IST687 - Music Classification Project

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

4/18/2019

Executive Summary

Table of Contents

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

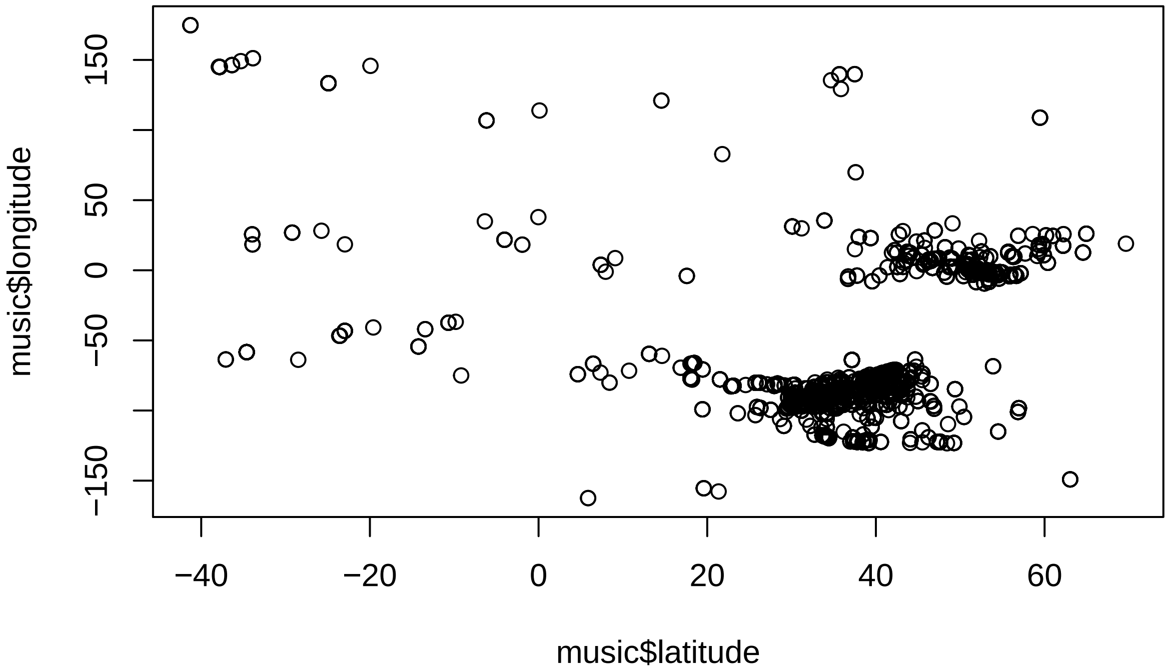
```
#Prior to importing, a new column artist.hotttnesss.label was adding with
#Hot(>.4590), Warm(<.4590 and >.3357), Cold(<.3357).  Four rows with blanks in
#famiiarity were also deleted.
music <- read.csv(file="~/Library/Mobile Documents/com~apple~CloudDocs/Syracuse/IST687/Project/Music Pr
#Copy original data to a new dataframe music1 and exclude unneeded data
music1 <- music[-c(1:5,7,16,19,21:25,30,34)]
```

Features

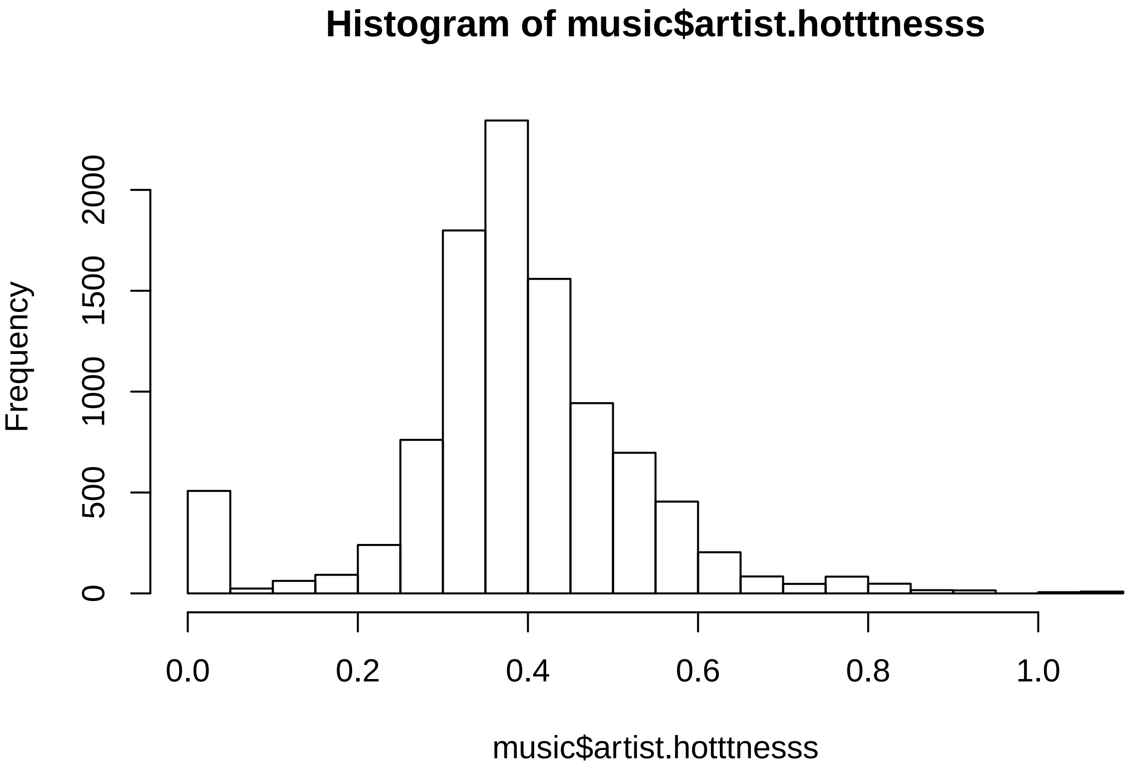
```
#View the number of Cold/Warm/Hot labels
table(music1$artist.hotttnesss.label)
```

```
##
## Cold  Hot Warm
## 2870 2376 4750
```

