IST687 Final Project: Music Classification Analysis

**Authors:**

Sebastian Castro

John Fields

Courtney Smith

Jeremy Wallner

Table of Contents

[Overview 2](#_Toc9963110)

[Data 2](#_Toc9963111)

[Data Collection (where did we get the dataset) 2](#_Toc9963112)

[Data Summary (what does the dataset contain) 2](#_Toc9963113)

[Data Import & Cleaning (steps taken to import to R and clean up the dataset) 3](#_Toc9963114)

[Data Exploration (descriptive statistics) 4](#_Toc9963115)

[Data Visualization (histograms, boxplots, etc.) 4](#_Toc9963116)

[Predictive Model 4](#_Toc9963117)

[Summary (what are we trying to figure out) 4](#_Toc9963118)

[Models (models used, why, etc.) 4](#_Toc9963119)

[Results (which models worked, which didn’t) 4](#_Toc9963120)

[Conclusion 4](#_Toc9963121)

VERY ROUGH DRAFT

# Overview

The Million Song Dataset (MSD) is a collection of audio data and metadata for a million popular music tracks. The data is made freely available to encourage research on algorithm development, as well as help new researchers get started in the MIR field.

The purpose of this project is to analyze the Million Song Database to predict “Hot” artists and songs based on the attributes such as familiarity, artist location, loudness, terms used, etc. The analysis was done using R software on a 10,000 track subset of the data and our model was able to predict “Hot” songs with ~80% accuracy.

# Data

## Data Collection (where did we get the dataset)

As the original Million Song Dataset (MSD) is incredibly large (300GB), we based this analysis on a subset of 10,000 songs (1.8GB) for ease of manipulation/collaboration. The dataset was downloaded from [CORGIS](https://think.cs.vt.edu/corgis/csv/music/music.html).

## Data Summary (what does the dataset contain)

The dataset contains 36 variables and 9,996 observations. The table below identifies the various fields and provides a description for each as defined by [millionsongdataset.com](http://millionsongdataset.com/pages/field-list/).

|  |  |
| --- | --- |
| Field Name | Description |
| artist.hotttnesss | algorithmic estimation |
| artist.id | echo nest ID |
| artist.name | artist name |
| artist\_mbtags | tags from musicbrainz.org |
| artist\_mbtags\_count | tag counts from musicbrainz.org |
| bars\_confidence | confidence measure |
| bars\_start | beginning of bars, usually on a beat |
| beats\_confidence | confidence measure |
| beats\_start | result of beat tracking |
| duration | in seconds |
| end\_of\_fade\_in | seconds at the beginning of the song |
| familiarity | algorithmic estimation |
| key | key the song is in |
| key\_confidence | confidence measure |
| latitude | latitude |
| location | location name |
| longitude | longitude |
| loudness | overall loudness in db |
| mode | major or minor |
| mode\_confidence | confidence measure |
| release.id | album ID |
| release.name | album name |
| similar | echo nest artist IDs (sim. algo. unpublished) |
| song.hotttnesss | algorithmic estimation |
| song.id | Echo Nest song ID |
| start\_of\_fade\_out | time in sec |
| tatums\_confidence | confidence measure |
| tatums\_start | smallest rythmic element |
| tempo | estimated tempo in bpm |
| terms | echo nest tags |
| terms\_freq | echo nest tags freqs |
| time\_signature | estimate of number of beats per bar, e.g. 4 |
| time\_signature\_confidence | confidence measure |
| title | song title |
| year | song release year from musicbrainz or 0 |
| artist.hotttnesss.label | artist hotness level as assigned by this team |

## Data Import & Cleaning (steps taken to import to R and clean up the dataset)

After downloading the dataset, we wrote code to import the dataset and review its structure. We quickly found that we did not need all 36 columns, and identified several to remove to make the data easier to work with. Why did we decide to delete these specific columns?

Additionally, we manually created two sets of artist hotness levels – one set containing 3 values and the second containing 5, in order to run our model on both sets to see if one methodology resulted in a more precise prediction.

*3-Value Methodology:*

We manually reviewed the artist hotness numeric value, and assigned labels as follows:

* Cold: < 25%
* Warm: 26% - 50%
* Hot: 51%+

*5-Value Methodology:*

We created various quantiles (95%, 75%, 50%, 25%) based on the artist hotness value and assigned labels as follows:

* Hot: >95% Quantile (0.6011861)
* Warm: >75% Quantile (0.453858)
* Tepid: >50% Quantile (0.3807423)
* Cool: >25% Quantile (0.3252656)
* Frigid: <25% Quantile (0.3252656)

Lastly, we ran code to omit NAs from out dataset.

## Data Exploration (descriptive statistics)

After importing and cleaning the data, we began analyzing the dataset by computing basic statistics at the aggregate as well as for each individual variable. Statistics include mean, median, min, max, standard deviation, quantiles, and skewness.

1. View DS at the aggregate
2. View DS for each field?
3. Correlation Matrix

## Data Visualization (histograms, boxplots, etc.)

We plotted various fields to get a visual sense of how the observations in our dataset are distributed, and identify any trends in our data.

PLOTS

* Song Hotness Distribution
* Artist Geo Map

Additionally, we

# Predictive Model

## Summary (what are we trying to figure out)

## Models (models used, why, etc.)

We selected the Random Forest model to predict the hotness level for an artist. We ran this two ways to determine which approach would result in the lowest error rate.

1. Three Labels
2. Five Labels

## Results (which models worked, which didn’t)

Running the model on the three labels resulted in a lower error rate (20.18% vs.35.64%).

# Conclusion