Songs Analysis

Thomas Astad Sve
University of California,
San Diego
& Norwegian University of
Science and Technology
thomaasv@stud.ntnu.no

Abstract

An analysis of songs

With the increasing popularity of the field known as "Big Data" we ask ourselves if by collecting information about songs, we can predict which song will be the next hit that will be listened all over the world. Can record labels use machine learning and big data to discover or create the next hit? By downloading a dataset known as the million songs dataset, I am trying to get an answer to this question.

Contents

1 Introduction

Music has had an important role in our culture troughout human history. As a result of this importance, followes a billion dollar industry. In 2014 alone, this industry generated 15 billion dollars [1]. Most of the income is made from mainstream popular songs and goes straight to the hands of record labels companies. Having a good understanding and of songs is a valuable skillset to have in this industry. Not only do the record labels spend millions of dollars on the artist and repertoire (AR) division to scout for the next big talent, . of what song will get popular is a valuable skillset to have in this industry, not only making the record labels rich [2], but also attracting listeners to radio station, making artists famous and help digital and physical music marketplaces.

This project will try to see if the next hit can be predicted from a given set of features.

2 Related Work

A similar project is done by three students at Stanford University that did a song popularity prediction with a computer science view [6]. In their work they

An example track description is available at: http://labrosa.ee.columbia.edu/millionsong/pages/example-track-description.

3 Dataset

3.1 Data

The million songs dataset is a data set with information about a million different songs [3]. The total dataset, with all the information included, as a size of 280GB. I do not have enough space on my computer to store a so big data set. Therefore I have focused on working on subset of the data set. There exist many different subsets of the dataset with different approaches and with different focuses. On the website of Columbia University, where the million song dataset has its webpage, you can find a 1% subset of the original dataset on 1.8GB. For this project I have worked on that subset, as well as worked on a data set that have the genres on a total of 56 000 songs.

3.2 Features

You can split the features of the dataset into two parts. The track data and the sequence data. From the track data we have features such as duration, tempo, loudness, confidence, song hotttnesss, tempo and year. From the sequence data we have timing information (start, duration of sequence) as well as loudness, pitch and timbre features in each sequence. I have decided to focus on the track data as these features are easier to comprehend and explain. Using the sequence data I would be trying to figure out a deep psychological patterns in what kind of music appeal to people.

3.3 Statistical Distribution

From figure 1, we can see the distrubution between the different genres in the data set. There is clearly a majority of the classic pop and rock genre, while surprisingly the pop genre have few samples. This is a typical problem in data sets that one class have more samples than the other

