IST687 - Music Classification Project

Team 2 - Sebastian Castro, John Fields, Courtney Smith, Jeremy Wallner

5/13/2019

## Executive Summary

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.

## Table of Contents

Executive Summary Data Analysis Conclusion Final proofing

## Introduction

## Related Work

Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.

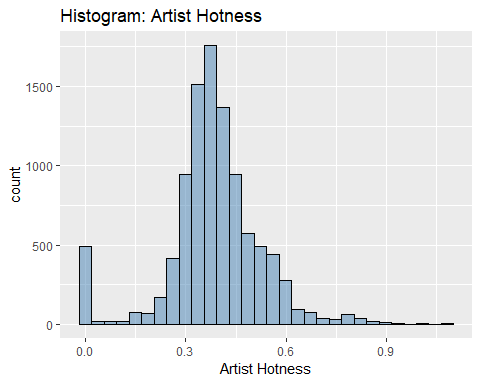
## Dataset

#New code from Courtney to change from 3 to 5 categories of artist hotness  
music <- read.csv("~/Intro data science/Music project/newmusic.csv")  
str(music)

## 'data.frame': 9996 obs. of 36 variables:  
## $ ï..artist.hotttnesss : num 0.402 0.417 0.343 0.454 0.402 ...  
## $ artist.id : Factor w/ 3885 levels "AR009211187B989185",..: 1269 2353 2168 715 3606 2128 1249 129 857 2506 ...  
## $ artist.name : Factor w/ 4409 levels "!!!","(hed) p.e.",..: 682 3796 3560 67 1569 1891 3204 4168 3117 802 ...  
## $ artist\_mbtags : Factor w/ 277 levels "","0.333","60s",..: 1 52 1 262 1 1 1 1 1 1 ...  
## $ artist\_mbtags\_count : num 0 1 0 1 0 0 0 0 0 0 ...  
## $ bars\_confidence : num 0.643 0.007 0.98 0.017 0.175 0.121 0.709 0.142 0.806 0.047 ...  
## $ bars\_start : num 0.585 0.711 0.732 1.306 1.064 ...  
## $ beats\_confidence : num 0.834 1 0.98 0.809 0.883 0.438 0.709 0.234 0.44 1 ...  
## $ beats\_start : num 0.585 0.206 0.732 0.81 0.136 ...  
## $ duration : num 219 148 177 233 210 ...  
## $ end\_of\_fade\_in : num 0.247 0.148 0.282 0 0.066 ...  
## $ familiarity : num 0.582 0.631 0.487 0.63 0.651 ...  
## $ key : num 1 6 8 0 2 5 1 4 4 7 ...  
## $ key\_confidence : num 0.736 0.169 0.643 0.751 0.092 0.635 0 0 0.717 0.053 ...  
## $ latitude : num 37.2 35.1 37.2 37.2 37.2 ...  
## $ location : Factor w/ 1046 levels " "," NC"," UbA!, Minas Gerais",..: 157 584 705 517 705 705 720 150 705 705 ...  
## $ longitude : num -63.9 -90 -63.9 -63.9 -63.9 ...  
## $ loudness : num -11.2 -9.84 -9.69 -9.01 -4.5 ...  
## $ mode : int 0 0 1 1 1 1 1 0 1 0 ...  
## $ mode\_confidence : num 0.636 0.43 0.565 0.749 0.371 0.557 0 0.16 0.652 0.473 ...  
## $ release.id : int 300848 300822 514953 287650 611336 41838 25824 8876 358182 692313 ...  
## $ release.name : Factor w/ 7830 levels "'60s Pop Number 1's",..: 2191 1746 3535 2334 4351 4744 1565 2468 4930 6081 ...  
## $ similar : Factor w/ 2837 levels "AR00K8N11C8A41687B",..: 2408 2225 1145 304 2331 1313 1101 1500 2577 715 ...  
## $ song.hotttnesss : num 0.602 NA NA NA 0.605 ...  
## $ song.id : Factor w/ 9996 levels " Polovtsian Dances / Rimsky-Korsakov: Russian Easter",..: 5350 1014 9225 5465 2424 9456 5094 3118 3378 7971 ...  
## $ start\_of\_fade\_out : num 219 138 172 217 199 ...  
## $ tatums\_confidence : num 0.779 0.969 0.482 0.601 1 0.136 0.467 0.292 0.121 1 ...  
## $ tatums\_start : num 0.285 0.206 0.421 0.563 0.136 ...  
## $ tempo : num 92.2 121.3 100.1 119.3 129.7 ...  
## $ terms : Factor w/ 459 levels "","8-bit","acid jazz",..: 216 34 372 327 325 396 45 329 300 46 ...  
## $ terms\_freq : num 1 1 1 0.989 0.887 ...  
## $ time\_signature : num 4 4 1 4 4 3 1 3 4 4 ...  
## $ time\_signature\_confidence: num 0.778 0.384 0 0 0.562 0.454 0 0.408 0.487 0.878 ...  
## $ title : Factor w/ 9705 levels "","'57 Chevrolet",..: 3681 7526 481 7475 2531 8282 4347 2194 6311 4002 ...  
## $ year : int 0 1969 0 1982 2007 0 0 0 1984 0 ...  
## $ artist.hotttnesss.label : Factor w/ 3 levels "Cold","Hot","Warm": 3 3 3 3 3 3 1 2 1 3 ...

colnames(music)[1] <- "artist.hotttnesss"  
  
#New code from Jeremy importing of song list  
newmusic <- read.csv("~/Intro data science/Music project/newmusic3.csv")  
  
#head(newmusic)  
newmusic2 <- newmusic  
newmusic3 <- newmusic2[-c(1:2,4:9,11,13:14,19:20,23,36)]  
  
newmusic3 <- na.omit(newmusic3)  
cmbomusic <- newmusic3  
  
  
##Artist Hotness Histogram  
library(ggplot2)  
ggplot(music, aes(x=artist.hotttnesss)) + geom\_histogram(color="black", fill="steelblue", alpha=0.5) + xlab("Artist Hotness") + ggtitle("Histogram: Artist Hotness")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



