What is the Best Song to Recommend?

Liam Hayes

Department of Mathematics, Colorado State University, MATH 331

Motivation

Spotify and other streaming services are much more than digital libraries of music. Music streaming services today function as personal DJs and a music discovery platform, using models to create playlists and recommend music that users are likely to enjoy.

Platforms that do this well create value that benefits all spaces that music touches. Listeners will enjoy their music more and find songs that they might otherwise not hear. Artists also benefit because their music will reach more people that are likely to enjoy their music.

Luckily, there are a lot of available data that supports building these models. For example, Spotify decided to crowd source this problem by publicly providing data about listening behavior and songs with song attributes. This is the data we will be using.

Methodology

First, we classify types of songs. We want to do this because it simplifies the prediction process. We can fit a lot of information about a song in a predictor with song classifications. To do this we are using k-means classification, an unsupervised approach because we don't know exactly what groups we want the songs in.

Then we can summarize the session based on song classification, summary statistics about the songs that they have listened to previously in the session, and other metrics that describe the session like number of songs they have skipped and number of songs they have listened to. We are also using these as predictors because different types of sessions can influence what types of songs the user is in the mood for and how likely they are to skip any given song.

Some more predictors describing the current song are also used on top of the song classifications. Things like tempo, release year, energy, length, key, and mode can contain information that our song classifications might not be able to account for.

Finally, we fit different classification trees using boosting and random forests to predict whether a song will be skipped or not.

Improvements

A different way to represent key and mode can make these predictors less noisy. For example, a song in the key of C major tends to sound better when played after

1. Another song in the key of C major or minor
2. A song in the key of F major
3. A song in the key of G major or minor
4. ...

Using music theory knowledge to represent key and mode in a relative way (think circle of fifths) can make this model much better at recognizing what songs will sound better to the user at that point in time.

We could also try to use unsupervised learning methods to group listeners based on their listening history outside of the current session. Also group sessions based on session data. Having these groups can reduce bias by accounting for different types of listeners and different types of sessions.

Skip Prediction with Trees

We fit two methods of tree-based models: a random forest and a boosted model. The boosted model ended up performing better and took less resources to run.

Boosting is a technique to reduce bias while maintaining a low variance. In the context of decision trees, we will:

1. Grow a low variance high bias tree (small decision tree)
2. Compute residuals for that tree
3. Fit a new low variance high bias tree to the residuals
4. Add the new tree to the previous tree
5. Repeat until the bias-variance trade off is optimal

As opposed to other tree based methods, boosting learns very slowly to avoid over-fitting.

There are three tuning parameters to optimize a boosted tree.

- Number of trees (B)
- Amount of shrinkage for each tree (λ)
- Number of splits in each tree (d)

Chosen Tuning Parameters

$$B = 1000$$

$$\lambda = 0.24$$

$$d = 1$$

Results

The error rate for this model was 0.1977. This means we predicted correctly 80.22% of the time.

		Truth	
		False	True
Guess	False	2146	611
	True	1594	6800

Table 1. Confusion Matrix

We can see that when the user does not skip, our error rate is much higher (42.6%). When the user does skip, our error rate is 8.2%.

