Machine Learning Engineer Nanodegree

Capstone Proposal

Shubham Gupta
October 24th, 2017

Proposal

Domain Background

Not many years ago, it was inconceivable that the same person would listen to the Beatles, Vivaldi, and Lady Gaga on their morning commute. But, the glory days of Radio DJs have passed, and musical gatekeepers have been replaced with personalizing algorithms and unlimited streaming services.

While the public's now listening to all kinds of music, algorithms still struggle in key areas. Without enough historical data, how would an algorithm know if listeners will like a new song or a new artist. And, how would it know what songs to recommend to its brand new users. Since content recommendation is at the heart of most subscription-based media stream platforms. Hence a good recommendation system can vastly enhance user experience and increase user engagement.

Problem Statement

The goal of the project is to exploit the provided user history and music metadata to develop a recommendation system. Given a copy of user listening history, the aim of the project is to predict what songs a set of predetermined users would listen to during the next month. The dataset is from KKBOX, Asia's leading music streaming service, holding the world's most comprehensive Asia-Pop music library with over 30 million tracks. They currently use a collaborative filtering based algorithm with matrix factorization and word embedding in their recommendation system but believe new techniques could lead to better results.

Datasets and Inputs

The data is provided by the WSDM-KKBox Music Recommendation Challenge on Kaggle. KKBOX provides a training data set consists of information of the first observable listening event for each unique user-song pair within a specific time duration. Metadata of each unique user and song pair is also provided.

Tables

train.csv

- 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 only present in train.csv. 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

- id: row id (will be used for submission, only present in test.csv)
- The remaining columns of the test.csv are same as the train.csv, except the test.csv does not contain 'target' variable.

songs.csv

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

- 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

Solution Statement

The solution to the problem is to correctly predict the chances that a user will listen to a song again within a month after the user's very first observable listening event. First, I will use various data exploration and visualization techniques using seaborn and matplotlib libraries to find any relation between features in same and different csv files. Secondly, I will perform feature selection and combine features from songs.csv and members.csv with the features in train and test csv files. Lastly, I will perform one-hot encoding on categorical features and normalisation on numerical features.

Since this is a binary classification problem, hence I will use Light Gradient Boost Model (Light GBM) and XGBoost to train the models. Finally, I will tune various parameters of these models and use cross validation to decide which model performs best.

Benchmark Model

For this problem, the benchmark score set by the WSDM-KKBox Music Recommendation Challenge is 0.60643. I will use the two algorithms to generate predictions that can beat this score.

Evaluation Metrics

The predictions are evaluated on area under the ROC (AUROC) curve between the predicted probability and the observed target. The ROC (Receiver Operating Characteristic) curve is created by plotting the true positive rate (TPR) or recall against the false positive rate (FPR) or probability of false alarm at various threshold settings. A ROC space is defined by FPR and TPR as x and y axes, respectively, which depicts relative trade-offs between true positive and false positive. The best possible prediction method would yield a point in the upper left corner or coordinate (0,1) of the ROC space, representing 100% sensitivity (no false negatives) and 100% specificity (no false positives).

Since this project is based on the WSDM Kaggle competition, I will take the leaderboard score as my evaluation.

Project Design

Firstly, I will read all the four (train, test, songs, members) csv files. Then I will take a glimpse of the shape of the data and the data types stored in these csv files. Further on I will search for any null values in the data. Now, before I step towards any preprocessing of the data, I will visualize the data in each of the csv files separately to find relations between the features in them. Then I will perform visualizations between the features train, songs and members csv files and find if there is any correlation between them.

Next, step would be data extraction and feature formation. In this step I will extract the registration dates and expiration dates (stored in YY%mm%dd format) and store them in separate columns. Further, I will merge various columns of songs and members csv files with the columns in train and test csv files. Thus our training data will contain information from the songs and members csv files. Now, if any correlation is found between different features during visualization, then we can combine those features to form new features for the training dataset.

The next step would be preprocessing of data. Firstly, if any null values are found in our training dataset, then we will replace them with a constant for categorical features and mean for numerical features. Then we will search for any outliers in the training dataset and remove them. Lastly, we will perform either one-hot encoding or label encoding, whichever suits best, to our training dataset.

Finally to train our model I will be using two algorithms, Light GBM and XgBoost and tweaking their respective parameters to gain maximum auc score. But before we train the model I will generate a validation data from our training dataset. This is essential so that we can check how well the model is performing (locally) by calculating the auc score on the validation dataset. While the final evaluation of the model will be made by score generated by the test dataset on the leaderboard of the Kaggle competition page. The selection of the best model would be based on the score achieved by them on the competition leaderboard.

References

1. https://www.kaggle.com/c/kkbox-music-recommendation-challenge#description
2. https://en.wikipedia.org/wiki/Receiver_operating_characteristic
3. http://www.wsdm-conference.org/2018/call-for-participants.html