##Function to create descriptive statistics for artist hotness  
descriptive\_stats <- function(vector) { library(moments)   
 result <- c(Mean=mean(vector),  
 Median=median(vector),  
 Min = min(vector),  
 Max = max(vector),  
 SD = sd(vector),  
 Quantile = quantile(vector, probs = c(0.25,.50,0.75, 0.95)),  
 Skewness = skewness(vector) )  
 print(result)   
}  
descriptive\_stats(music$artist.hotttnesss)

## Mean Median Min Max SD   
## 0.3857065 0.3807564 0.0000000 1.0825026 0.1434688   
## Quantile.25% Quantile.50% Quantile.75% Quantile.95% Skewness   
## 0.3255062 0.3807564 0.4539300 0.6011861 -0.1483509

##Methodology for assigning artist hotness levels - uses quantiles from descriptitive\_statistics function  
#95% Quantile: 0.6011861 - Hot  
#75% Quantile: 0.453858 - Warm  
#50% Quantile: 0.3807423 - Tepid  
#25% Quantile: 0.3252656 - Cool  
##Code for assigning labels based on above quantiles  
music$artist.hotness.label <- ifelse(music$artist.hotttnesss >=0.6011861, "Hot",  
 ifelse(music$artist.hotttnesss >=0.453858 & music$artist.hotttnesss <0.6011861, "Warm",  
 ifelse(music$artist.hotttnesss >=0.3807423 & music$artist.hotttnesss <0.453858, "Tepid",  
 ifelse(music$artist.hotttnesss >=0.3252656 & music$artist.hotttnesss <0.3807423, "Cool",  
 ifelse(music$artist.hotttnesss < 0.3252656, "Frigid","Else")))))  
unique(music$artist.hotness.label)

## [1] "Tepid" "Cool" "Warm" "Frigid" "Hot"

#End of new code from Courtney  
#Prior to importing, a new column artist.hotttnesss.label was adding with   
#Hot(>.4590), Warm(<.4590 and >.3357), Cold(<.3357). Four rows with blanks in   
#famiiarity were also deleted.

music <- na.omit(music)  
#Copy original data to a new dataframe music1 and exclude unneeded data  
music <- music[-c(2:5,7,16,19,21:25,30,34)]  
music$artist.hotness.label <- as.factor(music$artist.hotness.label)   
str(music)

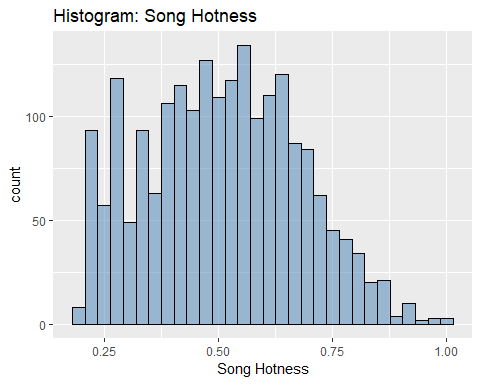
## 'data.frame': 5648 obs. of 23 variables:  
## $ artist.hotttnesss : num 0.402 0.402 0.332 0.296 0.352 ...  
## $ bars\_confidence : num 0.643 0.175 0.806 0.873 0.018 0.013 1 0.507 0.125 0.03 ...  
## $ beats\_confidence : num 0.834 0.883 0.44 0.873 1 0.699 1 0 0.768 1 ...  
## $ beats\_start : num 0.585 0.136 1.226 0.112 0.429 ...  
## $ duration : num 219 210 270 219 245 ...  
## $ end\_of\_fade\_in : num 0.247 0.066 5.3 2.125 0.357 ...  
## $ familiarity : num 0.582 0.651 0.427 0.36 0.545 ...  
## $ key : num 1 2 4 5 7 9 10 7 8 7 ...  
## $ key\_confidence : num 0.736 0.092 0.717 0.354 0.07 0.205 0 1 0.041 0.725 ...  
## $ latitude : num 37.2 37.2 37.2 35.2 37.2 ...  
## $ longitude : num -63.9 -63.9 -63.9 -80 -63.9 ...  
## $ loudness : num -11.2 -4.5 -13.5 -10.02 -7.54 ...  
## $ mode\_confidence : num 0.636 0.371 0.652 0.485 0.686 0.305 0.198 0.829 0.516 0.756 ...  
## $ start\_of\_fade\_out : num 219 199 259 207 227 ...  
## $ tatums\_confidence : num 0.779 1 0.121 0.229 0.728 1 0.774 0.377 0.767 0.238 ...  
## $ tatums\_start : num 0.285 0.136 1.226 0.112 0.173 ...  
## $ tempo : num 92.2 129.7 86.6 146.8 118 ...  
## $ terms\_freq : num 1 0.887 0.96 0.956 1 ...  
## $ time\_signature : num 4 4 4 1 4 4 1 4 5 4 ...  
## $ time\_signature\_confidence: num 0.778 0.562 0.487 0 0.835 0 0.319 0.756 0.579 0.931 ...  
## $ year : int 0 2007 1984 0 0 0 0 1987 0 2004 ...  
## $ artist.hotttnesss.label : Factor w/ 3 levels "Cold","Hot","Warm": 3 3 1 1 3 3 1 3 1 2 ...  
## $ artist.hotness.label : Factor w/ 5 levels "Cool","Frigid",..: 4 4 1 2 1 1 2 4 1 5 ...

