Musical Genre Classification

Christina Sousa
Fernando Anorve

The Ohio State University

December 2, 2019

STAT 6750 Final Presentation

Objective

- Investigate different statistical classification methods applied to data from The Free Music Archive (FMA) (Defferrard et al., 2016)
  - $k-$Nearest Neighbors
  - Naive Bayes
  - Random Forest
  - XGBoost (Boosted Random Forest)
- Predict the musical genre in a subset of this dataset
- The subset of pre-processed audio features is currently available through the repository `https://github.com/mdeff/fma`

Data

Four tables of pre-processed audio features:

- `tracks.csv`: Metadata of each track, including ID, title, genres, and tags.
- `genres.csv`: All 163 genre IDs
- `features.csv`: common features extracted with *LibROSA*, python package for music and audio analysis.
- `echonest.csv`, audio and social features provided by Echonest (now Spotify) for a subset of 13,129 tracks.

Data

- We restrict our attention to $n = 13,129$ records for which both *LibROSA* and *Echonest* features are available.
- $p = 544$ potential predictors:
  - 13 are basic **song and artist features** (e.g *album title*, *artist location*, *duration*)
  - 12 are Echonest **social and audio features** (e.g. *danceability*, *acousticness*, *hotttnesss*)
  - 519 are various **signal processing features** extracted from frequency and time domains of raw audio *(LibROSA)*.
  - Target is 'track.genre_top'. 9 genres.
- (FMA paper)

Data

- Goal: Build an **interpretable** model that predicts well.
- Hence we focus mostly on the **song and artist features** and **social and audio features**.
- Train-validation-test split: 70%-15%-15%.
- Genres *Experimental*, *Instrumental*, and *Blues* did not have sufficient training data.
- Records with missing *top_genre* were also dropped.

EDA & Feature Selection

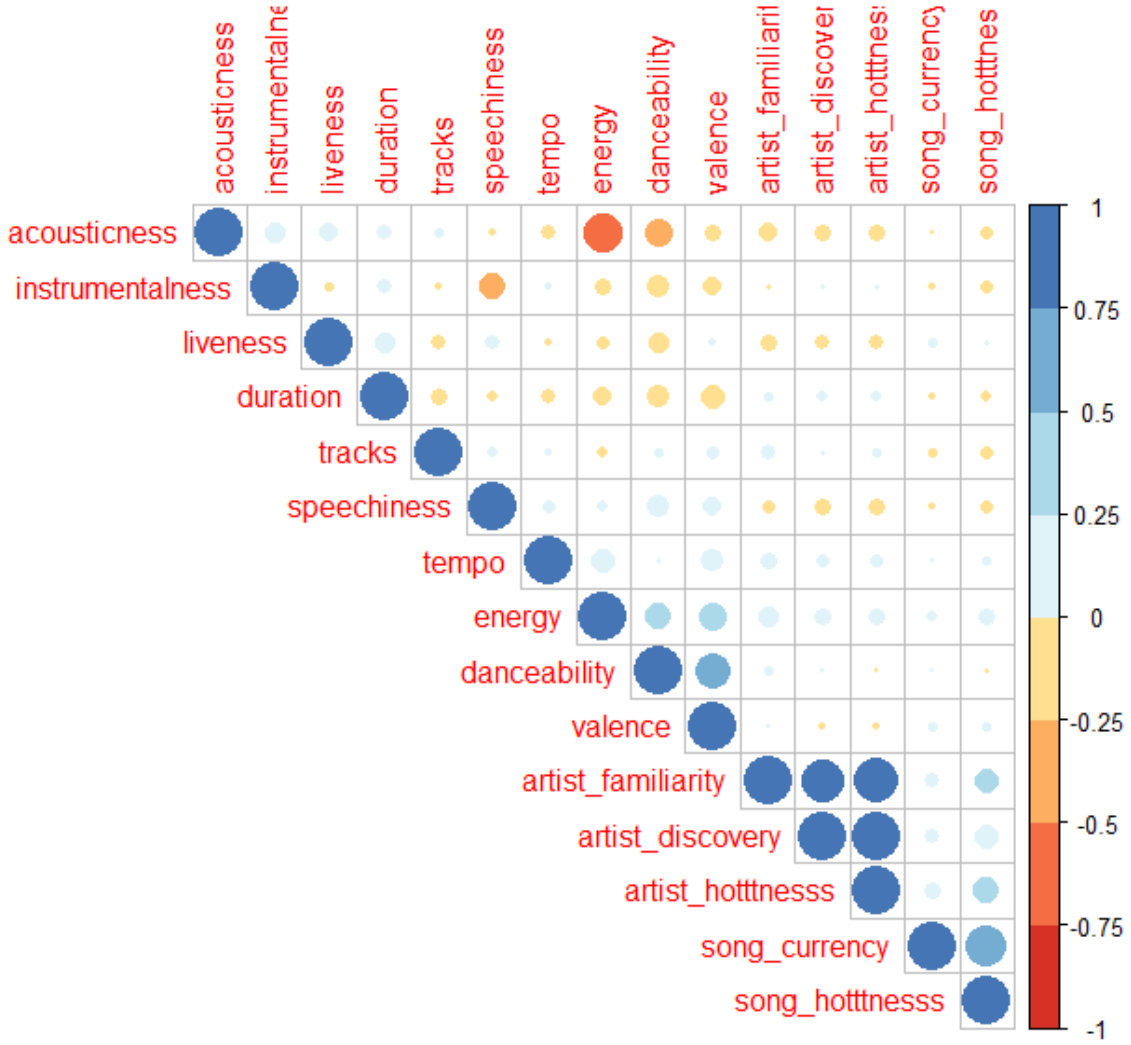
- Perform 532 Chi-Squared tests for each of the Echonest and LibRosa features
  - Continuous variables binned into three bins each (maintains minimum cell counts).
  - Use Bonferonni adjustment
  - Large data implies very small p-values
- Hope to identify useful predictors
- Sort results by feature group