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



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



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

#Code from https://rpubs.com/spoonerf/global_map #Need to figure out what to put in locs locs<-read.csv("my_locations.csv") locs<- sp_dups<-data.frame(ddply(locs,.(Longitude,Latitude),nrow)) sp_dups$loc_id < -1 : length(sp_dups$Longitude) sp_dups_df<-merge(sp_dups, locs, by=c("Longitude","Latitude")) loc<-data.frame(sp_dups_df$Longitude, sp_{d}ups_{d}f$Latitude,sp_dups_df$V1) loc<-unique(loc) colnames(loc)<-c("Longitude", "Latitude", "V1")

coordinates(loc)<-c("Longitude","Latitude") proj4string(loc) <- CRS("+proj=longlat")

loc_df<-data.frame(loc)

theme_opts <- list(theme(panel.grid.minor = element_blank(), panel.grid.major = element_blank(), panel.background = element_blank(), plot.background = element_rect(fill="white"), panel.border = element_blank(), axis.line = element_blank(), axis.text.x = element_blank(), axis.text.y = element_blank(), axis.ticks = element_blank(), axis.title.x = element_blank(), axis.title.y = element_blank(), plot.title = element_text(size=22)))

library(maps) library(mapdata)

ggplot(data=loc_df, aes(Longitude, Latitude, group=NULL,fill=NULL,size=V1))+#, 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="No. of Populations"))+ coord_equal()+ theme_opts

Methods

```
#Do analysis to determine hot/warm/cold artists based on hotttnesss

#The ramdom forest analysis (106-163) 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(music1), replace=TRUE,prob=c(0.7,0.3))
train <- music1[ind==1,]
test <- music1[ind==2,]

#Run randomForest on music1
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
set.seed(222)
rf <- randomForest(music1[,-21],music1[,21])
print(rf)
```

```
##
## Call:
##  randomForest(x = music1[, -21], y = music1[, 21])
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 19.18%
## Confusion matrix:
##      Cold  Hot Warm class.error
## Cold 2083   14  773   0.2742160
## Hot     8 1968  400   0.1717172
## Warm  455  267 4028   0.1520000
```

