Song Recommendation System with Spotify

Ata Deniz Arslaner

[1]

A. Introduction

This is an article about building a recommendation system using Spotify's data to predict the items that the user might like. Recommendation systems recommend certain things that people might like based on your own watch history or you and friends watch history as a collective. This article explains how to get Spotify's own data using Spotify API and spotipy library and to create a structured CSV file, preprocess the data for getting clear results from the classification algorithm, which is a decision tree in this case, and finally making inferences according to the results. There are some experiments about examining the dataset such as a quick look at correlations between features and specific features with the target variable. Some tries has been done to define a parameter (threshold value) for feature selection with correlation coefficient metric. For the prediction the decision tree algorithm applied for the classification. Python programming language used as an interpreter, and the platform used is Pycharm for python.

Steps
1. Data extraction
2. Data preprocessing (Creating graphs, metrices, feature selection etc.)
3. Working on features' correlations with graphs and making inferences
4. Prediction and Recommendation

[1]

B. Data description

- Dataset extracted from Spotify with Spotify’s API technic:
- Creating a playlist with liked and unliked songs
- Getting the playlist from user
- Some features extracted according to songs
- Finally data transformed the structured csv. format.

After the data is transformed to csv format with API, our data set consists of approximately 10 thousand variables and has 18 dimensions. that is, there are 17 properties belonging to each variable. A dataframe with 13 features was created by removing the track_num, track_id, track_name columns from these features. The purpose of this is to examine the data better and make more specific comments. Let's explain that 13 dimensional data. We deal with the data that is without first_artist and liked(target variable) columns.

The features are:

- **acousticness:** A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.

- **danceability:** Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.

- **duration_ms:** The duration of the track in milliseconds.

- **energy:** Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.

- **instrumentalness:** Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.
- **key:** The key the track is in. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on. If no key was detected, the value is -1.

- **liveness:** Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.

- **loudness:** The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typically range between -60 and 0 db.

- **mode:** Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.

- **speechiness:** Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.

- **tempo:** The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.

- **time_signature:** An estimated time signature. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure). The time signature ranges from 3 to 7 indicating time signatures of "3/4", to "7/4".

- **valence:** A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

track_num	track_id	track_name	first_artist	acousticness	danceability	duration_ms	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_s
1	5bvCNjedRi8BWt1Z9D58ih	Dry Your Tears	Jungle	0.981	0.187	80000	0.257	0.839	5	0.19	-14.245	1	0.0377	78.359	5
2	2dSbH1DD8CEqHqyS1Ey0bd	Dickmatized	Jada Kingdom	0.529	0.817	140337	0.82	0	1	0.11	-3.59	0	0.256	101.206	4
3	239SrBPZi0xs1OQZmedr6q	A Tribe Called Kotori	Oliver Koletzki	0.71	0.627	370453	0.496	0.831	5	0.343	-11.866	0	0.0396	95.0	4
4	5BAhmGN4WgfCuSISIeLFO2	Chaque Fois	Stavroz	0.0516	0.927	450232	0.498	0.487	0	0.323	-13.736	1	0.308	118.998	4
5	6o2FQGObDH7kCpYlvJH2ai	Get Low	Butch U	0.323	0.77	221714	0.88	0.293	8	0.111	-3.734	1	0.0883	104.981	4
6	7K1TsEgLCoNxr7tvPRJgrL	Make It Happen	RÜFÜS DU SOL	0.00882	0.595	312120	0.723	0.0515	9	0.575	-7.639	0	0.0275	119.987	4
7	5rScp9IpziIsR6b93vcnQ	Walking On A Dream	Empire of the Sun	0.257	0.871	198440	0.701	7.52e-06	5	0.0589	-5.594	0	0.0458	126.975	4
8	5w3CtwPT5e0k8zg2LyjvNN	Commitment Issues	Central Cee	0.463	0.8	150664	0.71	0	5	0.339	-6.403	0	0.295	147.171	4
9	6AOkOnjFHab76VW08Q8ZsI	Out Till Late	Micks	0.488	0.804	205845	0.78	0	2	0.374	-5.984	1	0.341	141.973	4
10	4GSyA0YDBON0qD8fvqmPan	Wraith	Chance the Rapper	0.162	0.957	147692	0.528	1.3e-06	7	0.615	-6.025	0	0.128	130.001	4

C. Steps

There are 3 parts that we applied some methods; 1. Data extraction, API technic has been applied in this part for getting datasets as csv format, 2. Data preprocessing, the method that choosen is feature selection. With correlation coefficient metric correlations of features with each other and target variable has been calculated. The dimension of data reduced to 8 included “liked” column. 3. the method that we preferred in this part is classification and the algorithm for classification method is decision tree.