EDA & Feature Selection

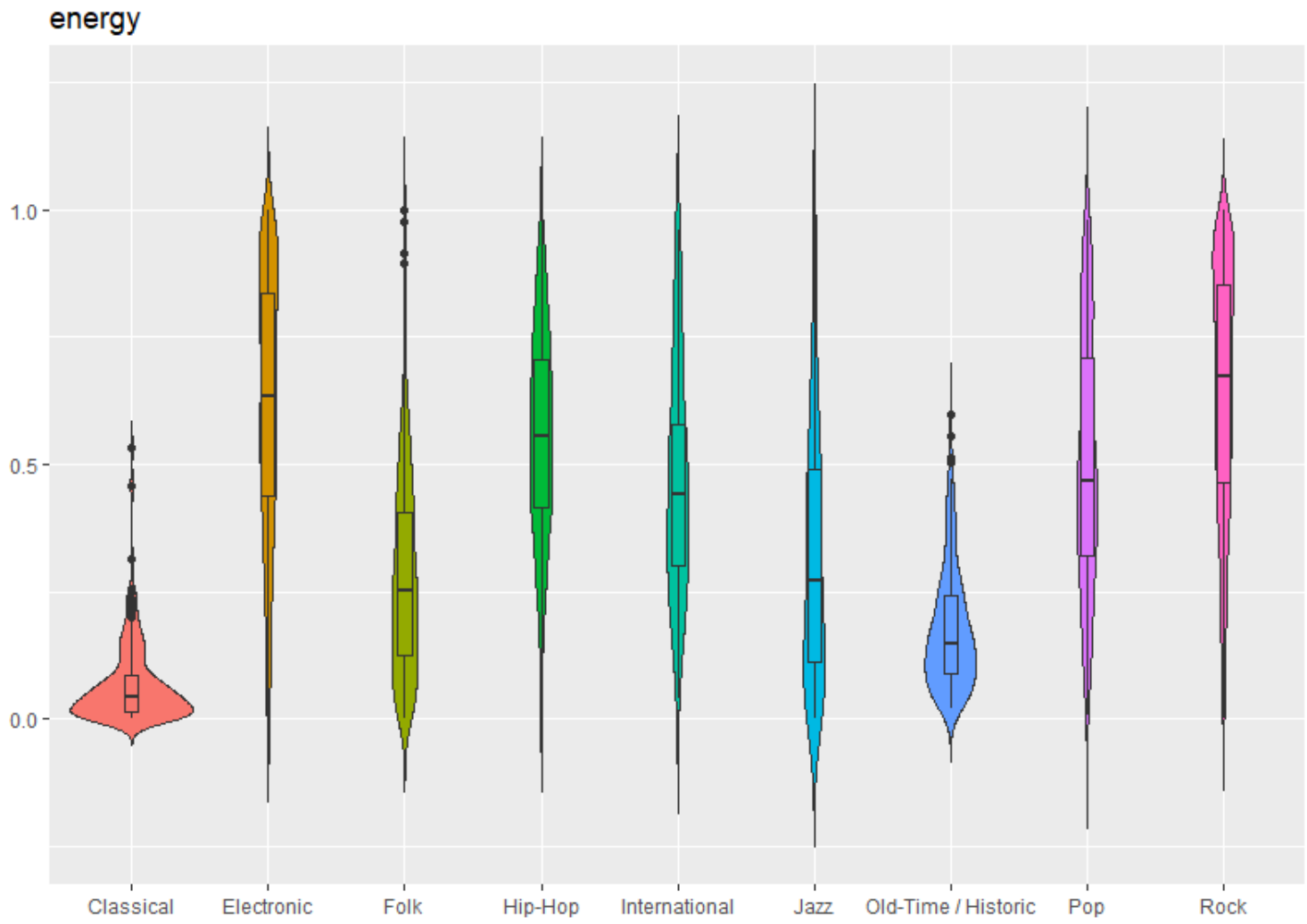
Variable	p-adj
audio_features.energy	0
audio_features.danceability	2.98E-300
audio_features.acousticness	1.75E-289
audio_features.speechiness	1.23E-161
audio_features.instrumentalness	1.24E-149

