CS 209a Final Project: Spotify
Scope of Work & Preliminary EDA

Group Number

We are **Group 44**. Group members (alphabetical) are:

- Anna Davydova
- Michael S. Emanuel
- Avriel Epps
- Erin Williams

Group Communications

Our group has met in person twice. We have set up a team Slack channel for real time communications. We have established a GitHub organization for sharing source code, and created a repository for this project. We have also created a shared folder on Dropbox so the team can have synchronized access to large data files that are not suitable for GitHub. Our members live off campus and our numbers include three parents, so we've engineered our team workflows to allow us to collaborate efficiently when working from home.

TF Communications

Our group has been in contact with our TF, Will Claybaugh. We had an initial contact with him when we discovered that the link to the Spotify Million Playlist Dataset had expired because the underlying competition ended on June 30, 2018. Will kindly provided us a link to an archived copy of the data. We are in the process of scheduling an in person meeting with Will that is targeted to last 15-30 minutes. That meeting is intended to be an informal and interactive session. We will bounce our current plans off him, and get his advice about possible adjustments or pitfalls in our strategy.

Project Questions

According to the Project Description, the primary goal of the project is Automatic Playlist Generation—creating a model to discover given the base playlist and the additional playlist data (e.g. title, number of followers). Using the Million Playlist Dataset, we will attempt to answer the following concrete question: Given the playlist metadata (e.g. title, etc.) and every track *but the last one*, how well can we predict the next track?

In particular, we would like to make three increasingly difficult predictions: the *artist*, the *album*, and finally the *track*. Each of these predictions would be output by the model as a set of probabilities. We will score the predictions on the basis of their cross-entropy loss. We would also compute an "uninformed" model that had no information at all. This model would just guess the population average frequency for artists, albums, and tracks. We will then compute a dimensionless *entropy ratio* of the entropy of our predictions divided by the entropy of the uninformed model. A lower ratio would be better; zero would mean perfect accuracy on every prediction, one would mean no added information compared to the uninformed model. We will compute a blended loss function using the entropy loss ratio for the predictions of the artist, album and track. Proposed weights will be 30% for the artist, 20% for the album, and 50% for the track.

While the model will be trained using cross-entropy loss, we will also score the model based on its classification accuracy for artist, album and track. We might score its accuracy not only on the first guess, but also allow the model to make up to 3 or 5 guesses, and compute its accuracy on that basis. The rationale for this is that a recommender system that gave you 3-5 choices for the next track could provide a useful user experience. We will report these scores and highlight them in the model assessment we publish. Cross entropy loss is an excellent metric for model training, but end users (e.g. executives at Spotify) want to know the accuracy of these forecasts.

We may investigate a variant in which we attempt to make conditional forecasts. In particular, we may try to predict the album of the next track given the artist, and the track given the album. There are two reasons for this. First, music tracks have a natural structure with artists, albums, and tracks, and taking advantage of this structure may help improve our forecasts. Second and more importantly, the underlying business use case involves creating a recommendation for users. While users may prefer a recommendation for the next track that can play automatically, some my be interested in sequential recommendations. A system configured this way could give a user five sorted choices for the next artist, followed by five choices for the album given their artist selection, and finally five choices for the next track.

For the Cold Start problem, we can develop a parallel version of our model where both training and testing are limited to a universe of short playlists. We will make a useful definition of a short playlist on the basis of a histogram of playlist lengths from the data and our business judgment about what makes sense for a company such as Spotify. We can augment our data and create additional data points by truncating a longer playlist early, e.g. we look at just the first 8 tracks and try to predict the 9th one.

If time permits, we may also investigate the Playlist Popularity problem. This question asks us to predict the popularity of a playlist based on its contents. We would define the popularity as the log of the number of followers. We will probably limit the data to playlists that have been around long enough to attract a full base of followers. The cutoff threshold for how old a playlist must be before we consider it to be “seasoned” would be based on the data. The rationale for this exercise is that Spotify might use a tool like this to promote user generated or curated playlists to its user base. Of course it’s easy to know if a mature playlist is popular—just see how many followers it has! This would allow Spotify to predict how many followers a new playlist might attract over its life.

Action Plan for Milestone Three

The tasks and deadlines below are preliminary and speculative. It is NOT expected that this is what will actually happen! This is rather intended to be the first iteration of a work plan that will evolve if necessary. We should assess at least once a week if we're on track with the plan, adjust it as necessary, and ensure that we're on track to product the Milestone 3 deliverables.

Data Collection (Oct. 25)

- Download Million Songs Dataset to team Dropbox. (**MSE, Oct. 16**)
- Download Million Playlist Dataset to team Dropbox.
    - Unpack the data. (**MSE, Oct. 18**)
    - Port Python utilities with the mpd to Python 3. (**MSE, Oct. 18**)
- Investigate the Spotify API; how do we access it? Which audio features are available? Do we want to download the audio features and ISRC numbers for our tracks?
    - Report; team decision on whether to download: (**XYZ, Oct. 23**)
    - Download data (**XYZ, Oct. 25**)
- Investigate Lyrics Wiki—can we link song lyrics using our available track data? (**XYZ, Oct. 25**)

Exploratory Data Analysis (Nov. 12)

- Cumulative frequency of artists, albums, and tracks: plot a curve showing the cumulative popularity of these three entities in descending order. Popularity measured by the sum of the number of followers on each track. Popularity of albums and artists is an aggregation of track popularity. Recommend a popularity threshold for us to limit the number of artists, albums, and tracks we consider in our models. (**XYZ, Oct. 28**)
- Additional Preliminary EDA on MPD (**XYZ, Oct. 28**):
    - Histograms of playlist length by number of tracks and in minutes. How long are these playlists?
    - Histogram of the number of followers. What is the distribution?
    - Explore most common track titles. How much do they cover? Can we glean useful information from the track titles?
- Preliminary assessment about Genre. Can we assign useful genre tags to artists, albums or tracks? How (e.g. ISRC numbers linking to another dataset)? Is it worth the effort? (**XYZ, Oct. 28**)
- Develop preliminary EDA strategy: Based on above work propose to rest of the team an EDA strategy, with additional EDA tasks as necessary. (**XYZ and ABC, Oct 30**).
- Carry out additional EDA tasks—assigned after above; completed by **Nov. 12**

***Baseline Model** (**Nov. 14**)*

- Python code to create and persist data frames for the following logical entities: artists, songs, tracks, playlists. Objective is to have two data frames for each type: a small one for quick exploration, and the full sized one. (MSE Oct. 28)
- Develop the uninformative model and cross entropy loss function; compute prediction accuracy for next 1, next 3, next 5. (MSE Oct. 30)

- First pass at predicting the artist, album, and track using “simple” techniques, e.g. a softmax classifier with linear weights of available features. (**MSE Nov. 12**)
- Review and critical feedback of Baseline Model; should be provided informally on an ongoing basis as code is posted to GitHub. More formal presentation of feedback, both an assessment on quality and suggestions about future work: (**XYZ [and ABC] , Nov. 14**)

Milestone 3 Deliverables (Nov. 26)

- Description of Data for Milestone 3(**XYZ, Nov. 24**)
- Exploratory Data Analysis Findings for Milestone 3 (**XYZ, Nov. 24**)
- Preliminary Milestone 3 merging above and making one round of edits (**XYZ, Nov. 25**)
- Feedback / suggested changes before Milestone 3 deadline (**Everyone, Nov. 26**)
- Final version of Milestone 3 and submission on Canvas (**XYZ, Nov. 27**)