1. API Technic

For this project, Spotify’s content such as playlist and tracks were used. To access user-related data through the API, an application must be authorized by the user, using client credentials. This data is retrieved through Spotify’s API, which returns a JSON file containing the response content when queried. There are many queries you can send to this API, such as getting the user’s saved tracks or a playlist’s items, removing a track from a playlist, following a friend, checking if a track is saved or not, getting the albums of an artist and so on. In this project, to get the audio features of every single track in two playlists (one for liked and one for unliked songs), first we queried the API for the playlist items and then requested the audio features of that track. In the dataset, with the proper data cleaning, there are a total of 9996 tracks and their audio features available.

2. Data preprocessing

Firstly we checked are there any missing value in the dataset if there is any missing value we can fill with median of the feature or most frequent method but since there is no missing value in dataset it is not necessary. We calculate the correlations between features and specific feature with target variable and the method that we preferred is feature selection because we already extract the features with spotify API system therefor again extracting new features is not the case we should work with the same features in every process. Also the explainability is easier. For selecting feature from the existence columns the correlations coefficients with target variables method was chosed since we do not interest with relevance between features we only worked with relevance between features and target variable. There must be parameter for correlation coefficient for electing some features according to threshold value that we determined as 0.1. There have been several attempts at what value should be used for threshold and finally decided on 0.1. Finally we reduced the dimension 13 to 5 so we worked with 5 feature.

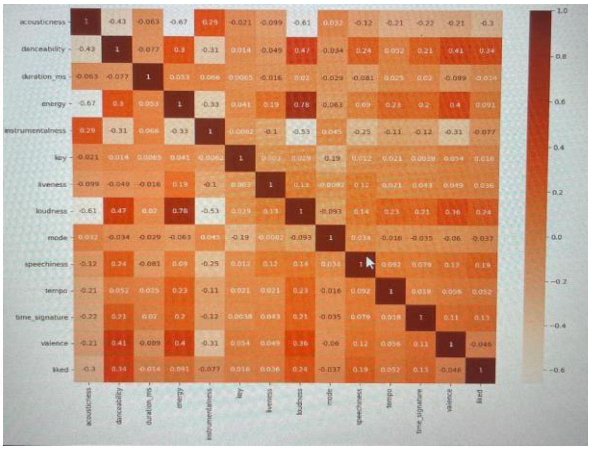
3. Model selection and prediction

Until now we made preprocess of the data. Firstly the splitted into 2 part; test and train %30 for test part %70 for train part, there also some tries for deciding on splitting the data. The model selected and the variables put on the own classes. The model that preferred is classification. Since we have target variable that we have to predict we should choose classification method instead of clustering also there must be 2 class in this case these are yes or no(1 – 0) so regression is not the case. So for the classification algorithm we decide on decision tree. For decision tree we have threshold value for parameter that determine the depth of tree this threshold determined according to entropies of the features it means impurity of the variable. The purest feature was put on the last node.

D. Experiments and some conclusions about the experiments

Data Preprocessing

as we mentioned there are experiments about correlations for determine the threshold value of correlation coefficients



this is the heatmap of correlation coefficients of features but we interest only the last column.
the correlation coefficients of features with target variables(liked) are:

acousticness 0.299174
danceability 0.343320
duration_ms 0.014434
energy 0.090874
instrumentalness 0.077025
key 0.015963
liveness 0.035516
loudness 0.238656
mode 0.036766
speechiness 0.185767
tempo 0.051706
time_signature 0.125790
valence 0.046216
liked 1.000000

so the 0.1 is the best for that value because when we increase that we can skip some important features.

formula of coefficient correlation;

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$$

E. Conclusion

We applied classification model with decision tree algorithm for predicting recommending the songs to user. The dataset taken from spotify with API system, for the preprocessing part the coefficient correlation comparison with target variable has been applied and some features elected (feature selection). our splitting data procedure is pretty well for that case. we used accuracy score for evaluation metric and checked the applicability of our model. Since it is %80 it is enough for the recommendation program but actually this is not a pretty well percentage because the rate of error so high for such ML cases.

References

https://developer.spotify.com/documentation/web-api/reference/#/operations/get-audio-features