Variable	p-adj
mfcc.max.01	0
mfcc.max.03	0
mfcc.max.04	0
mfcc.max.06	0
mfcc.mean.01	0
mfcc.mean.04	0
mfcc.mean.06	0
mfcc.mean.07	0
mfcc.mean.08	0
mfcc.mean.09	0
mfcc.mean.10	0
mfcc.mean.12	0
mfcc.median.01	0
mfcc.median.04	0
mfcc.median.06	0
mfcc.median.07	0
mfcc.median.08	0
mfcc.median.09	0
mfcc.median.10	0
mfcc.median.12	0
spectralcontrast.max.07	0
spectralcontrast.mean.02	0
spectralcontrast.mean.04	0
spectralcontrast.mean.05	0
spectralcontrast.median.04	0
spectralcontrast.median.05	0
spectralrolloff.skew.01	0
[illegible]	[illegible]

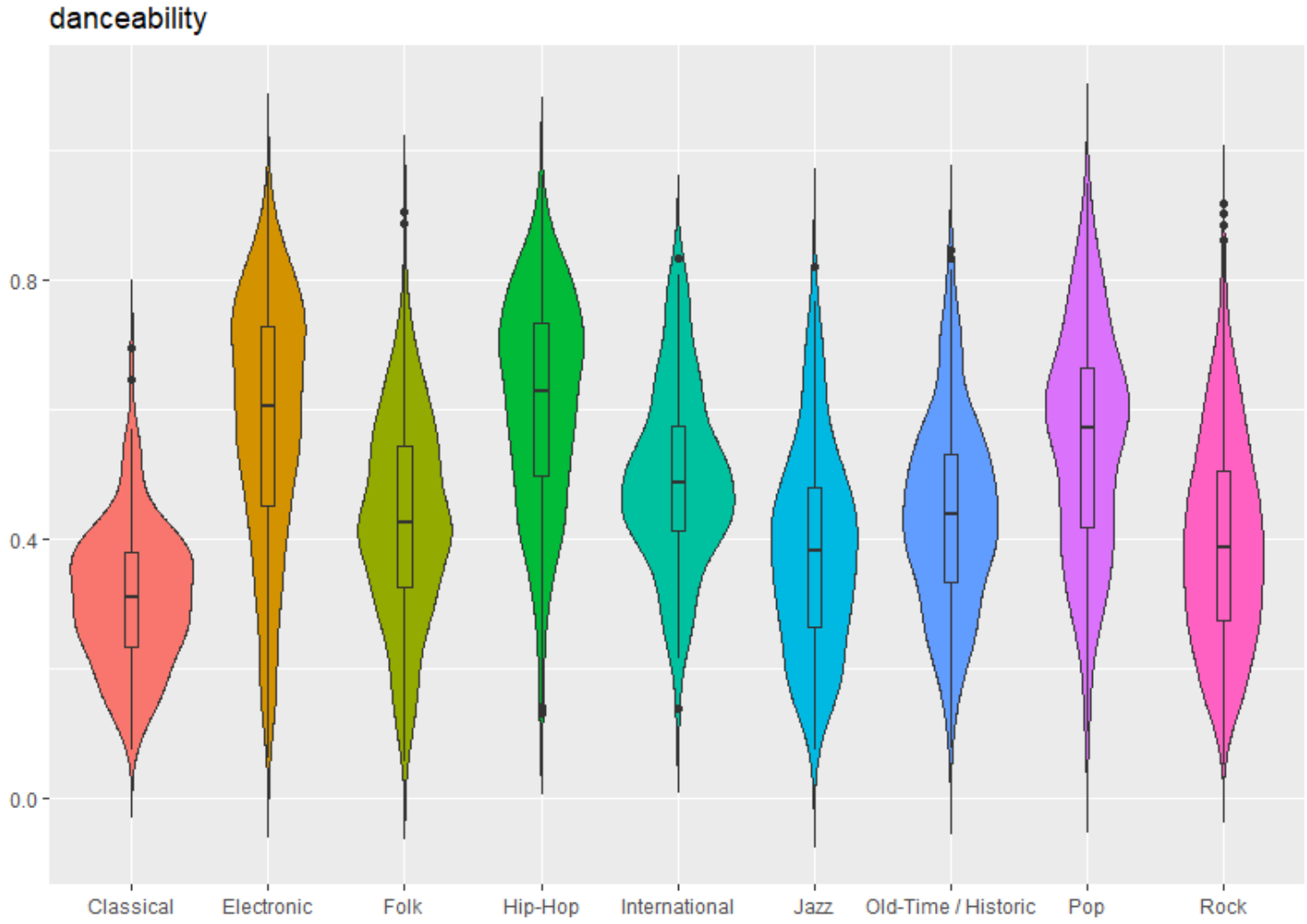
EDA & Feature Selection



EDA & Feature Selection



EDA & Feature Selection

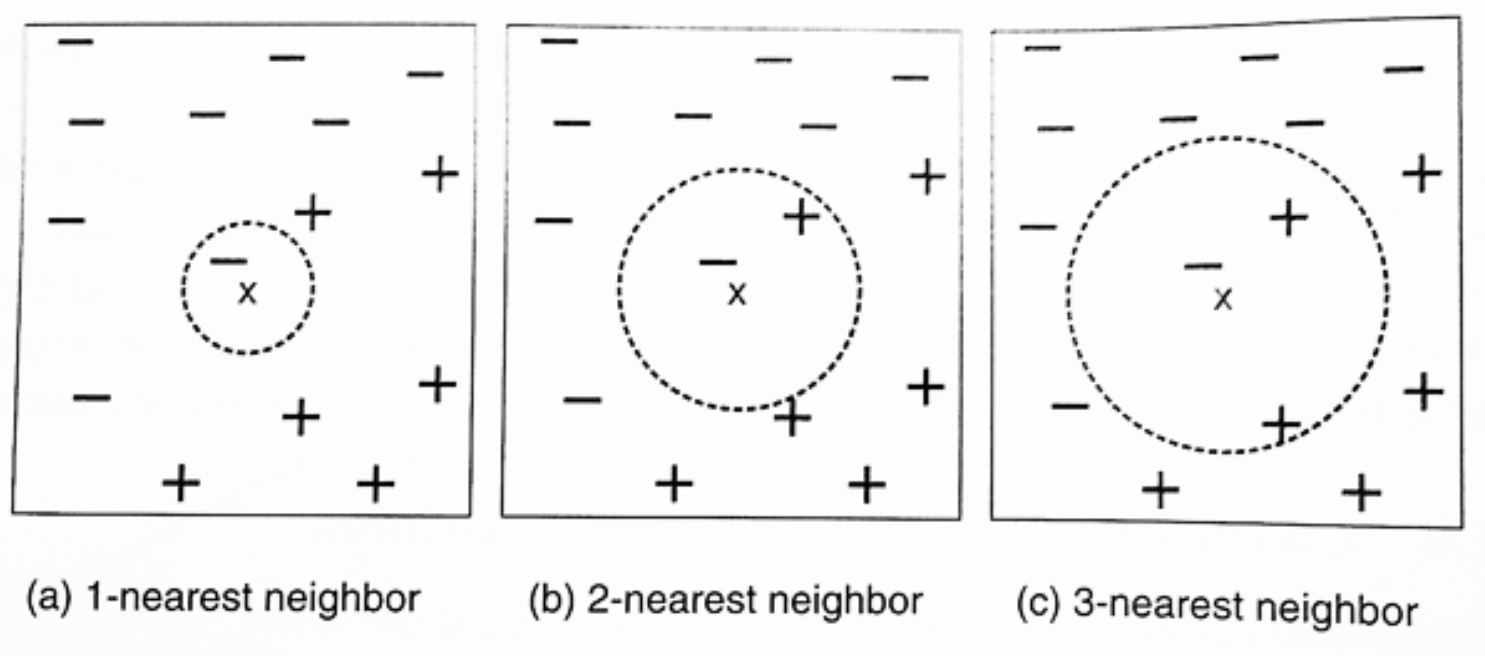


K-Nearest Neighbors

k Nearest Neighbors

Idea: similar data points will have similar labels.

The algorithm combines the k-nearest neighbors' labels to determine the label of the testing data point through a majority vote.