##SONG HOTNESS HISTOGRAM From Jeremy  
cmbomusic[cmbomusic==0]<- NA  
#cmbomusic2 <- cmbomusic[-c(5,6)]  
cmbomusic3 <- na.omit(cmbomusic)  
cmbomusic3$song.hotttnesss.label <- ifelse( cmbomusic3$song.hotttnesss >=0.6011861, "Hot",ifelse(cmbomusic3$song.hotttnesss >= 0.453858 & cmbomusic3$song.hotttnesss <0.6011861, "Warm", ifelse(cmbomusic3$song.hotttnesss >=0.3807423 & cmbomusic3$song.hotttnesss < 0.453858, "Tepid",ifelse(cmbomusic3$song.hotttnesss >=0.3252656 & cmbomusic3$song.hotttnesss <0.3807423, "Cool",ifelse(cmbomusic3$song.hotttnesss < 0.3252656, "Frigid","Else")))))  
unique(cmbomusic3$song.hotttnesss.label)

## [1] "Hot" "Tepid" "Cool" "Warm" "Frigid"

cmbomusic3 <- cmbomusic3[-c(2:3,12)]  
ggplot(cmbomusic3, aes(x=song.hotttnesss)) + geom\_histogram(color="black", fill="steelblue", alpha=0.5) + xlab("Song Hotness") + ggtitle("Histogram: Song Hotness")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



##Function to create descriptive statistics for song hotness  
descriptive\_stats2 <- function(vector) { library(moments)   
 result <- c(Mean=mean(vector),  
 Median=median(vector),  
 Min = min(vector),  
 Max = max(vector),  
 SD = sd(vector),  
 Quantile = quantile(vector, probs = c(0.25,.50,0.75, 0.95)),  
 Skewness = skewness(vector) )  
 print(result)   
}  
descriptive\_stats2(cmbomusic3$song.hotttnesss)

## Mean Median Min Max SD   
## 0.5073226 0.5096410 0.1938578 1.0000000 0.1686679   
## Quantile.25% Quantile.50% Quantile.75% Quantile.95% Skewness   
## 0.3827233 0.5096410 0.6301876 0.7900643 0.1304601

cmbomusic3$song.hotness.label <- ifelse( cmbomusic3$song.hotttnesss >=0.64787976, "Hot",ifelse(cmbomusic3$song.hotttnesss >= 0.43437984 & cmbomusic3$song.hotttnesss <0.64787976, "Warm", ifelse( cmbomusic3$song.hotttnesss <0.43437984, "Cold","Else")))  
unique(cmbomusic3$song.hotness.label)

## [1] "Hot" "Cold" "Warm"

cmbomusic3$song.hotttnesss.label <- as.factor(cmbomusic3$song.hotttnesss.label)   
str(cmbomusic3)

## 'data.frame': 2037 obs. of 20 variables:  
## $ artist.name : Factor w/ 4408 levels "!!!","(hed) p.e.",..: 3571 3380 1641 2281 3260 140 128 2194 3424 1968 ...  
## $ latitude : num 47.6 37.2 53.5 37.2 37.2 ...  
## $ location : Factor w/ 1043 levels "","780","Aarhus, Denmark",..: 855 702 556 282 702 858 603 702 603 702 ...  
## $ longitude : num -122.33 -63.93 -2.25 -63.93 -63.93 ...  
## $ loudness : num -9.31 -6.08 -9.62 -10.54 -14.01 ...  
## $ release.id : int 15964 114401 186364 171807 512792 583091 192588 92902 15316 777947 ...  
## $ release.name : Factor w/ 7829 levels "'60s Pop Number 1's",..: 715 5751 1083 3597 921 909 372 5021 2205 914 ...  
## $ song.hotttnesss : num 0.654 0.43 0.346 1 0.694 ...  
## $ song.id : Factor w/ 9995 levels "SOAAAQN12AB01856D3",..: 3 6 7 11 15 16 19 24 29 37 ...  
## $ tatums\_confidence : num 0.898 1 0.445 0.388 0.484 0.873 0.408 0.284 0.992 1 ...  
## $ tatums\_start : num 0.1569 0.0346 0.089 0.1008 0.2263 ...  
## $ tempo : num 131 114 102 151 123 ...  
## $ terms : Factor w/ 458 levels "","8-bit","acid jazz",..: 10 216 8 37 301 198 108 77 328 37 ...  
## $ terms\_freq : num 1 1 1 0.998 0.82 ...  
## $ time\_signature : int 4 5 4 3 4 4 4 4 4 3 ...  
## $ time\_signature\_confidence: num 0.59 0.583 0.097 1 0.369 1 1 0.866 0.919 0.741 ...  
## $ title : Factor w/ 9704 levels "","'57 Chevrolet",..: 7342 6931 9501 3917 539 4665 6981 7116 3031 3729 ...  
## $ year : int 1991 2005 1988 1970 1977 2009 2008 2007 1998 2010 ...  
## $ song.hotttnesss.label : Factor w/ 5 levels "Cool","Frigid",..: 3 4 1 3 3 3 3 1 3 3 ...  
## $ song.hotness.label : chr "Hot" "Cold" "Cold" "Hot" ...

cmbomusic3$song.hotttnesss.label <- ifelse( cmbomusic3$song.hotttnesss >=0.6011861, "Hot",ifelse(cmbomusic3$song.hotttnesss >= 0.453858 & cmbomusic3$song.hotttnesss <0.6011861, "Warm", ifelse(cmbomusic3$song.hotttnesss >=0.3807423 & cmbomusic3$song.hotttnesss < 0.453858, "Tepid",ifelse(cmbomusic3$song.hotttnesss >=0.3252656 & cmbomusic3$song.hotttnesss <0.3807423, "Cool",ifelse(cmbomusic3$song.hotttnesss < 0.3252656, "Frigid","Else")))))  
unique(cmbomusic3$song.hotttnesss.label)

## [1] "Hot" "Tepid" "Cool" "Warm" "Frigid"

cmbomusic3$song.hotttnesss.label <- as.factor(cmbomusic3$song.hotttnesss.label)   
str(cmbomusic3)

