PLEASE DON'T STOP THE MUSIC

DSA211 G3 Group 5
Hong Jie • Jessica Low • Lim Yong Jie
Luo Junru • Siti Salihah

OVERVIEW

01

INTRO

Background, Data Cleaning & EDA

SPOTIFY

Spotify adds 20k new tracks daily, but few successfully make it onto music charts.

Our Goal: Predict which songs will make it big based on their composition.

ALL VARIABLES

Discrete	Continuous		Categorical
Sections	Danceability	Instrumentalness	Track
	Energy	Liveness	Artist
	Loudness	Valence	URI
	Speechiness	Tempo	Time_Signature
	Acousticness	Duration_ms	Key
		Chorus_Hit	Mode
			Target (Y Variable)

VARIABLE REMOVAL

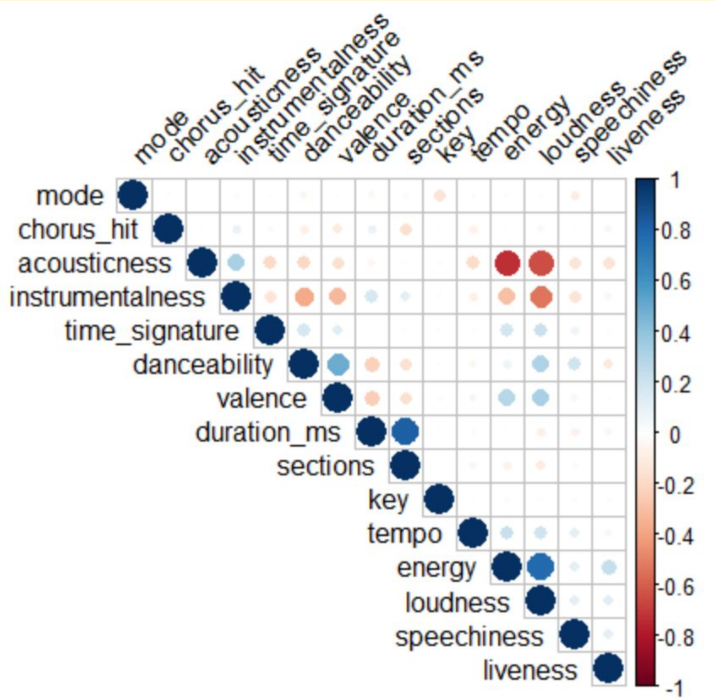
Discrete	Continuous		Categorical
Sections	Danceability	Instrumentalness	Track
	Energy	Liveness	Artist
	Loudness	Valence	URI
	Speechiness	Tempo	Time_Signature
	Acousticness	Duration_ms	Key
		Chorus_Hit	Mode
			Target (Y Variable)

Removed because of data type & too many categories to consider

VARIABLES - SECTIONS & ENERGY

With correlation magnitude threshold 0.7, 3 highly correlated variable pairs were identified:

1. *Duration_ms & Sections* (+ 0.813)
2. *Energy & Loudness* (+0.774)
3. *Acousticness & Energy* (-0.732)



FINAL VARIABLES

Discrete	Continuous		Categorical
Sections	Danceability	Instrumentalness	Track
	Energy	Liveness	Artist
	Loudness	Valence	URI
	Speechiness	Tempo	Time_Signature
	Acousticness	Duration_ms	Key
		Chorus_Hit	Mode
			Target (Y Variable)

Removed because of high collinearity

02

REGRESSION

Logistic Regression,
Lasso Regression &
Model Analysis

LOGISTIC REGRESSION

Reasons for model

- Binary Y/dependent Variable
- Explaining relationships between Independent variables and dependent variable

However ….

- All variables were used without variable selection
- Problem of overfitting

Thus ….