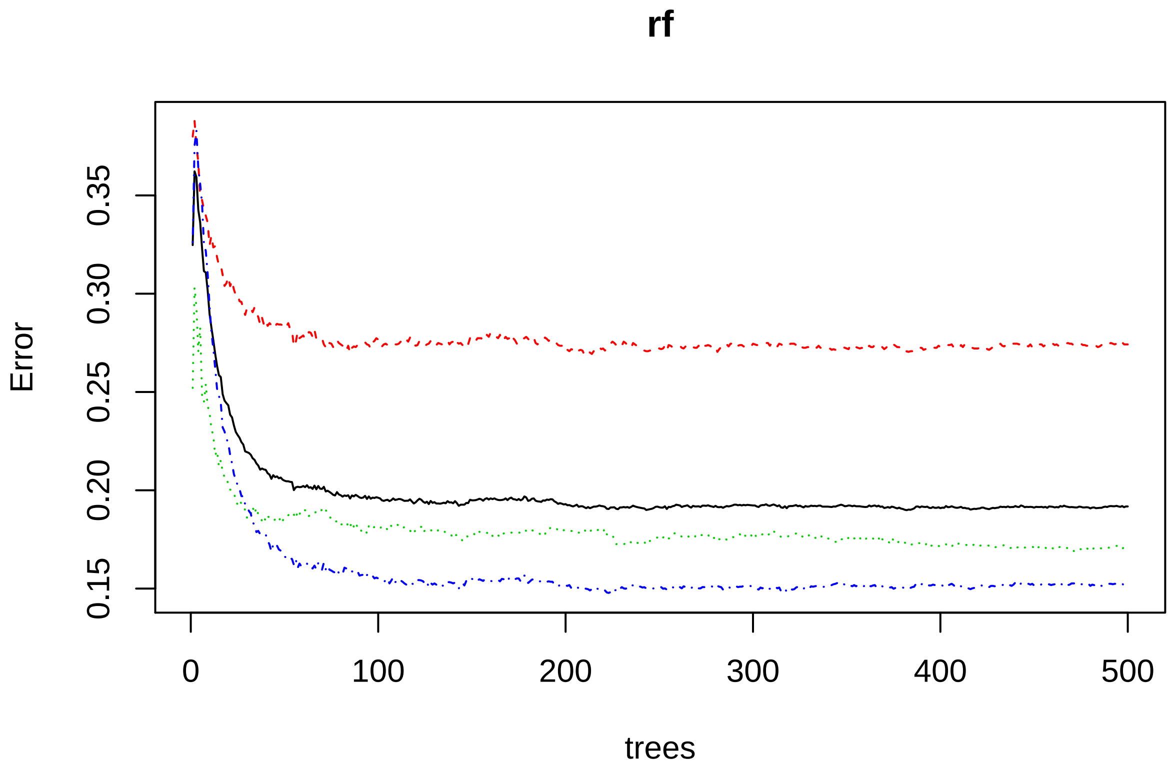
```
attributes(rf)
```

```
## $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"
```

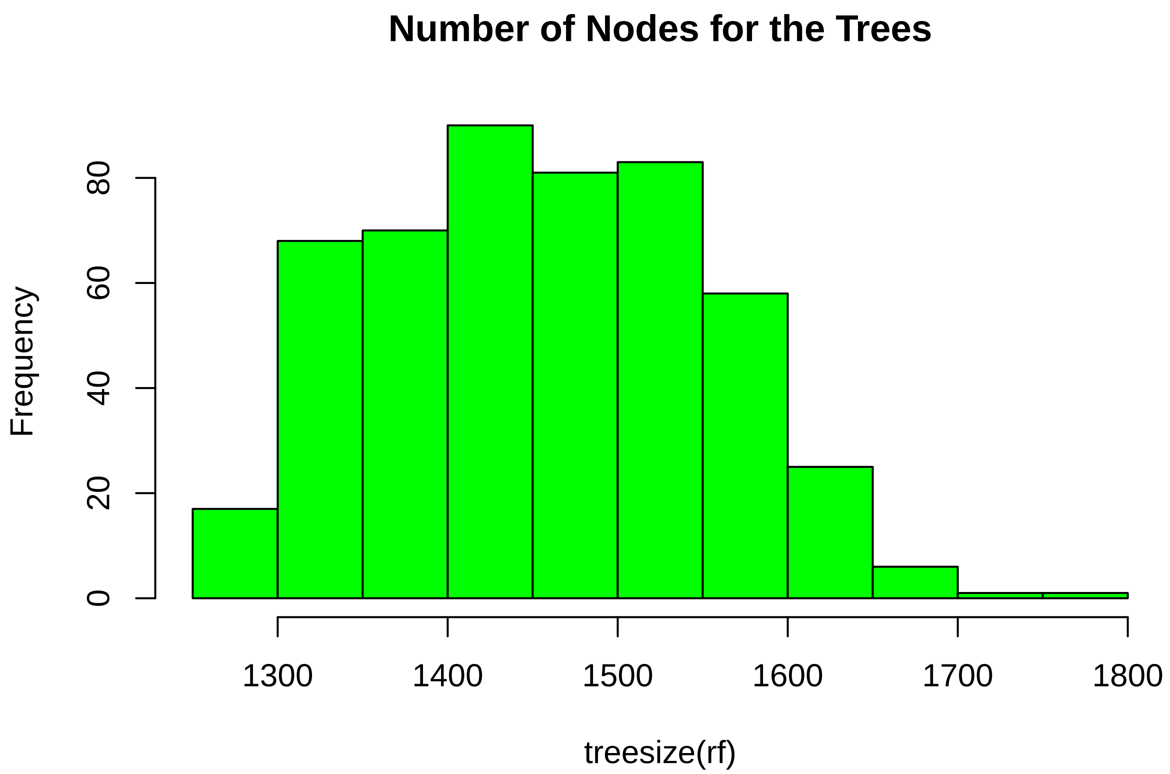
```
rf$confusion
```

```
##      Cold  Hot Warm class.error
## Cold 2083   14  773   0.2742160
## Hot     8 1968  400   0.1717172
## Warm  455  267 4028   0.1520000
```

