.087	0.05	-0.16	0.0038	0.014	0.0094	0.02	-0.069	-0.02	-0.035	-0.067
track_popularity	0.028	1	0.17	0.093	-0.0073	0.18	0.056	-0.17	-0.25	-0.046	0.083	-0.005	0.28	0.56	0.84	-0.045	0.18	-0.024	-0.15
danceability	-0.079	0.17	1	0.08	0.00025	0.16	0.18	-0.2	-0.15	-0.093	0.39	-0.14	0.1	0.15	0.11	0.08	0.3	-0.11	-0.21
energy	0.019	0.093	0.08	1	0.035	0.75	0.081	-0.65	-0.15	0.18	0.34	0.16	0.021	0.068	0.054	0.087	0.018	-0.071	-0.43
key ·	0.001	-0.0073	0.00025	0.035	1	0.014	0.008	-0.027	-0.0036	-0.0051	0.036	0.0076	0.006	-0.0054	-0.012	-0.012	-0.0069	-0.13	-0.022
loudness	-0.018	0.18	0.16	0.75	0.014	1	0.027	-0.57	-0.33	0.081	0.22	0.14	0.12	0.19	0.13	0.28	0.093	-0.055	-0.42
speechiness	-0.039	0.056	0.18	0.081	0.008	0.027	1	-0.011	-0.11	0.2	0.019	0.031	0.08	0.085	0.079	0.13	0.48	-0.089	0.025
acousticness ·	-0.082	-0.17	-0.2	-0.65	-0.027	-0.57	-0.011	1	0.19	-0.053	-0.16	-0.13	-0.078	-0.14	-0.12	-0.12	-0.13	0.088	0.63
instrumentalness	0.087	-0.25	-0.15	-0.15	-0.0036	-0.33	-0.11	0.19	1	-0.043	-0.16	-0.019	-0.13	-0.24	-0.21	0.011	-0.16	-0.039	0.19
liveness	0.05	-0.046	-0.093	0.18	-0.0051	0.081	0.2	-0.053	-0.043	1	0.014	0.023	-0.01	-0.01	-0.019	0.0084	0.07	-0.0061	0.0046
valence ·	-0.16	0.083	0.39	0.34	0.036	0.22	0.019	-0.16	-0.16	0.014	1	0.025	-0.021	-0.0065	0.019	-0.24	-0.071	-0.006	-0.17
tempo	0.0038	-0.005	-0.14	0.16	0.0076	0.14	0.031	-0.13	-0.019	0.023	0.025	1	-0.0043	-0.002	-0.0051	0.035	0.0013	0.013	-0.082
followers	0.014	0.28	0.1	0.021	0.006	0.12	0.08	-0.078	-0.13	-0.01	-0.021	-0.0043	1	0.58	0.35	0.098	0.15	-0.056	-0.049
artist_popularity	0.0094	0.56	0.15	0.068	-0.0054	0.19	0.085	-0.14	-0.24	-0.01	-0.0065	-0.002	0.58	1	0.64	0.066	0.25	-0.047	-0.12
album_popularity	0.02	0.84	0.11	0.054	-0.012	0.13	0.079	-0.12	-0.21	-0.019	0.019	-0.0051	0.35	0.64	1	0.017	0.2	-0.016	-0.096
release_year	-0.069	-0.045	0.08	0.087	-0.012	0.28	0.13	-0.12	0.011	0.0084	-0.24	0.035	0.098	0.066	0.017	1	0.22	-0.08	-0.07
track_explicit	-0.02	0.18	0.3	0.018	-0.0069	0.093	0.48	-0.13	-0.16	0.07	-0.071	0.0013	0.15	0.25	0.2	0.22	1	-0.09	-0.088
mode	-0.035	-0.024	-0.11	-0.071	-0.13	-0.055	-0.089	0.088	-0.039	-0.0061	-0.006	0.013	-0.056	-0.047	-0.016	-0.08	-0.09	1	0.069
in_playlist	-0.067	-0.15	-0.21	-0.43	-0.022	-0.42	0.025	0.63	0.19	0.0046	-0.17	-0.082	-0.049	-0.12	-0.096	-0.07	-0.088	0.069	1
	k_duration_ms -	ack_popularity -	danceability -	energy -	key -	loudness -	speechiness -	acousticness -	strumentalness -	liveness -	valence -	tempo -	followers -	rtist_popularity -	oum_popularity -	release_year -	track_explicit -	- mode -	in_playlist -

- 0.8 - 0.6 - -0.6

Results & Analysis

Classification method	Accuracy scores	Confusion matrix
Logistic Regression	0.95589	[[5749 120] [169 373]]
Random Forest Classification	0.99988	[[5869 0] [8 534]]
Gradient Boosting Regressor	0.973254	[[5869 0] [8 534]]
K-Neighbor Classification	0.96064	[[5723 146] [154 388]]

10 fold cross-validation with CV score **0.95492**

Literature Survey

- "Current Challenges and Visions in Music Recommender Systems Research" M. Schedl, H. Zamani, C.-W. Chen, Y. Deldjoo, M. Elahi.
 - Paper link: https://arxiv.org/pdf/1710.03208.pdf
 - Gives a broad definition of the problem and provides many new scopes and potential innovations for further research in Music Recommender Systems.
- "An Analysis of Approaches Taken in the ACM RecSys Challenge 2018 for Automatic Music Playlist Continuation" H. Zamani, M. Schedl, P. Lamere, C.-W. Chen.
 - Base Paper link: https://arxiv.org/pdf/1810.01520.pdf
 - Lays out a complete summary of important metrics and approaches to solve the problem of Automatic Music Playlist Continuation.

Thank you