KKBox's Music Recommendation Analysis

Introduction

Music is an important part of the daily life for lots of the people. Currently music is a billion-dollar global industry due to the high demand of the public for music consumption.

In 2017, 38 percent of the music industry revenues worldwide was attributed to streaming sector. Big players as Google with Youtube Music, Apple with Apple Music, Spotify or KKBox are trying to get their piece of that cake. In this highly competitive environment, being able to satisfy users needs by implementing and optimizing music recommendation systems will be crucial.

Problem definition

KKBOX was founded in 2004, and it is leading the music streaming industry in Asia. Its wide Asia-Pop music library contains more than 30 million tracks. The service allows you to listen to your music playlists from all your devices and uses a recommendation system combining matrix factorization and word embedding methods.

Nowadays, user music preference could vary from classical music to the latest hit. Recommendation systems need to predict whether a person will enjoy a new artist or a new song. This is especially challenging when the listener recently joined the service, since there is not enough historical data. This project will focus on answering the following question:

- Will a user listen to a song again in less than a month after the first time that it was listened?

Data description

The dataset contains a sample of over 7 million listening events performed by more than 30,000 users. Each event corresponds to the first one performed for a particular user-song pair in the selected time period. Data also contains details about the user, such as registration time or age, and song metadata.

Data is available on Kaggle:
https://www.kaggle.com/c/kkbox-music-recommendation-challenge/data

The full description of each of the files can be found below:

train.csv (7.4m rows, 30.8k users)

- msno: user id
- song_id: song id
- source_system_tab: the name of the tab where the event was triggered. System tabs are used to categorize KKBOX mobile apps functions. For example, tab `my library` contains functions to manipulate the local storage, and tab `search` contains functions relating to search.
- source_screen_name: name of the layout a user sees.
- source_type: an entry point a user first plays music on mobile apps. An entry point could be `album`, `online-playlist`, `song` .. etc.
- target: this is the target variable. `target=1` means there are recurring listening event(s) triggered within a month after the user's very first observable listening event, `target=0` otherwise .

test.csv (2.6m rows)

- id: row id (will be used for submission)
- msno: user id
- song_id: song id
- source_system_tab: the name of the tab where the event was triggered. System tabs are used to categorize KKBOX mobile apps functions. For example, tab `my library` contains functions to manipulate the local storage, and tab `search` contains functions relating to search.
- source_screen_name: name of the layout a user sees.
- source_type: an entry point a user first plays music on mobile apps. An entry point could be `album`, `online-playlist`, `song` .. etc.

songs.csv (2.3m rows)

The songs. Note that data is in unicode.

- song_id
- song_length: in ms
- genre_ids: genre category. Some songs have multiple genres and they are separated by `|`
- artist_name
- composer
- lyricist
- language

members.csv (34.4k rows)

User information.

- msno
- city

- bd: age. Note: this column has outlier values, please use your judgement.
- gender
- registered_via: registration method
- registration_init_time: format `%Y%m%d`
- expiration_date: format `%Y%m%d`

song_extra_info.csv (2.3m rows)

- song_id
- song name - the name of the song.
- isrc - International Standard Recording Code, theoretically can be used as an identity of a song. However, what worth to note is, ISRCs generated from providers have not been officially verified; therefore the information in ISRC, such as country code and reference year, can be misleading/incorrect. Multiple songs could share one ISRC since a single recording could be re-published several times.

Solution approach

Some of the steps that will be performed are:

1. Load data and clean data.
2. Exploratory analysis, looking for interesting patterns in the data, and correlations between variables.
3. Look for the best algorithm to predict the chances of a user plays again a song.
4. Draw conclusions.

Deliverables

The following documents will be uploaded in a Github repository:

5. **Code** from the project.
6. A **final paper** explaining the problem, approach, and findings. The document will include ideas for further research, as well as up to 3 recommendations for the client on how to use the findings.
7. A **slide deck** to present the analysis to the clients or the general public.