LASSO REGRESSION

Our Model

Danceability
+
Loudness
+
Mode
+
Instrumentalness
+
Liveness
+
Valence
+
Tempo
+
Duration_ms
+
Time_signature

=

Odds of a song charting

Reason for Model

- Reduce number of variables
- Reduce variance

Model Evaluation-Confusion Matrix

	Actual Charted	Actual Uncharted
Predicted Charted	2932	1171
Predicted Uncharted	252	1904

Sensitivity = **92.09%**

Specificity = **61.92%**

Overall Error Rate = 22.74%

Prediction Accuracy = **77.26%**

False Positive Rate = 38.08%

False Negative Rate = 7.91%

03

RECOMMENDATIONS

Insights & Analysis

RECOMMENDATIONS

Increasing Priority

- Low instrumentalness
- High danceability
- Low valence
- Low liveliness
- Low loudness
- Mode – Major modality
- Time signature – 4
- Faster tempo
- Short song duration

RECOMMENDATIONS

High Danceability

Low Instrumentalness

Lyrical dance songs with fast tempo

RECOMMENDATIONS

Low Valence Songs ☹️ > High Valence Songs 😃

Sad, soft songs 👍

OTHER INSIGHTS

Contradiction between valence & mode

- Low valence > High valence = Sad songs > Happy songs
- Major modality > Minor modality = Happy songs > Sad songs

OTHER INSIGHTS

- Preference of studio-recorded tracks over live tracks
- Magic of Signature 4
- Preference of shorter songs over longer ones

IN SUMMARY

Fast, danceable songs

Sad, soft songs

PRACTICAL APPLICATIONS

- Using model to gauge charting probability before launch
- Song priority for album

04

CONCLUSION

Limitations & Conclusion

LIMITATIONS

Omitted the variable “artist”

- Current: general model for all artists, but in real life, an artist’s popularity may play a large role in a song being charted or uncharted.
- Potential variable for future studies: Artist ranking

LIMITATIONS

Omitted the variable “key”

- For reasons previously mentioned
- Dummy variables can be created to further explore the potential relationship between song key & the probability of a song charting.

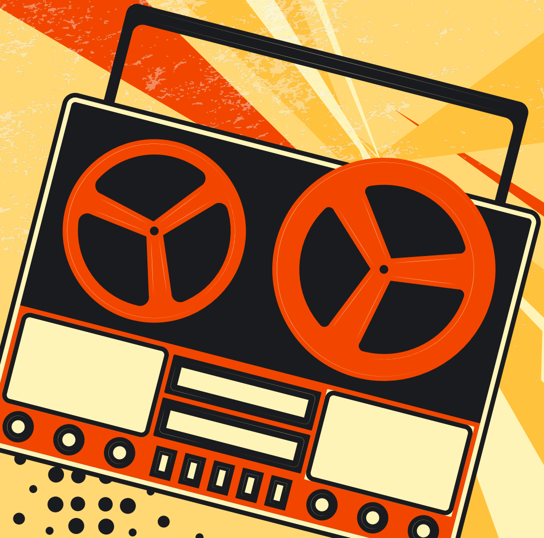
ARTISTS' CHOICE

Fast, danceable songs

Sad, soft songs

THANK YOU :)

Do ask us any questions
you may have!



R CODES – LIBRARIES


```
library(leaps)

library(glmnet)

library(dplyr)

library(corrplot)

library(caret)

library(tidyverse)
```

R CODES - EDA


```
spotify<-read.csv("C:/Users/YJ Lim/Desktop/Sportify Data/spotify-of-10s.csv")

summary(spotify)

dim(spotify)

#Removing categorical variables (track,artist,uri) and removing duplicates

library(dplyr)

spotify1 <- spotify %>% select(-track, - artist, -uri) %>% unique() #this code only works when there is dplyr
```

R CODES - EDA


```
#Removing discrete and target variable

pairs(spotify1%>%select(-time_signature,   -sections,   -key,
-mode, -target))

#Running correlation analysis

#Needs to get rid of target variable only from the dataset

correlation <- cor(as.matrix(spotify1[-16]))

round(correlation,3)

#visualisation

#install.packages("corrplot")

library(corrplot)

corrplot(correlation, type = "upper", order = "hclust",

        tl.col = "black", tl.srt = 45)
```

R CODES - EDA


```r
#Remove energy + sections (to avoid problem of multicollinearity)

library(dplyr)

spotify2<- spotify1%>%select(-energy, -sections, -key) %>% mutate(time_signature_dummy = ifelse(time_signature == 4,1,0)) %>% select(-time_signature)
```

R CODES – BINARY LOGISTIC REGRESSION


```
#Conducting a logistic regression with all the variables

spotifyALL<-glm(target~.,data=spotify2,family=binomial )

summary(spotifyALL)
```

