Conducere

Inferring Music Preferences with Machine Learning Techniques

Megan Hopp {hoppm@cs}
Connor Moore {moorec22@cs}
Naomi Musgrave {naomi.g.musgrave@gmail}
Svetlana Grabar {grabar.svetlana@gmail}

Data & Collection

Data

Goal: Collect enough data for proof of concept for machine learning methods.

Call to Action: Participants were asked to volunteer music data in the form of playlists that best represent their current listening tastes.

Results: Ultimately, the playlists collected for training varied. While some comprehensively represented a wide range of a user's listening habits, others were more specific niches of a user's mood, with names like "*quiet yearning*" and "*Emerald City Vibes*."

Collection & Feature Extraction

To collect training data, we harvested playlists using the Spotify API. We then used the EchoNest API to extract 9 key music features:

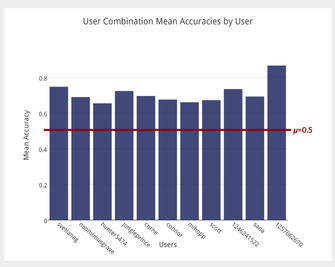
- danceability
- energy
- liveness
- loudness
- speechiness
- tempo
- valence
- instrumentalness
- acousticness

Random Forests

Random Forest Model

A random forest classifier fits a number of decision tree classifiers and uses averaging to improve the predictive accuracy. We used RFs for multi-class classification, testing different sizes of the forest, along with many different parameters for fine-tuning.

Forest Size	Accuracy
1	0.158
10	0.226
50	0.237
250	0.243
1000	0.257



Most Distinct Music Taste

The graph shows the results of user matchings in the form of average accuracies by user.

Binary Classification for User Combinations

Ran RF for every user matching in our full data set, so there were at most 2 classes to predict. Accuracies closer to 50% indicate similar music tastes, while higher accuracies indicate *more distinct* music tastes.

Accuracies with Combinations	
Baseline (random guess)	0.500
Min Reported	0.534
Max Reported	0.937
Average	0.711

Clustering

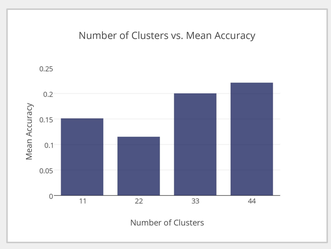
Clustering is an unsupervised learning model that attempts to separate n-dimensional data into distinct clusters. We used two clustering models, and then ran analyses over a mixture of two.

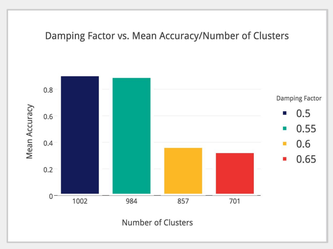
K-Means

K-means will search for exactly k clusters to split. We tried a number of clusters, all multiples of the number of playlists.

Affinity Propagation

Affinity propagation chooses clusters iteratively, without limiting the number of clusters. The effects of overfitting are pretty clear.

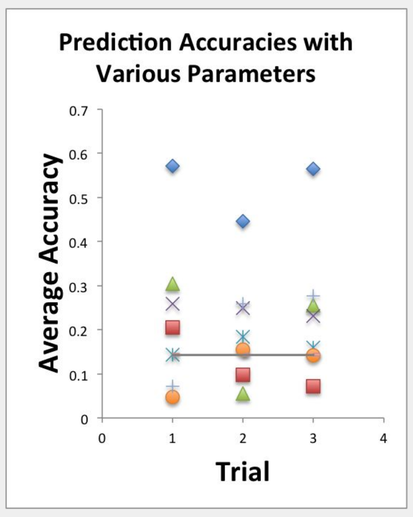




Combining these two models, by calculating 0.7 * (k-means) + 0.3 * (affinity), yielded a mean of 0.400, with dampened effects of overfitting.

Neural Networks

An unsupervised neural network learns a feature space describing the songs. This model is provided to a logistic regression classifier. The classifier maps a data point in n-dimensional feature space to a value between 0 and 1, and transforms this value into a label based on learned thresholds.



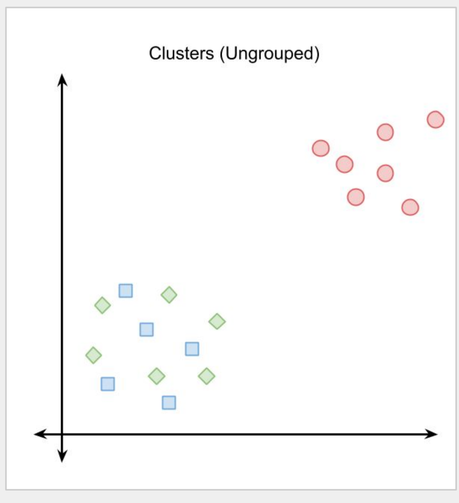
We trained the models on a variety of parameters and subsets of song features, with similar results. On average, the neural network and classifier yielded 18% - 20% accuracy (as opposed to 14% random guessing accuracy). The best performance was seen when sampling an even number of songs from each user, and from users with the most homogenous songs.

Conclusions & Future Work

All of these models had mixed results, when faced with similar music libraries. This is to be expected in classification models, but this data shows clearly that similarity in music tastes needs to be considered a bit more, maybe as strongly as classification. To that end, it would be worthwhile to consider a model that looks for similarity measurements, like the binary classification in Random Forests.

For example, say we wanted to look at clustering techniques that use similarity measurements to enhance results. The graph to the right is a pretty simple example of clustering data, where *triangle* and *square* have very similar taste based on the chosen features.

This example won't do so well, since *triangle* and *square* will end up in the same cluster and bring the accuracy down drastically.



But in our analysis, similarity in music taste should be a strength. So we combine similar tastes, and get a much better classification. Now we can be fairly sure that the two clusters are distinct. Any music that gets classified in the combined clusters can be further analyzed, or just classified in multiple music tastes.