## 'data.frame': 2037 obs. of 20 variables:  
## $ artist.name : Factor w/ 4408 levels "!!!","(hed) p.e.",..: 3571 3380 1641 2281 3260 140 128 2194 3424 1968 ...  
## $ latitude : num 47.6 37.2 53.5 37.2 37.2 ...  
## $ location : Factor w/ 1043 levels "","780","Aarhus, Denmark",..: 855 702 556 282 702 858 603 702 603 702 ...  
## $ longitude : num -122.33 -63.93 -2.25 -63.93 -63.93 ...  
## $ loudness : num -9.31 -6.08 -9.62 -10.54 -14.01 ...  
## $ release.id : int 15964 114401 186364 171807 512792 583091 192588 92902 15316 777947 ...  
## $ release.name : Factor w/ 7829 levels "'60s Pop Number 1's",..: 715 5751 1083 3597 921 909 372 5021 2205 914 ...  
## $ song.hotttnesss : num 0.654 0.43 0.346 1 0.694 ...  
## $ song.id : Factor w/ 9995 levels "SOAAAQN12AB01856D3",..: 3 6 7 11 15 16 19 24 29 37 ...  
## $ tatums\_confidence : num 0.898 1 0.445 0.388 0.484 0.873 0.408 0.284 0.992 1 ...  
## $ tatums\_start : num 0.1569 0.0346 0.089 0.1008 0.2263 ...  
## $ tempo : num 131 114 102 151 123 ...  
## $ terms : Factor w/ 458 levels "","8-bit","acid jazz",..: 10 216 8 37 301 198 108 77 328 37 ...  
## $ terms\_freq : num 1 1 1 0.998 0.82 ...  
## $ time\_signature : int 4 5 4 3 4 4 4 4 4 3 ...  
## $ time\_signature\_confidence: num 0.59 0.583 0.097 1 0.369 1 1 0.866 0.919 0.741 ...  
## $ title : Factor w/ 9704 levels "","'57 Chevrolet",..: 7342 6931 9501 3917 539 4665 6981 7116 3031 3729 ...  
## $ year : int 1991 2005 1988 1970 1977 2009 2008 2007 1998 2010 ...  
## $ song.hotttnesss.label : Factor w/ 5 levels "Cool","Frigid",..: 3 4 1 3 3 3 3 1 3 3 ...  
## $ song.hotness.label : chr "Hot" "Cold" "Cold" "Hot" ...

cmbomusic3$song.hotttnesss.label <- ifelse( cmbomusic3$song.hotttnesss >=0.6011861, "Hot",ifelse(cmbomusic3$song.hotttnesss >= 0.453858 & cmbomusic3$song.hotttnesss <0.6011861, "Warm", ifelse(cmbomusic3$song.hotttnesss >=0.3807423 & cmbomusic3$song.hotttnesss < 0.453858, "Tepid",ifelse(cmbomusic3$song.hotttnesss >=0.3252656 & cmbomusic3$song.hotttnesss <0.3807423, "Cool",ifelse(cmbomusic3$song.hotttnesss < 0.3252656, "Frigid","Else")))))  
unique(cmbomusic3$song.hotttnesss.label)

## [1] "Hot" "Tepid" "Cool" "Warm" "Frigid"

str(cmbomusic3)

## 'data.frame': 2037 obs. of 20 variables:  
## $ artist.name : Factor w/ 4408 levels "!!!","(hed) p.e.",..: 3571 3380 1641 2281 3260 140 128 2194 3424 1968 ...  
## $ latitude : num 47.6 37.2 53.5 37.2 37.2 ...  
## $ location : Factor w/ 1043 levels "","780","Aarhus, Denmark",..: 855 702 556 282 702 858 603 702 603 702 ...  
## $ longitude : num -122.33 -63.93 -2.25 -63.93 -63.93 ...  
## $ loudness : num -9.31 -6.08 -9.62 -10.54 -14.01 ...  
## $ release.id : int 15964 114401 186364 171807 512792 583091 192588 92902 15316 777947 ...  
## $ release.name : Factor w/ 7829 levels "'60s Pop Number 1's",..: 715 5751 1083 3597 921 909 372 5021 2205 914 ...  
## $ song.hotttnesss : num 0.654 0.43 0.346 1 0.694 ...  
## $ song.id : Factor w/ 9995 levels "SOAAAQN12AB01856D3",..: 3 6 7 11 15 16 19 24 29 37 ...  
## $ tatums\_confidence : num 0.898 1 0.445 0.388 0.484 0.873 0.408 0.284 0.992 1 ...  
## $ tatums\_start : num 0.1569 0.0346 0.089 0.1008 0.2263 ...  
## $ tempo : num 131 114 102 151 123 ...  
## $ terms : Factor w/ 458 levels "","8-bit","acid jazz",..: 10 216 8 37 301 198 108 77 328 37 ...  
## $ terms\_freq : num 1 1 1 0.998 0.82 ...  
## $ time\_signature : int 4 5 4 3 4 4 4 4 4 3 ...  
## $ time\_signature\_confidence: num 0.59 0.583 0.097 1 0.369 1 1 0.866 0.919 0.741 ...  
## $ title : Factor w/ 9704 levels "","'57 Chevrolet",..: 7342 6931 9501 3917 539 4665 6981 7116 3031 3729 ...  
## $ year : int 1991 2005 1988 1970 1977 2009 2008 2007 1998 2010 ...  
## $ song.hotttnesss.label : chr "Hot" "Tepid" "Cool" "Hot" ...  
## $ song.hotness.label : chr "Hot" "Cold" "Cold" "Hot" ...

cmbomusic3$song.hotttnesss.label <- as.factor(cmbomusic3$song.hotttnesss.label)   
cmbomusic3$song.hotttnesss.label <- as.factor(cmbomusic3$song.hotttnesss.label)   
str(cmbomusic3)

## 'data.frame': 2037 obs. of 20 variables:  
## $ artist.name : Factor w/ 4408 levels "!!!","(hed) p.e.",..: 3571 3380 1641 2281 3260 140 128 2194 3424 1968 ...  
## $ latitude : num 47.6 37.2 53.5 37.2 37.2 ...  
## $ location : Factor w/ 1043 levels "","780","Aarhus, Denmark",..: 855 702 556 282 702 858 603 702 603 702 ...  
## $ longitude : num -122.33 -63.93 -2.25 -63.93 -63.93 ...  
## $ loudness : num -9.31 -6.08 -9.62 -10.54 -14.01 ...  
## $ release.id : int 15964 114401 186364 171807 512792 583091 192588 92902 15316 777947 ...  
## $ release.name : Factor w/ 7829 levels "'60s Pop Number 1's",..: 715 5751 1083 3597 921 909 372 5021 2205 914 ...  
## $ song.hotttnesss : num 0.654 0.43 0.346 1 0.694 ...  
## $ song.id : Factor w/ 9995 levels "SOAAAQN12AB01856D3",..: 3 6 7 11 15 16 19 24 29 37 ...  
## $ tatums\_confidence : num 0.898 1 0.445 0.388 0.484 0.873 0.408 0.284 0.992 1 ...  
## $ tatums\_start : num 0.1569 0.0346 0.089 0.1008 0.2263 ...  
## $ tempo : num 131 114 102 151 123 ...  
## $ terms : Factor w/ 458 levels "","8-bit","acid jazz",..: 10 216 8 37 301 198 108 77 328 37 ...  
## $ terms\_freq : num 1 1 1 0.998 0.82 ...  
## $ time\_signature : int 4 5 4 3 4 4 4 4 4 3 ...  
## $ time\_signature\_confidence: num 0.59 0.583 0.097 1 0.369 1 1 0.866 0.919 0.741 ...  
## $ title : Factor w/ 9704 levels "","'57 Chevrolet",..: 7342 6931 9501 3917 539 4665 6981 7116 3031 3729 ...  
## $ year : int 1991 2005 1988 1970 1977 2009 2008 2007 1998 2010 ...  
## $ song.hotttnesss.label : Factor w/ 5 levels "Cool","Frigid",..: 3 4 1 3 3 3 3 1 3 3 ...  
## $ song.hotness.label : chr "Hot" "Cold" "Cold" "Hot" ...