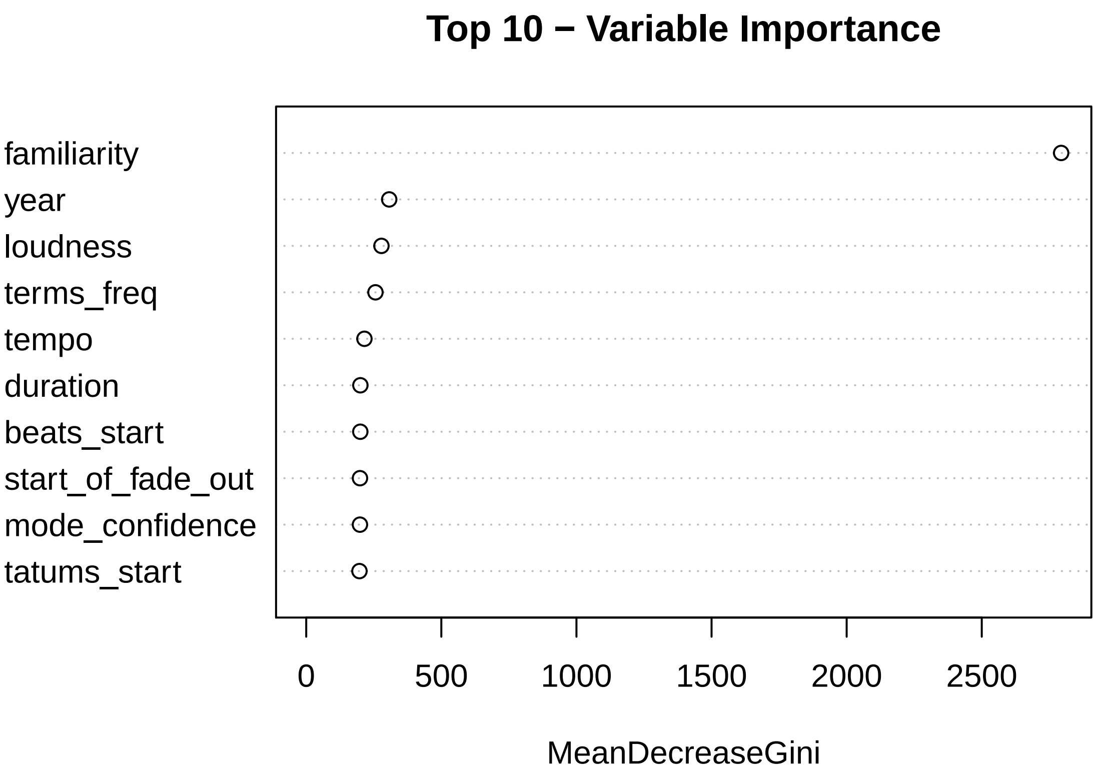
```
#Run randomForest again with tune mtry data from below
#Need HELP to fix the next line of code so it works...
#rf <- randomForest(artist.hotttnesss.label ~.,data=music1,ntree=200,mtry=8,
#importance=TRUE,proximity=TRUE)

#Prediction & Confusion Matrix - train data
#library(caret)
#p1<-predict(rf,train)
# For some reason this is returning an error buit p2 below is working
#confusionMatrix(p1,train)

#Predition & Confusion Matrix - test data
#p2<-predict(rf,test)
#confusionMatrix(p2,test$artist.hotttnesss.label)

#Error rate of Random Forest
plot(rf)
```



```
#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")
```



```
# 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)
```

```
##                          MeanDecreaseGini
## bars_confidence                 190.72247
## beats_confidence                166.50136
## beats_start                     200.11836
## duration                        200.41104
## end_of_fade_in                  158.14247
## familiarity                    2793.89271
## key                             123.67454
## key_confidence                  186.25796
## latitude                        158.41154
## longitude                       144.10180
## loudness                        278.31880
## mode_confidence                 198.81990
## start_of_fade_out               198.91655
## tatums_confidence               175.52538
## tatums_start                    196.93868
## tempo                           215.29801
## terms_freq                      256.20849
## time_signature                   57.60156
## time_signature_confidence       144.43782
## year                            306.99483
```

```
varUsed(rf)
```

```
##  [1] 40656 36125 42149 40892 32952 73071 29208 40149 22266 21818 45831
## [12] 41661 40861 37682 41306 44176 30635 13853 31723 20627
```

```
#Mulit-dimenstional Scaling Plot
#The code below causes R to lock up...
```

```
#MDSplot(rf,train$artist.hotttnesss.label)
```

Results

Conclusion

Appendices