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

Results

k Nearest Neighbors											
		Reference									
		Classical	Electroni	Folk	Hip-Hop	Internati	Jazz	Old-Time	Pop	Rock	
Predicted	Classical	25	0	11	0	0	1	1	0	2	63%
	Electroni	4	182	9	18	0	1	0	1	111	56%
	Folk	14	7	35	7	1	1	6	4	66	25%
	Hip-Hop	0	27	0	79	0	0	0	1	28	59%
	Internati	0	0	5	1	1	0	3	0	13	4%
	Jazz	1	6	8	1	0	0	3	0	11	0%
	Old-Time	3	0	6	1	1	1	43	0	3	74%
	Pop	0	10	6	4	0	0	0	1	28	2%
	Rock	1	48	36	15	3	2	2	1	468	81%
		52%	65%	30%	63%	17%	0%	74%	13%	64%	

Naive Bayes

Naive Bayes

This method is based on "naive" assumption that

$$P(x_0, \ldots, x_n | c_i) = P(x_0|c_i)P(x_1|c_i) \ldots P(x_n|c_i)$$

Using a Bayesian approach:

$$\begin{aligned} P(c_i|x_0, \ldots, x_n) &\propto P(x_0, \ldots, x_n|c_i)P(c_i) \\ &\propto P(c_i) \prod_{j=1}^{n} P(x_j|c_i) \end{aligned}$$