#View the number of Cold/Warm/Hot labels   
 table(cmbomusic3$song.hotttnesss.label)

##   
## Cool Frigid Hot Tepid Warm   
## 171 337 629 278 622

cmbomusic3$song.hotness.label <- ifelse( cmbomusic3$song.hotttnesss >=0.64787976, "Hot",ifelse(cmbomusic3$song.hotttnesss >= 0.43437984 & cmbomusic3$song.hotttnesss <0.64787976, "Warm", ifelse( cmbomusic3$song.hotttnesss <0.43437984, "Cold","Else")))  
unique(cmbomusic3$song.hotness.label)

## [1] "Hot" "Cold" "Warm"

## Features

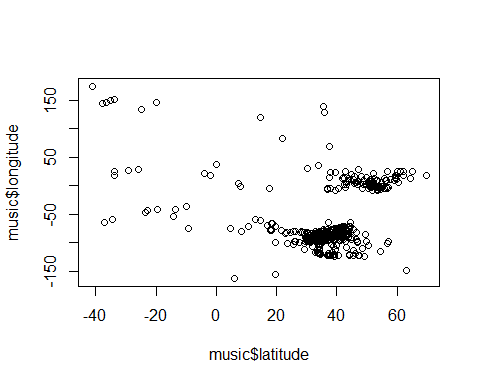
#View the number of Cold/Warm/Hot labels   
table(music$artist.hotttnesss.label)

##   
## Cold Hot Warm   
## 1180 1579 2889

#View the number of Frigid/Cool/Tepid/Warm/Hot labels   
table(music$artist.hotness.label)

##   
## Cool Frigid Hot Tepid Warm   
## 1444 973 278 1566 1387

#Plot artists latitude and longitude  
plot(music$latitude,music$longitude)



cmbomusic3$song.hotness.label <- as.factor(cmbomusic3$song.hotness.label)   
cmbomusic3$song.hotness.label <- as.factor(cmbomusic3$song.hotness.label)   
str(cmbomusic3)

## 'data.frame': 2037 obs. of 20 variables:  
## $ artist.name : Factor w/ 4408 levels "!!!","(hed) p.e.",..: 3571 3380 1641 2281 3260 140 128 2194 3424 1968 ...  
## $ latitude : num 47.6 37.2 53.5 37.2 37.2 ...  
## $ location : Factor w/ 1043 levels "","780","Aarhus, Denmark",..: 855 702 556 282 702 858 603 702 603 702 ...  
## $ longitude : num -122.33 -63.93 -2.25 -63.93 -63.93 ...  
## $ loudness : num -9.31 -6.08 -9.62 -10.54 -14.01 ...  
## $ release.id : int 15964 114401 186364 171807 512792 583091 192588 92902 15316 777947 ...  
## $ release.name : Factor w/ 7829 levels "'60s Pop Number 1's",..: 715 5751 1083 3597 921 909 372 5021 2205 914 ...  
## $ song.hotttnesss : num 0.654 0.43 0.346 1 0.694 ...  
## $ song.id : Factor w/ 9995 levels "SOAAAQN12AB01856D3",..: 3 6 7 11 15 16 19 24 29 37 ...  
## $ tatums\_confidence : num 0.898 1 0.445 0.388 0.484 0.873 0.408 0.284 0.992 1 ...  
## $ tatums\_start : num 0.1569 0.0346 0.089 0.1008 0.2263 ...  
## $ tempo : num 131 114 102 151 123 ...  
## $ terms : Factor w/ 458 levels "","8-bit","acid jazz",..: 10 216 8 37 301 198 108 77 328 37 ...  
## $ terms\_freq : num 1 1 1 0.998 0.82 ...  
## $ time\_signature : int 4 5 4 3 4 4 4 4 4 3 ...  
## $ time\_signature\_confidence: num 0.59 0.583 0.097 1 0.369 1 1 0.866 0.919 0.741 ...  
## $ title : Factor w/ 9704 levels "","'57 Chevrolet",..: 7342 6931 9501 3917 539 4665 6981 7116 3031 3729 ...  
## $ year : int 1991 2005 1988 1970 1977 2009 2008 2007 1998 2010 ...  
## $ song.hotttnesss.label : Factor w/ 5 levels "Cool","Frigid",..: 3 4 1 3 3 3 3 1 3 3 ...  
## $ song.hotness.label : Factor w/ 3 levels "Cold","Hot","Warm": 2 1 1 2 2 2 2 1 3 2 ...