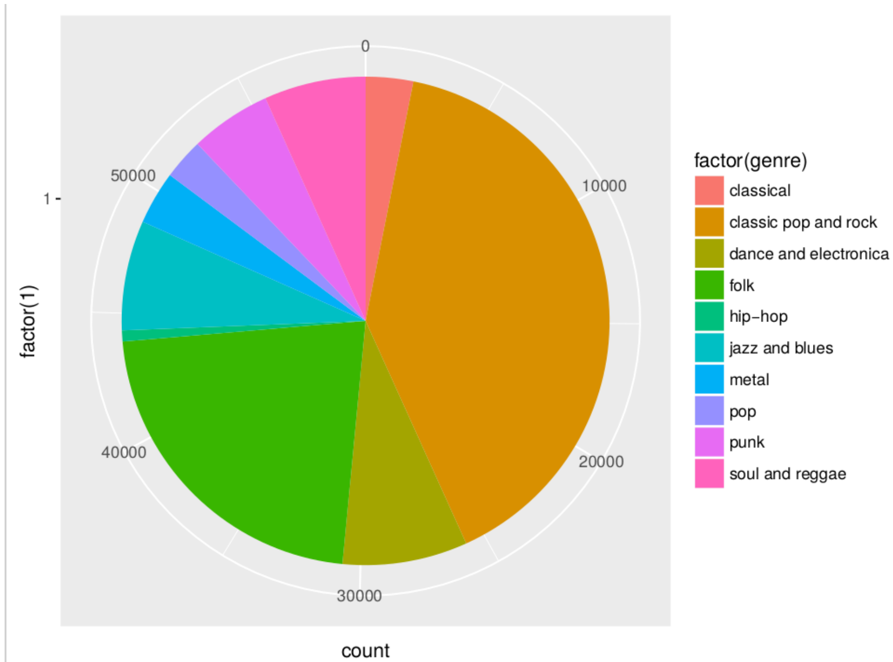

Figure 1: Distribution of the genres in the genre data set

classes, and can lead to a bad results. One solution can be to merge together genres to create fewer and bigger classes, or keep on increasing the total samples with more tracks of the classes with low samples.

4 Methods and Quantitative Results

4.1 Exploratory Data Analysis

One thing I wanted to analyse, was to try and find a correlation between different variables.

From figure 2 we see the correlation between genre classes and loudness values. Not surpirsingly, the metal genre is the loudest genre with most of the samples close to 0 and the punk genre has the second highest loudness value. The figure also makes it clear that it will be hard to predict genre just by loudness value, and each of the genres have a samples with many different values.

4.2 Predictive Analytics Model

This project have focused on predicting three different problems. The first problem is to predict a songs popularity. When predicting the popularity, I used the 25% hottest song (top 25% of the feature song hotttnesss). The second is predicting which year a song is from. Last, we predicted the genre of the song. To classify these three problems we used two different dataset. One was the one percentage subset of the million song dataset, and the other was a genre dataset that had