R CODES – LASSO REGRESSION


```
#Lasso Regression
trainp <- sample(1:nrow(spotify2),nrow(spotify2)/2)
testp <- -trainp
#Get half of the data set as training set
spotify2.train <- spotify2[trainp,]
#The rest is training set
spotify2.test <- spotify2[testp,]
train.x <- model.matrix(target~., data=spotify2.train)[,-1]
train.y <-spotify2.train$target
test.x <- model.matrix(target~., data=spotify2.test)[,-1]
test.y <- spotify2.test$target
grid <- 10^seq(10, -2, length=100)
```

R CODES – LASSO REGRESSION


```
#Getting the lambda with smallest CV error

lasso.mod <- cv.glmnet(train.x, train.y, alpha=1,family="binomial", lambda=grid)

lambda.lasso <- lasso.mod$lambda.min

lambda.lasso

#Getting the prediction from the best lambda

lasso.pred <- predict(lasso.mod, newx=test.x, s=lambda.lasso)

#Getting the mse/test error

mean((test.y-lasso.pred)^2)

x <- model.matrix(target~., data=spotify2)[,-1]
y <- spotify2$target

#Using all data to refit model

out.lasso <- glmnet(x,y, alpha=1, lambda = grid, family="binomial")

#Get final model with y-intercepts by using the best lambda

lasso.coef <- predict(out.lasso, type="coefficients", s=lambda.lasso)[1:13,]

lasso.coef[lasso.coef !=0]
```

R CODES – CONFUSION MATRIX (LASSO REGRESSION)


```
#creating a function to calculate log_odds for each data point
library(tidyverse)
log_odds <- function(spotify){
        log = lasso.coef[1] + (spotify$danceability*lasso.coef[2]) +
(spotify$loudness*lasso.coef[3]) + (spotify$mode*lasso.coef[4]) +
(spotify$instrumentalness*lasso.coef[7]) + (spotify$liveness*lasso.coef[8]) +
(spotify$valence*lasso.coef[9]) + (spotify$tempo*lasso.coef[10]) +
(spotify$duration_ms*lasso.coef[11]) +
(spotify$time_signature*lasso.coef[13]) }
#function to convert log_odds to probability
logit2prob <- function(logit){
  odds <- exp(logit)
  prob <- odds / (1 + odds)
  return(prob)}
logodds <- log_odds(spotify2)
prob <- logit2prob(logodds)
prob_df <- data.frame(p= prob)
```

R CODES – CONFUSION MATRIX (LASSO REGRESSION)


```r
#add column of probability from lasso regression
spotify4<- add_column(spotify2, p =prob_df$p)
#new column where u = 1 if p>= 0.5, and u = 0 if p < 0.5)
spotify5 <- add_column(spotify4, u = ifelse(spotify4$p >=0.5,1,0))


pred_1_actual_1 <- nrow(spotify5 %>% filter(target == 1 & u ==1))
pred_0_actual_0 <- nrow(spotify5 %>% filter(target == 0 & u ==0))

pred_1_actual_0 <- nrow(spotify5 %>% filter(target == 0 & u ==1))
pred_0_actual_1 <- nrow(spotify5 %>% filter(target == 1 & u ==0))

sensitivity <- pred_1_actual_1 / (pred_1_actual_1 + pred_0_actual_1)
specificity <- pred_0_actual_0 / (pred_0_actual_0 + pred_1_actual_0)

overall_err = (pred_1_actual_0 + pred_0_actual_1) / nrow(spotify2)
Prediction_Accuracy = 1-overall_err
```

R CODES – CONFUSION MATRIX (LASSO REGRESSION)


```
false_negative_rate <- pred_0_actual_1 / (pred_1_actual_1 + pred_0_actual_1)
false_positive_rate <- pred_1_actual_0 / (pred_0_actual_0 + pred_1_actual_0)

sensitivity
specificity
overall_err
Prediction_Accuracy
false_negative_rate
false_positive_rate

pred_1_actual_1
pred_0_actual_0
pred_1_actual_0
pred_0_actual_1
```

References

Music Business Worldwide (2018). In A&R, 'gut vs. data' isn't a binary choice. Retrieved from https://www.musicbusinessworldwide.com/in-ar-gut-vs-data-isnt-actually-a-binary-choice/

The Overman (2019). The Spotify Hit Predictor Dataset (1960-2019). Retrieved from https://www.kaggle.com/theoverman/the-spotify-hit-predictor-dataset#dataset-of-10s.csv