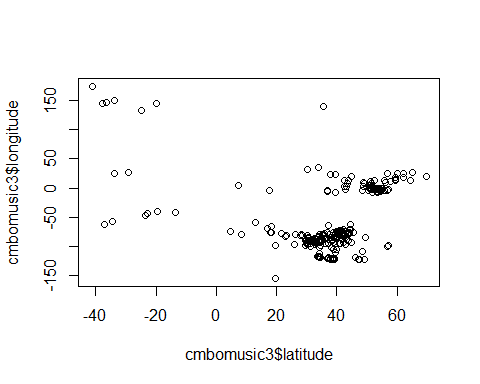
#View the number of Cold/Warm/Hot labels   
table(cmbomusic3$song.hotness.label)

##   
## Cold Hot Warm   
## 707 440 890

#View the number of Frigid/Cool/Tepid/Warm/Hot labels   
 table(cmbomusic3$song.hotttnesss.label)

##   
## Cool Frigid Hot Tepid Warm   
## 171 337 629 278 622

#Plot artists latitude and longitude  
 plot(cmbomusic3$latitude,cmbomusic3$longitude)



#Plot artist hotttnesss  
#hist(music$artist.hotttnesss,breaks=20)  
#hist(music$artist.hotness,breaks=20)

#THIS IS INCOMPLETE CODE FOR PLOTTING ADDITIONAL DATA… #Create a map of the world mapWorld <- borders(“world”, colour=“gray50”, fill=“white”)

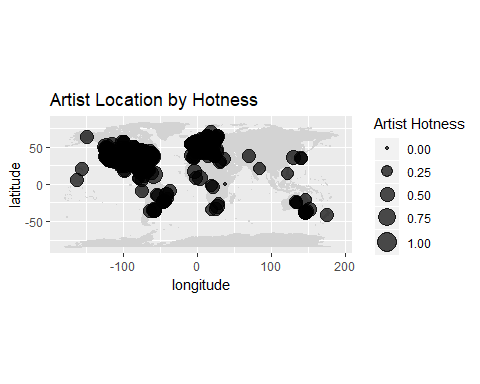
#New code from John for creating a map of the world showing latitude/longitude and artist hotness  
#Code based on info from https://rpubs.com/spoonerf/global\_map  
library(dplyr)

##   
## Attaching package: 'dplyr'

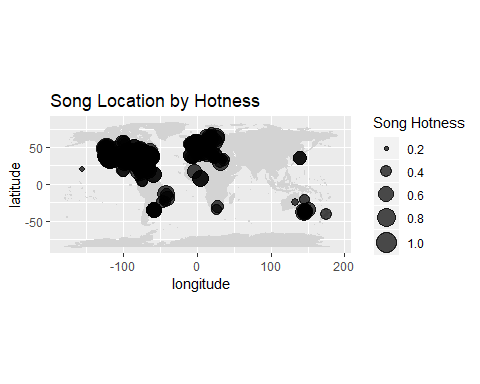
## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

loc<-data.frame(music$longitude,music$latitude,music$artist.hotttnesss)  
loc<-unique(loc)  
colnames(loc)<-c("longitude", "latitude","artist hotness")  
loc\_df<-data.frame(loc)  
library(maps)  
library(mapdata)  
library(ggplot2)  
ahworld <- ggplot(data=loc\_df, aes(longitude, latitude, group=NULL,fill=NULL,size=artist.hotness))+#, fill=hole)) +   
 borders(fill="light grey",colour="light grey")+  
 geom\_point(color="black",alpha=I(7/10))+  
 scale\_size(range=c(1,7), guide = "legend",labs(size="Artist Hotness"))+  
 coord\_equal()+ ggtitle("Artist Location by Hotness")  
ahworld



#New code from John for creating a map of the world showing latitude/longitude and artist hotness  
#Code based on info from https://rpubs.com/spoonerf/global\_map  
library(dplyr)  
songlc<-data.frame(cmbomusic3$longitude,cmbomusic3$latitude,cmbomusic3$song.hotttnesss)  
songlc<-unique(songlc)  
colnames(songlc)<-c("longitude", "latitude","song hotness")  
songlc\_df<-data.frame(songlc)  
library(maps)  
library(mapdata)  
library(ggplot2)  
songlc\_dfwrld <- ggplot(data=songlc\_df, aes(longitude, latitude, group=NULL,fill=NULL,size=song.hotness))+#, fill=hole)) +   
 borders(fill="light grey",colour="light grey")+  
 geom\_point(color="black",alpha=I(7/10))+  
 scale\_size(range=c(1,7), guide = "legend",labs(size="Song Hotness"))+  
 coord\_equal()+ ggtitle("Song Location by Hotness")  
songlc\_dfwrld



## Methods

#Do analysis to determine hot/warm/cold artists based on hotttnesss  
#The ramdom forest analysis is from a training video by Bharatendra Rai   
#at https://www.youtube.com/watch?v=dJclNIN-TPo  
#Data Partition - ind = independent samples  
#The code below runs in console but not R Markdown  
set.seed(123)  
ind<- sample(2,nrow(music), replace=TRUE,prob=c(0.7,0.3))  
train <- music[ind==1,]  
test <- music[ind==2,]  
#Run randomForest on 3 levels  
library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

set.seed(222)  
rf <- randomForest(music[,-21:-22],music[,21])  
print(rf)

##   
## Call:  
## randomForest(x = music[, -21:-22], y = music[, 21])   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 7  
##   
## Mean of squared residuals: 821516.8  
## % Var explained: 17.13

attributes(rf)

## $names  
## [1] "call" "type" "predicted"   
## [4] "mse" "rsq" "oob.times"   
## [7] "importance" "importanceSD" "localImportance"  
## [10] "proximity" "ntree" "mtry"   
## [13] "forest" "coefs" "y"   
## [16] "test" "inbag"   
##   
## $class  
## [1] "randomForest"

rf$confusion

## NULL

#Run randomForest on 5 levels  
library(randomForest)  
set.seed(222)  
rf2 <- randomForest(music[,-21:-22],music[,22])  
print(rf2)

##   
## Call:  
## randomForest(x = music[, -21:-22], y = music[, 22])   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 0%  
## Confusion matrix:  
## Cold Hot Warm class.error  
## Cold 1180 0 0 0  
## Hot 0 1579 0 0  
## Warm 0 0 2889 0

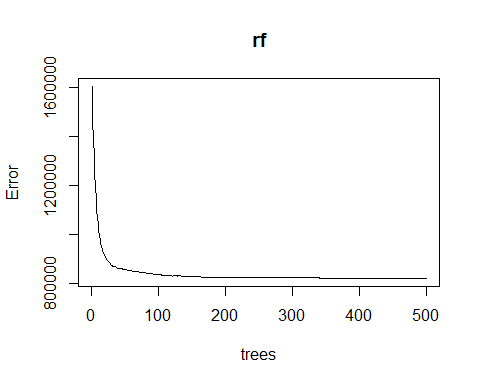
attributes(rf2)