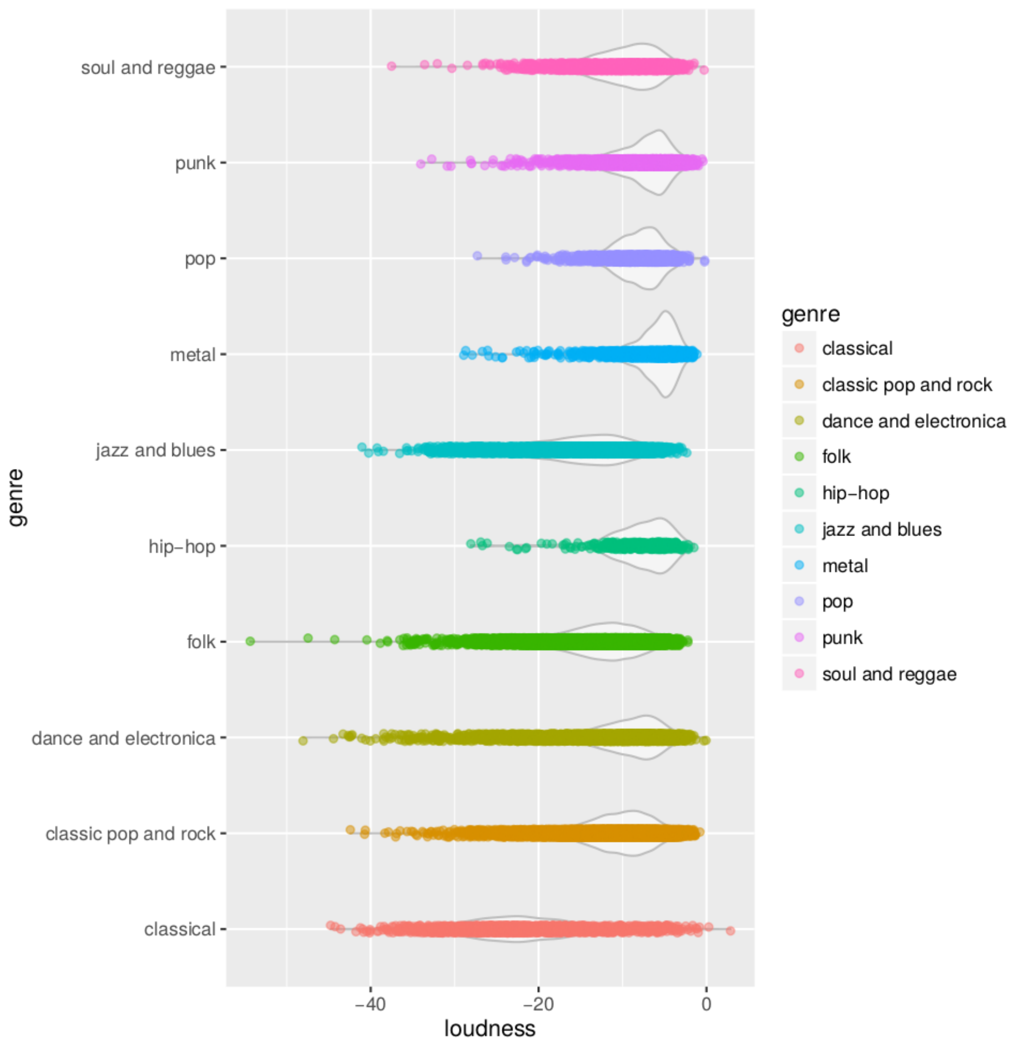

Figure 2: The correlation between Genre class and loudness

already genre as a value. The genre dataset was larger than the one percentage subset, but missed year and song hottness.

For predictive analysis, this project have used three different classification methods. This is used becuase often different classification models can be better for different kinds of problems. By using several different models, we have a better chance of getting a better total results. The three different methods used is **Decision Tree**, **Support Vector Machine (SVM)** and **Random Forest**.

4.2.1 Predicting if a song is popular

4.2.2 Predicting a song release year

4.2.3 Predicting a song genre

4.3 Testing and evaluating the Models

Table 1: Scores on testset for each learning algorithm by problem

Model	Popular	Year	Genre	MEAN
DT	0.846	0.816	0.989	**0.959**
RF	0.847	**0.818**	0.985	0.946
SVM	0.781	0.786	**0.990**	0.958

From table 1 we can see

4.3.1 Predicting if a song is popular

4.3.2 Predicting a song release year

4.3.3 Predicting a song genre

5 Conclusion

6 Acknowledgement

Many thanks to Professor Roger Bohn for providing me with his guidance and advise throughout the project.

References

[1] Tim Ingham, "Global record industry income drops below 15BN dollars for first time in decades", paril 14 2015, abailable at: http://www.musicbusinessworldwide.com/global-record-industry-income-drops-below-15bn-for-first-time-in-history/

[2] Mike Masnick, Yes, "Major Record Labels Are Keeping Nearly All The Money They Get From Spotify, Rather Than Giving It To Artists", Feb 5th 2015, available at: https://www.techdirt.com/articles/20150204/07310329906/yes-major-record-labels-are-keeping-nearly-all-money-they-get-spotify-rather-than-giving-it-to-artists.shtml

[3] 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.

[4] Million Song Dataset, official website by Thierry Bertin-Mahieux, available at: http://labrosa.ee.columbia.edu/millionsong/

[5] http://labrosa.ee.columbia.edu/millionsong/blog/11-2-28-deriving-genre-dataset

[6] James Pham, Edric Kyauk and Edwin Park. "Predicting Song Popularity" (2015) http://cs229.stanford.edu/proj2015/140$_{r}eport.pdf$

[7] Derek Thompson. "The Shazam Effect". The Atlantic (2014). http://www.theatlantic.com/magazine/archive/2014/12/the-shazam-effect/382237/

[8] Tristan Jehan and David DesRoches. "Analyzer Documentation". The Echonest http://developer.echonest.com/docs/v4/$_{s}tatic/AnalyzeDocumentation.pdf$

