Team: Adarsh Salapaka, Harsh Tandon, Wenping Du, Xiaoyu Guo, Xiyu Wu

Spotify Popularity Predictor & Explorer

Introduction:

DJ Khaled boldly claimed to always know when a song will be a hit. We decided to further investigate this by asking three key questions: Are there certain characteristics for hit songs, what are the largest influencers on a song's success, and whether old songs even predict the popularity of new songs? Predicting how popular a song will be is no easy task. Every song has key characteristics including lyrics, duration, artist information, temp, beat, loudness, chord, etc. To answer whether a song will be popular based on these key characteristics, we made use of:

a) Spotify Tracks DB provided by Kaggle

b) Machine Learning prediction models - Random Forests and Linear Regression

Note: At the end of this document, you will find the link to the app and steps on how to run it.

Problem Statement:

The objective of this project is to do exploratory data analysis using the R shiny app and determine whether the popularity of a song depends on the relationship between multiple variables that Spotify provides. We want to figure out the effect, each of the variables has on the popularity. We got multiple attributes for a single song such as - **danceability, energy, tempo and valence**. We decided to further investigate by asking the key question: Are there certain characteristics for hit/popular songs?

Dataset:

The dataset that we used for this analysis has **228,160 rows with 18 rich attributes** for each of them as explained in the introduction. We had to do the regular data cleaning and wrangling process to go forward with it.

Data Cleaning and Wrangling:

- Attributes such as *'time_signature*' is supposed to have 5 categorical levels showing **0/4 beats, ¼ beats, 2/4 beats, ¾ beats, and 4/4 beats**. The value '*x/4*' is misinterpreted by .csv file and is stored as '*1st April, 2nd April*', etc. This attributed was transformed into a categorical value having its true nature of values.
- Column name '*genre*' on .csv file is read differently in Macbooks and Windows. Windows laptops read it as '*ï..genre*', while Macbook reads it as '*genre*'. A small snippet of code is added in the main file to take care of this problem.
- Attributes like **'*mode*', '*key*', and '*time_signature*'** have to be converted into numeric type for our prediction models.

The new libraries used:

1. **Caret:** The caret package (short for **C**lassification **A**nd **RE**gression **T**raining) is a set of functions that attempt to streamline the process for creating predictive models. The package contains tools for:
   a. data splitting
   b. pre-processing
   c. feature selection
   d. model tuning using resampling
   e. variable importance estimation

2. **DT:** The R package **DT** provides an R interface to the JavaScript library **DataTables**. R data objects (matrices or data frames) can be displayed as tables on HTML pages, and Data Tables provides filtering, pagination, sorting, and many other features in the tables.

Machine learning for predictions:

Random Forests

- It can be used for both classification and regression tasks.
- It has provided accuracy **(9% more accurate than Naive Bayes and around 11% more accurate than Linear Regression)**.
- Random forest classifier will handle the missing values and maintain the accuracy of a large proportion of data. It has the power to handle a large data set with higher dimensionality which was really helpful for our dataset.
- However, due to the extremely high latency (or long loading time) in shiny for each prediction, we decided to use linear regression.

Linear Regression

- Linear Regression is a machine learning algorithm based on supervised learning. It performs the task to predict a dependent variable value (which is the popularity of a song in our project) based on given independent variables (song features).
- Linear Regression model is saved as a '.rda' file. This file is nothing but our trained model. Before running our shiny app, we load this trained model into global environment. Once loaded, it predicts popularity based on inputs given by user.

Limitations or issues faced:

Training our dataset on Random Forest with optimized number of trees and mtry nearly took 6 hours on a 16GB RAM, with Intel core i7 machine. For a new user, loading the trained model nearly takes 30-40 minutes depending on system configurations. Thus, faced with hardware limitations, we decided to run our predictions using Linear Regression. By doing this we were losing around 11% accuracy, but this loss seemed acceptable when high latency comes into the picture.

Learnings and conclusions:

We found the following results where we were able to predict the success rates of the popularity and then comparing those results with the Spotify Hotness Meter. The results predicted by our model were quite in line with Spotify and the accuracy provided by the Linear Regression algorithm were 53% overall for the test dataset vs training dataset.

Song name - Artist	Predicted Success Rate	Spotify Hotness (0-1)
Stressed Out - Twenty One Pilots	**53%**	0.83
Sideshow - Blue Magic	**34%**	0.52
God's Plan - Drake	**57%**	0.87
Happy - Pharrel Williams	**53%**	0.74
Anaconda - Nicki Minaj	**48%**	0.65

Team: Adarsh Salapaka, Harsh Tandon, Wenping Du, Xiaoyu Guo, Xiyu Wu

Link for project: https://drive.google.com/file/d/1DtMw4ijDqwTmZAriFyJT_nPT64Yt7F1N/view?usp=sharing

*Link for project Video **(Part 1 - Global.R):***
https://drive.google.com/file/d/1r_Ta-04yEoo_dIQRCKWmdDAGMPoaFIN9/view?usp=sharing

*Link for project Video **(Part 2 - Shiny App):***
https://drive.google.com/file/d/1zmkYzACcOB4m4uf0M_MQhzWevZILZ6h-/view?usp=sharing

Steps on how to run the app:

Step 1: Download the zip folder from here and unzip the file into a folder in your system.
Step 2: Open "***Global.R***" file in R studio.
Step 3: Set working directory of this file with a forward slash at the end of address:
For example: setwd("D:/1st Qtr Study Material/R/Final/HTandon/Spotify**/**")
Your address should look something like this: "~address**/HTandon/Spotify/** "

Step 4: Run "***Global.R****"* file. Withing 15 seconds, R Shiny app will open.
Step 5: Play around with the app.

About the files inside zip folder:

1. **Global.R :** Loads the dataset into the environment. Does data cleaning. Loads trained model for predictions. Runs the app.
2. **LinearRegressionModel.rda** : ML model trained on our dataset. We train the model once and use it directly to save processing time.
3. **Popularity_PredictorExplorer Folder**: contains 2 shiny files:
    a. **ui.R**: contains code for UI of our shiny app
    b. **server.R**: contains functionality code of our shiny app
4. **SpotifyFeatures.csv**: CSV file containing our dataset