## $names  
## [1] "call" "type" "predicted"   
## [4] "err.rate" "confusion" "votes"   
## [7] "oob.times" "classes" "importance"   
## [10] "importanceSD" "localImportance" "proximity"   
## [13] "ntree" "mtry" "forest"   
## [16] "y" "test" "inbag"   
##   
## $class  
## [1] "randomForest"

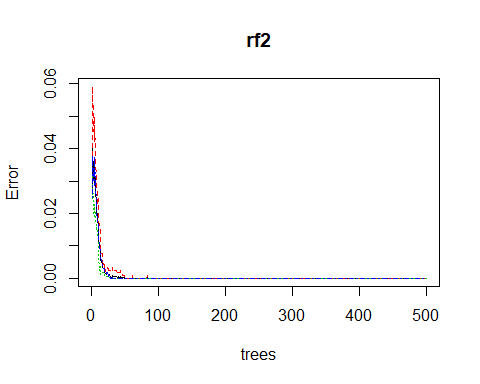
rf2$confusion

## Cold Hot Warm class.error  
## Cold 1180 0 0 0  
## Hot 0 1579 0 0  
## Warm 0 0 2889 0

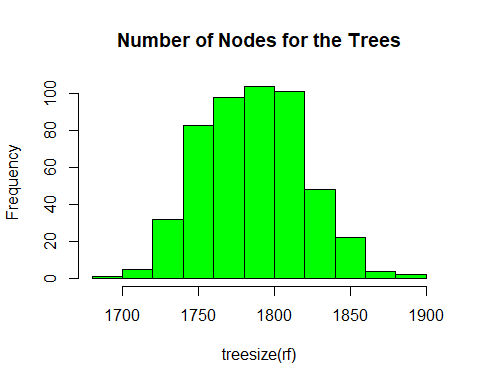
#Run randomForest again with tune mtry data from below  
#rfx <- randomForest(artist.hotness.label ~.,data=music,ntree=200,mtry=8,importance=TRUE,proximity=TRUE)  
#Prediction & Confusion Matrix - train data  
#library(caret)  
#p1<-predict(rfx,train)  
#confusionMatrix(p1,train)  
#Predition & Confusion Matrix - test data  
#p2<-predict(rfx,test)  
#confusionMatrix(p2,test$artist.hotness.label)  
#Error rate of Random Forest  
plot(rf)



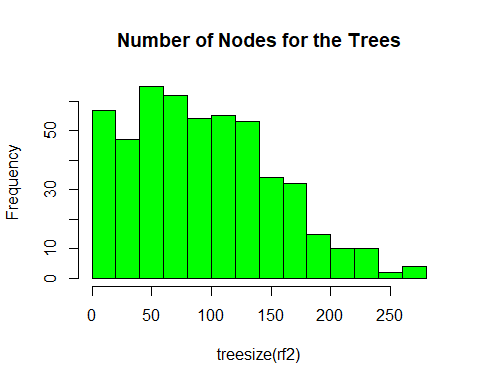
plot(rf2)



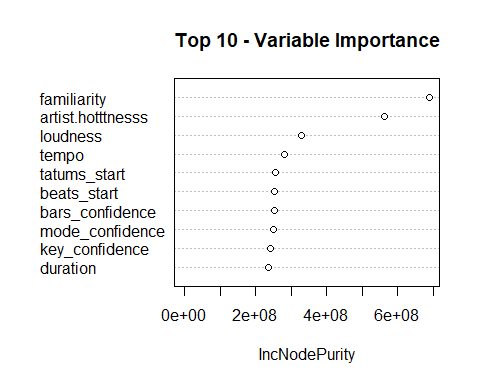
#The error rate is not improving after ~100 trees  
#Tune mtry  
#t <- tuneRF(train[,-21],train[,21],  
# stepFactor=.5,  
# plot=TRUE,  
# ntreeTry=200,  
# trace=TRUE,  
# improve=0.05)  
#No. of nodes for the trees  
hist(treesize(rf),  
 main="Number of Nodes for the Trees",  
 col="green")



hist(treesize(rf2),  
 main="Number of Nodes for the Trees",  
 col="green")



# Variable Importance  
# Familiarity is much more important than the other variables. Should it be removed and run again?  
varImpPlot(rf,  
 sort=T,  
 n.var=10,  
 main="Top 10 - Variable Importance")



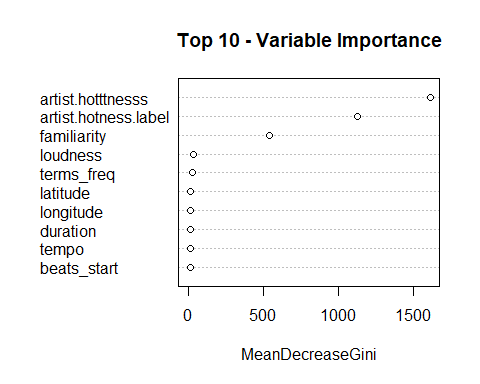
importance(rf)

## IncNodePurity  
## artist.hotttnesss 563218297  
## bars\_confidence 252523207  
## beats\_confidence 207506459  
## beats\_start 252820957  
## duration 236106459  
## end\_of\_fade\_in 193328655  
## familiarity 690579780  
## key 143069066  
## key\_confidence 242295223  
## latitude 153228080  
## longitude 147901164  
## loudness 327902631  
## mode\_confidence 249019821  
## start\_of\_fade\_out 235057180  
## tatums\_confidence 229094593  
## tatums\_start 255065275  
## tempo 280336312  
## terms\_freq 216110169  
## time\_signature 68065241  
## time\_signature\_confidence 176545618  
## artist.hotness.label 137216609

varUsed(rf)

## [1] 53148 50555 44625 50215 47644 41067 54137 36936 49115 29607 28673  
## [12] 56423 50059 47475 47232 50012 52395 35324 20773 40339 6991

varImpPlot(rf2,  
 sort=T,  
 n.var=10,  
 main="Top 10 - Variable Importance")



importance(rf2)