Assumption: $\text{logit}\, x_j|c_i \sim N(\mu_i, \sigma_i^2)$

Results

Naive Bayes											
		Reference									
		Classical	Electroni	Folk	Hip-Hop	Internati	Jazz	Old-Time	Pop	Rock	
Predicted	Classical	25	0	10	0	0	1	2	0	2	63%
	Electroni	4	188	8	12	0	1	0	2	111	58%
	Folk	13	5	39	6	1	1	6	3	67	28%
	Hip-Hop	0	27	0	79	0	0	0	1	28	59%
	Internati	0	0	5	1	0	0	2	0	15	0%
	Jazz	2	4	6	1	0	2	2	0	13	7%
	Old-Time	1	0	5	0	1	1	44	0	6	76%
	Pop	0	10	6	4	0	0	0	1	28	2%
	Rock	3	44	40	15	2	0	3	1	468	81%
		52%	68%	33%	67%	0%	33%	75%	13%	63%	

Ensemble Methods

Random Forest

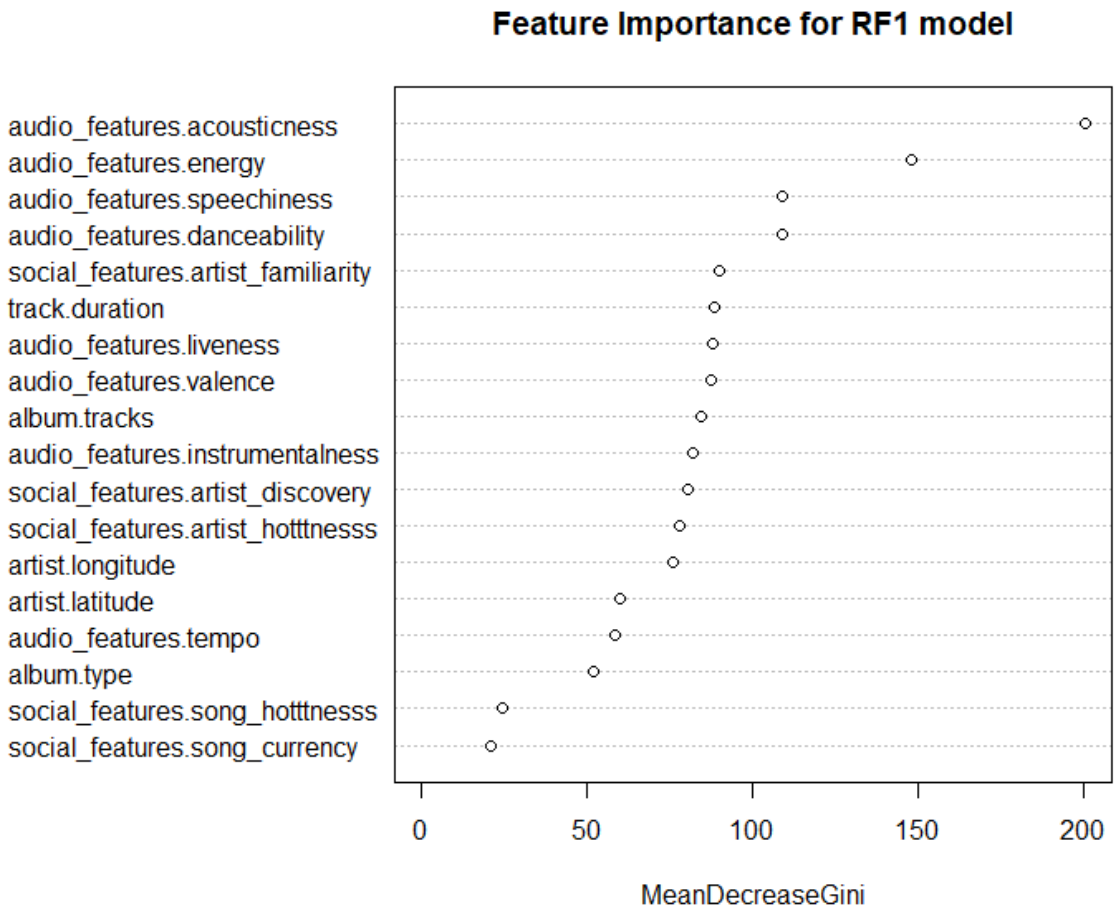
- Classification trees provide a nonparametric approach.
- At each node of the tree, generate binary split which yields the greatest separation of the data.
- Single trees tend to overfit.
- Breiman's Random Forests: average many trees, classify by majority vote. (Breiman, 2001)
- Each tree uses subset of predictors and a random sample of the data.

