Skip prediction using decision trees

Jasmijn Bookelmann
Radboud University
Nijmegen, Netherlands
jasmijn.bookelmann@ru.nl

***Abstract*—NOTE THAT THIS IS AN ASSIGNMENT PAPER NOT BASED ON ACTUAL RESEARCH - Skip prediction is predicting whether a user will skip a song or not. This is a good indication of how much the user likes this song. Therefore, it is an important component of recommender systems. Such as the spotify algorithm. We have created a skip prediction model using the spotify sessions database, which contains listening sessions and metadata of the songs in it. Our model predicts based on this initial session whether a user will skip the next song or not. For creating this model, we have first preprocessed the data by summarizing the session into unary variables. After this we apply decision trees to classify this data.**

I. Introduction

Automatic recommendations are getting increasingly popular with the digitalisation of music. They have an important role in the consumption of music nowadays. [something more about how important this is]

Predicting whether or not a user will skip a song is known as skip prediction. This is a good method measure how much a user likes this song. Thus it is often used in recommender systems, such as the playlist creator or autoplay from Spotify.

Spotify released a database containing information about user's listening sessions. [This database has nearly 130 million entires].

II. Background

Every record has two main parts: The listening session and the track. The goal of our model is to predict whether the track will be skipped based on the listening session and it's metadata.

A. Listening Sessions

The listening sessions contain the tracks in the order the user listened to. [They contain up to 20 tracks.] The first half of these sessions are used to predict whether the tracks in the second half will be skipped or not.

The session records have the following properties: Whether the user has Spotify Premium or not, the action causing the listening session to start, the date etc.

B. Tracks

The track entries contain data about their audio features, provided by the spotify API. Information such as the bounciness, dancability and key.

In addition, the duration, popularity and release year. A track is skipped if a user did not listen to the entire track. There are three metrics which measure this:

- `skip_1`: The track was only played very briefly
- `skip_2`: The track was only played briefly
- `skip_3`: Most of the track was played
- `not_skipped`: The track was played in its entirity

We use `skip_2` as ground-truth.

[something more about the data]

III. Methods

We first preprocess the data such that it can be used to train our model. After this we find the optimal parameters for our model using k-fold cross validation.

A. Preprocessing

In this section we discuss the features of every record which are used by our data mining model.

1) Track Features: As mentioned in the Background section, every track has 21 features containing it's audio features and duration, popularity and release year. We use the audio features without any any modification.

2) Session Skips: Our models do not take sequential data into account, thus we need to summarize the session skips into single features. These are the features of every track in the session indicating whether the user skipped this track or not. In order to use this for calculations we need to convert the skip to a number. We use `skip_2` as indicator of whether a user skipped or not in a track, just like the ground-truth. If `skip_2` is true, then our skip number equals 1, otherwise is 0.

In order to summarize this we create two features:

- `%_skip`: The percentage of skips. E.g. if a user has skipped 3 out of 10 tracks `%_skip` is equal to 30%
- `sd_skip`: The standard deviation of the skips. So if a user has skipped none of the tracks `sd_skip` is equal to 0. Or if a user has skipped 3/10 tracks the `sd_skip` is equal to 0.46. This is a measure of how irregular the user's skipping behavior is.

3) Session Accoustic Features: In order to summarize the accoustic data of the session tracks we create two features. One features is the average accoustic data of all the skipped tracks. The other feature is the average accoustic data of all the unskipped tracks. This allows us to see the average information of the preferenced and unpreferenced tracks.

B. Transforming other features

In order to make the feature more representative of the data we also adjust the following features:

[The date is rewritten to be one of the 7 days of the week instead of the absolute date.]

[Popularity, year of release and duration wil be kept the same.]

C. Applying Decision Trees

In order to train and apply our model, we use DecisionTreeClassificer from the python library scikit-learn.

D. Selecting parameters

In order to optimize the parameters for the decision trees we apply nested k-fold cross validation. K-fold cross validation is a method which given a k, splits the dataset into k groups. Then using one of the groups as training set and the others as test set, it measures the performance of the model.

We apply cross validation for the following parameter ranges: In total we trained 48 models to check which parameters are optimal.

- max depth: 10, 15, 20, 25
- min samples split: 15, 20, 30
- min weight fraction leaf: 0 1 3 5

This resulted in max depth = 15, min samples split = 30, min weight fraction leaf = 0.

IV. Results and Discussion

[results]

[what could have been better] -¿ [neural networks]

V. Conclusion

TODO

Acknowledgment

I would like to thank Tim Kersten, Ivo Melse and Daniël Mol for giving feedback on this paper.

References

[1] TODO