## MeanDecreaseGini  
## artist.hotttnesss 1609.271906  
## bars\_confidence 10.026093  
## beats\_confidence 8.582830  
## beats\_start 10.394477  
## duration 10.656481  
## end\_of\_fade\_in 10.254585  
## familiarity 537.989493  
## key 5.785440  
## key\_confidence 8.490916  
## latitude 13.482175  
## longitude 12.116437  
## loudness 32.799029  
## mode\_confidence 9.648894  
## start\_of\_fade\_out 10.284684  
## tatums\_confidence 7.939542  
## tatums\_start 10.273939  
## tempo 10.641052  
## terms\_freq 24.524975  
## time\_signature 3.346170  
## time\_signature\_confidence 6.196274  
## artist.hotness.label 1129.118077

varUsed(rf2)

## [1] 7690 1844 1611 1994 1922 1573 4255 1262 1776 1594 1573 2334 1828 1900  
## [15] 1631 1942 1969 1861 599 1296 3996

#Mulit-dimenstional Scaling Plot  
#The code below causes R to lock up...  
#MDSplot(rf,train$artist.hotttnesss.label)  
  
  
cmbomusic4 <- na.omit(cmbomusic3)  
cmbomusic5 <- cmbomusic4[-c(1,3,7:9,13,17,20)]  
str(cmbomusic5)

## 'data.frame': 2037 obs. of 12 variables:  
## $ latitude : num 47.6 37.2 53.5 37.2 37.2 ...  
## $ longitude : num -122.33 -63.93 -2.25 -63.93 -63.93 ...  
## $ loudness : num -9.31 -6.08 -9.62 -10.54 -14.01 ...  
## $ release.id : int 15964 114401 186364 171807 512792 583091 192588 92902 15316 777947 ...  
## $ tatums\_confidence : num 0.898 1 0.445 0.388 0.484 0.873 0.408 0.284 0.992 1 ...  
## $ tatums\_start : num 0.1569 0.0346 0.089 0.1008 0.2263 ...  
## $ tempo : num 131 114 102 151 123 ...  
## $ terms\_freq : num 1 1 1 0.998 0.82 ...  
## $ time\_signature : int 4 5 4 3 4 4 4 4 4 3 ...  
## $ time\_signature\_confidence: num 0.59 0.583 0.097 1 0.369 1 1 0.866 0.919 0.741 ...  
## $ year : int 1991 2005 1988 1970 1977 2009 2008 2007 1998 2010 ...  
## $ song.hotttnesss.label : Factor w/ 5 levels "Cool","Frigid",..: 3 4 1 3 3 3 3 1 3 3 ...

cmbomusic5$song.hotness.label <- as.factor(cmbomusic4$song.hotness.label)   
rf3 <- randomForest(cmbomusic5[,-12:-13],cmbomusic5[,13])  
rf3

##   
## Call:  
## randomForest(x = cmbomusic5[, -12:-13], y = cmbomusic5[, 13])   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 50.96%  
## Confusion matrix:  
## Cold Hot Warm class.error  
## Cold 292 26 389 0.5869873  
## Hot 63 124 253 0.7181818  
## Warm 237 70 583 0.3449438

print(rf3)

##   
## Call:  
## randomForest(x = cmbomusic5[, -12:-13], y = cmbomusic5[, 13])   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 50.96%  
## Confusion matrix:  
## Cold Hot Warm class.error  
## Cold 292 26 389 0.5869873  
## Hot 63 124 253 0.7181818  
## Warm 237 70 583 0.3449438

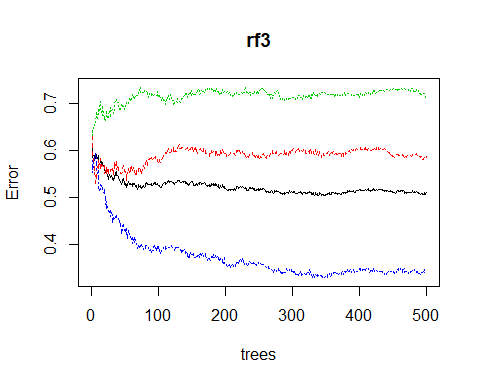
attributes(rf3)

## $names  
## [1] "call" "type" "predicted"   
## [4] "err.rate" "confusion" "votes"   
## [7] "oob.times" "classes" "importance"   
## [10] "importanceSD" "localImportance" "proximity"   
## [13] "ntree" "mtry" "forest"   
## [16] "y" "test" "inbag"   
##   
## $class  
## [1] "randomForest"

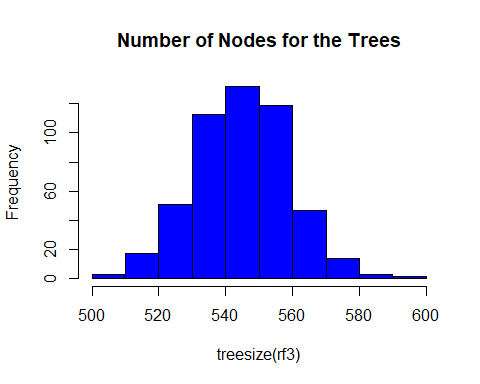
rf3$confusion

## Cold Hot Warm class.error  
## Cold 292 26 389 0.5869873  
## Hot 63 124 253 0.7181818  
## Warm 237 70 583 0.3449438

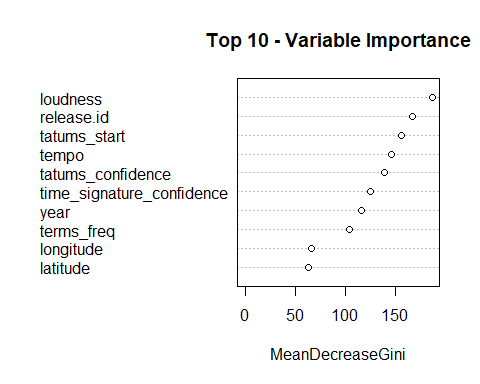
plot(rf3)



hist(treesize(rf3),  
 main="Number of Nodes for the Trees",  
 col="blue")



varImpPlot(rf3,  
 sort=T,  
 n.var=10,  
 main="Top 10 - Variable Importance")

 ## Results

## Conclusion

## Appendices