Results

Random Forest											
		Classical	Electronic	Folk	Hip-Hop	Internatior	Jazz	Old-Time /	Pop	Rock	Accuracy
	Classical	3	5	2	0	0	0	0	0	0	30%
	Electronic	0	93	2	11	0	0	0	1	21	73%
	Folk	1	14	46	2	1	0	0	0	36	46%
	Hip-Hop	0	16	2	76	0	1	0	1	21	65%
	International	0	0	2	0	11	2	0	1	7	48%
	Jazz	0	4	6	0	0	22	0	0	13	49%
	Old-Time / Historic	0	0	0	0	0	1	0	0	4	0%
	Pop	0	13	1	0	0	0	0	19	31	30%
	Rock	0	18	2	3	1	3	0	3	317	91%

Overall accuracy: **69.48%**

Results


Feature Importance for RF1 model
audio_features.acousticness
audio_features.energy
audio_features.speechiness
audio_features.danceability
social_features.artist_familiarity
track.duration
audio_features.liveness
audio_features.valence
album.tracks
audio_features.instrumentalness
social_features.artist_discovery
social_features.artist_hotttnesss
artist.longitude
artist.latitude
audio_features.tempo
album.type
social_features.song_hotttnesss
social_features.song_currency
0
50
100
150
200
MeanDecreaseGini

XGBoost

- "XGBoost" stands for "Extreme Gradient Boosting." (Chen and Guestrin, 2016)
- Boosting: fit many trees to **adaptively reweighted** versions of the data.
- At each tree-fitting, assign **greater weights** to those points misclassified by the last tree.
- Gradient boosting: uses **gradient descent** to minimize errors.
- XG boosting: Optimize hardware and use gradient descent to minimize errors in a **fraction of the time**.
- `https://youtu.be/HD6SRBWKGUE?t=29`

Results

XGBoost											
		Classical	Electronic	Folk	Hip-Hop	Internation	Jazz	Old-Time / I	Pop	Rock	Accuracy
	Classical	32	1	1	0	0	1	0	0	0	91%
	Electronic	0	239	6	18	1	5	0	5	32	78%
	Folk	1	7	88	1	1	3	2	2	9	77%
	Hip-Hop	0	16	2	120	0	0	0	0	4	85%
	International	0	0	1	0	9	0	0	0	0	90%
	Jazz	0	1	0	0	0	25	0	0	0	96%
	Old-Time / Historic	2	0	1	0	1	0	46	0	0	92%
	Pop	0	0	0	0	0	0	0	23	2	92%
	Rock	1	42	20	12	7	12	0	16	560	84%

Overall accuracy: **82.87%**

Model Selection

Summary		
	Method	**Validation Accuracy**
	KNN	60.52%
	Naieve Bayes	63.43%
	Random Forest	69.48%
	XGBoost	**82.87%**

- XGBoost test set accuracy: 82.00%

Conclusions

Conclusions

- Chi-squared tests suggest class and predictors are not independent
- XGBoost achieved the most accurate prediction ($\sim 83\%$)
- k-NN achieved the least accurate prediction ($\sim 61\%$)
- Using the Echonest features resulted in a better predicting accuracy than the Librosa features used by Defferrard (2016)

Disclaimer: A model trained on the FMA data set will only generalize to songs similar to those contained in the data!

The artists included in the FMA data are **not** well-known, top-40 artists.

References

Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM. 2016, pp. 785–794.

Michaël Defferrard et al. *FMA: A Dataset For Music Analysis*. 2016. arXiv: 1612.01840 [cs.SD].

Yoav Freund, Robert E Schapire, et al. "Experiments with a new boosting algorithm". In: Citeseer. 1996.

Yannis Panagakis, Constantine Kotropoulos, and Gonzalo R Arce. "Music genre classification via sparse representations of auditory temporal modulations". In: *2009 17th European Signal Processing Conference*. IEEE. 2009, pp. 1–5.