[9] dede

Appendices

Project code

```
##
## Thomas Astad Sve
##
##

## Create new environment, classification
cls <- new.env()

##
## Load dataset
##

## Predict popularity
#cls$dt <- read.csv("msd_mean_pop.csv")

## Predict genre
cls$dt <- read.table(file = "msd_genre_dataset.txt", header = TRUE, sep = ",", comment = "#", quote=NULL, fill=TRUE)

## Build the training/validate/test datasets.
## Split into 70/15/15 train/test/val
val <- 0.15
test <- 0.15
train <- 0.7

cls$nrow <- nrow(cls$dt)
cat(paste("Number of rows: ", cls$nrow, "\n"))

cls$sample <- cls$train <- sample(cls$nrow, train*cls$nrow)
cls$validate <- sample(setdiff(seq_len(cls$nrow), cls$train), val*cls$nrow)
cls$test <- setdiff(setdiff(seq_len(cls$nrow), cls$train), test*cls$nrow)

##cat(paste("Distribution: ", cls$train ,"/", cls$validate,"/", cls$test, "\n"))


## The following variable selections have been noted.

cls$input <- c("loudness", "tempo", "time_signature", "key", "mode", "duration")

cls$numeric <- c("loudness", "tempo", "time_signature", "key", "mode", "duration")

#cls$target  <- "popular"
cls$target <- "genre"
cls$ident   <- "track_id"
#cls$ignore  <- c("artist_name", "title", "energy", "danceability", "song_hotttnesss")

##
## Decision Tree
##

library(rpart, quietly=TRUE)

## Build the Decision Tree model.
cat(paste("Classifying a decision tree \n"))
cls$dtfit <- rpart(genre ~ .,
    data=cls$dt[cls$train, c(cls$input, cls$target)],
    method="class")


## Predict on test-set
printcp(cls$dtfit)
##cls$dtpred <- predict(cls$dtfit, newdata = cls$dt[cls$test, c(cls$input, cls$target)], type = "prob")
##cat(paste("Decision tree results: ", cls$dtpred, "\n"))

##
## Support vector machine.
##

##library(kernlab, quietly=TRUE)

## Build a Support Vector Machine model.
##cls$ksvm <- ksvm(as.factor(genre) ~ .,
##       data=cls$dt[cls$train,c(cls$input, cls$target)],
##       kernel="rbfdot",
##       prob.model=TRUE)

## Generate a textual view of the SVM model.

##
## Evaluate results
##

library(ggplot2)
```

```
library(plyr)


## plot tree
cat(paste("Saving a plot of the decision Tree \n"))
jpeg('dt_plot.jpg')
plot(cls$dtfit, uniform=TRUE,
          main="Classification Tree for genre")
text(cls$dtfit, use.n=TRUE, all=TRUE, cex=.8)
dev.off()

## Generate a Confusion Matrix
cat(paste("Saving a confusion Matrix of the decision Tree \n"))
jpeg('dt_confusion_matrix.jpg')
plot(cls$dtfit, uniform=TRUE,
          main="Classification Tree for genre")
text(cls$dtfit, use.n=TRUE, all=TRUE, cex=.8)
dev.off()

## Plot count of genre distribution
##distribution <- count(cls$dt, "genre")

##g1<-ggplot(cls$dt, aes(x=factor(1), fill=factor(genre))) + geom_bar(width = 1)

##plot(g1 + coord_polar(theta="y"))

##plot(g1+geom_violin(alpha=0.5, color="gray")+geom_jitter(alpha=0.5, aes(color=genre),
##          position = position_jitter(width = 0.1))+coord_flip())


## Plot relation between loudness and genre
##g<-ggplot(cls$dt, aes(x=genre, y=loudness))

##plot(g+geom_violin(alpha=0.5, color="gray")+geom_jitter(alpha=0.5, aes(color=genre),
##          position = position_jitter(width = 0.1))+coord_flip())
```