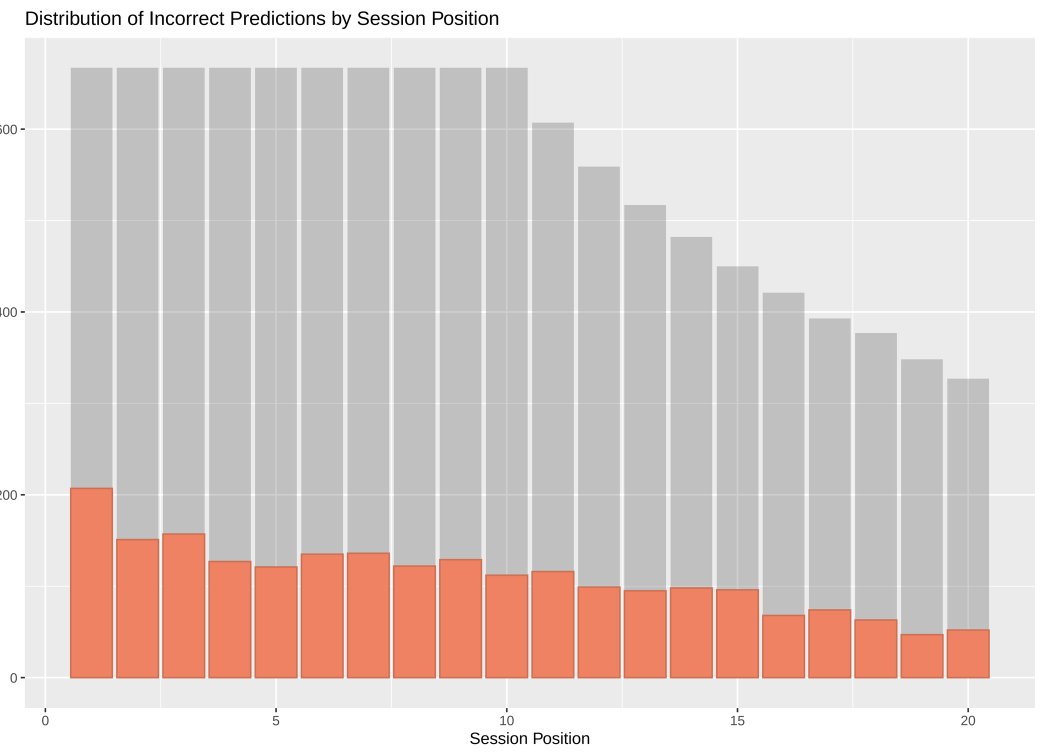

Figure 1. Error rate by session position

Song Classification

K-Means Clustering

K-means clustering is an unsupervised method to group similar data together. Since we don't have genre labels, we can create our own arbitrary "genres" with k-means clustering.

This method works by grouping the data into k groups and minimizing within-group sum of squares. In other words, it minimizes the distance from the mean of a given group to all data points in that group for all groups.

k is a tuning parameter that the modeler chooses. This decides the number of groups.

Clustering Results

We decided to use $k = 13$ groups.

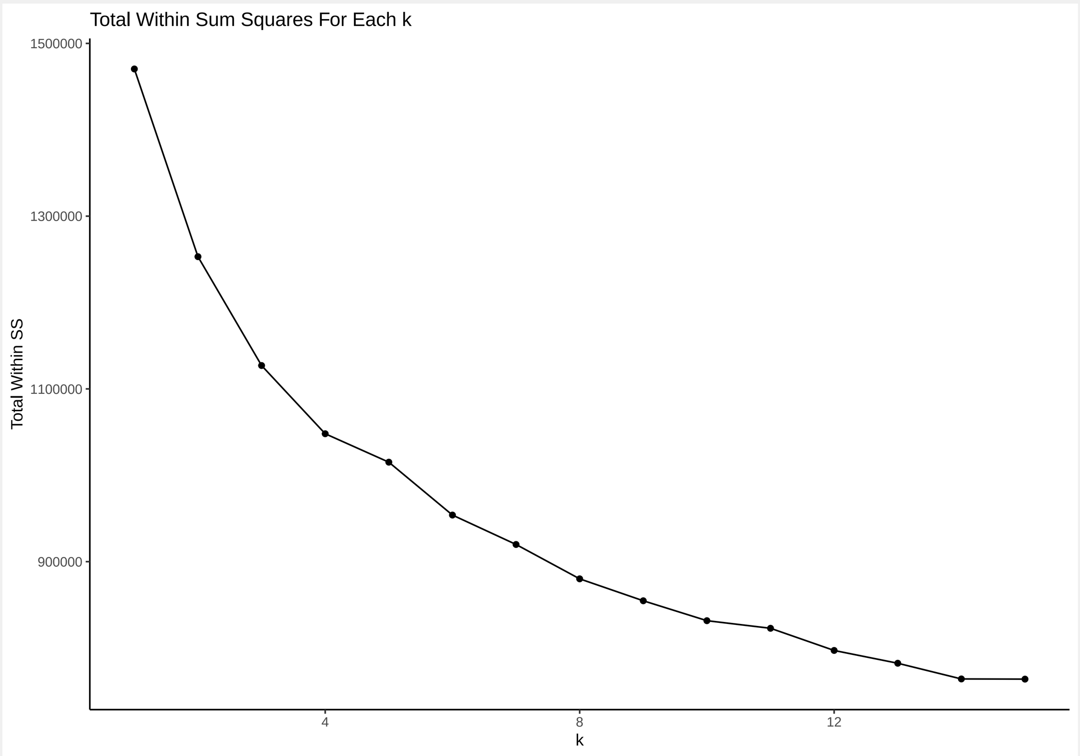

Figure 2. Within SS for values of k

Cross Validation

Cross validation is an essential technique for fitting models. In order to asses the accuracy of any given model, we need to test on data that is **out of sample**. This is because a trained model learns the patterns in the training data set. If the model learns too many patterns from the training data not present in the data we want to predict on, the model is over-fit and will not perform well on out of sample data.

To remedy this, we use k-fold cross validation to test the model.

1. Partition data into k equal parts
2. Train the model on $k-1$ parts
3. Compute some metric to evaluate the model on the left out part
4. Repeat, leaving out a different part each time
5. Compute an overall metric from the metrics in step 3

References

Brost, B., Mehrotra, R., & Jehan, T. (2019). The music streaming sessions dataset, *Proceedings of the 2019 Web Conference*, ACM.