Machine Learning Nanodegree

Capstone Proposal

Yuanqi Wang

December 22, 2017

I Domain Background

This project is based on the "WSDM - KKBox's Music Recommendation Challenge" from Kaggle.[1] The aim of this project is to build a better music recommendation system for KKBox, a music streaming service in Asia. KKBox currently employs "a collaborative filtering based algorithm with matrix factorization and word embedding in their recommendation system."[2]

In this particular project, we aim to improve KKbox's recommendation system by improving the ability to predict how likely a user is going to listen a particular song *again* during certain time window. Since the task is to classify whether a user is going to listen a song again or not, this is in essence, a binary classification problem.

Classification problems fall into the bigger umbrella of supervised learning — perhaps one of most common types of Machine Learning. Classification algorithm has wide applications, such as spam filtering, fraud detection, character detection, just to name a few. Some of the algorithms for tackling a classification problem include logistic regression, decision tree, support vector machine, neural network, and ensemble methods such as random forest, boosting, etc.

Even though classification algorithms have been studied extensively and used widely in industry, new research that improve on existing methods and explore new algorithms keep this active and thriving domain.[4] The decision of choosing a particular algorithm is often based on the specific requirement and constraints of

[1] <www.kaggle.com/c/kkbox-music-recommendation-challenge>

[2] <www.kaggle.com/c/kkbox-music-recommendation-challenge> A recommender system is "a technology that is deployed in the environment where items (products, movies, events, articles) are to be recommended to users (customers, visitors, app users, readers) or the opposite."[3] Loosely speaking, there are two types of recommendation system: content based and collaborative-based. See <www.towardsdatascience.com/how-to-build-a-simple-song-recommender-296fcbc8c85> Content-based system recommends item based on what the history of a particular user, while collaborative-based system recommends item to a user based on what other similar users liked.See <www.analyticsvidhya.com/blog/2015/10/recommendation-engines/>

[4] Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." (2007): 3-24.

the problem at hand (e.g. speed of classification, tolerance to noise, the number of features and instances, etc.)

II Problem Statement

As stated earlier, this is a binary classification problem. The problem is to predict how likely a particular user listens to a particular song for more than one time.

III Datasets and Inputs

The datasets are obtained from Kaggle. It includes five separate files that contain information about the users, songs, and the system environment when the event is triggered:[5]

1. `train:`
   - `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.
   - `source_screen_name:` the name of the layout a user sees.
   - `source_type:` the entry point a user first plays music on mobile apps. An entry point could be album, online-playlist, song... etc.
   - `target:` target variable. target=1 means there are recurring listening event(s) triggered within a month after the users very first observable listening event, target=0 otherwise.
2. `members:`

[5] See: www.kaggle.com/c/kkbox-music-recommendation-challenge/data

- `msno`: user id (for merging)
- `city`
- `age`
- `gender`
- `registration method`
- `registration_init_time`: format %Y%m%d
- `expiration_date`: format %Y%m%d

3. `songs: song id, song length, genre ids, artist name, composer, lyricist, language`

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

4. `song_extra_info: song id, song name, international standard recording code`

    - `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 infor-

mation 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.

The number of observations in the training set is 7,377,418. After merging the `train, members, songs,` and `song_extra_info`, there are a total of 19 features (excluding `target`). These features could be informative in calculating the likelihood of a song being listened again by a user. Certain variables are categorical variables (e.g. genre_ids, , etcs), therefore they need to be transformed through one-hot encoding. Several variables have relatively large amount of missing observations (such as gender, lyricist, etc.), more data exploration will shed light on how these missing values should be treated. Finally, certain variable contains information that needs to be extracted. For example, isrc contains which country the song is published in, and which year it was released. Extracting such information from existing variables will also create a richer dataset.

IV Solution Statement

I plan to test and compare a variety of algorithm including SVM, Logistic Regression, Random Forests, Boosting, as well as Neural Networks (inspired by the discussion on Kaggle forum).

V Benchmark Model

I will use Logistic Regression without any feature engineering as the Benchmark Model.

VI Evaluation Metrics

The evaluation metrics is "area under the ROC curve between the predicted probability and the observed target." or "AUC".[6] Therefore, it is the rank of the probability that matters.

VII Project Design

This will be an iterative process:

1. Combine datasets
2. Exploratory data analysis (dealing with missing data, observe distribution, etc.)
3. Feature engineering
4. Set up Cross Validation
5. Train using different classifiers; User GridSearch technique when applicable.
6. Compare results (AUC) from different classifiers.
7. Identify the most promising classifier.

[6] <www.kaggle.com/c/kkbox-music-recommendation-challenge#evaluation>