Predicting the popularity of songs based on Spotify attributes

TARTU ÜLIKOOL
arvutiteaduse instituut

Birgit Sõrmus
Kaspar Kadalipp

IT Akadeemia
toetab Skype™

Introduction

Since music is a huge part of a lot of people's lives, including ours, we figured there could be no better topic for this project.

The goal was to take different datasets from Kaggle that all include songs with their attributes and try to predict if a song would be popular or not based on those features. We found a very large dataset with over 200 000 songs, so we split that into a trainset and a testset. For testing, we also used tops of 2017 and 2018, assuming that those were all popular songs.

Another goal we set based on the tops of 2017 and 2018 was to analyze the features of those songs and see whether the music people listen to had changed in a year or not.

Data

All of the data we used was taken from Kaggle.

For the comparison we used two datasets, one including the top 100 of 2017 and the other including the top 100 of 2018. Both datasets include different attributes for each song.

The third dataset that was used was a large dataset with over 200 000 songs that included the same attributes.

The attributes were acousticness, danceability, duration_ms, energy, instrumentalness, key, liveness, loudness, mode, speechiness, tempo, and valence.

Preprocessing

Since the datasets didn't have any null values, we didn't have to worry about dealing with those. However, the large dataset had some duplicate rows, so we had to remove them.

Although the attributes in the different datasets were the same, there were some that were named slightly differently, so we had to take that into consideration.

We also noticed, that the best results were obtained when we only used songs with popularity over 60 for popular and songs with popularity under 40 for non-popular songs for training, so we discarded those that were inbetween.

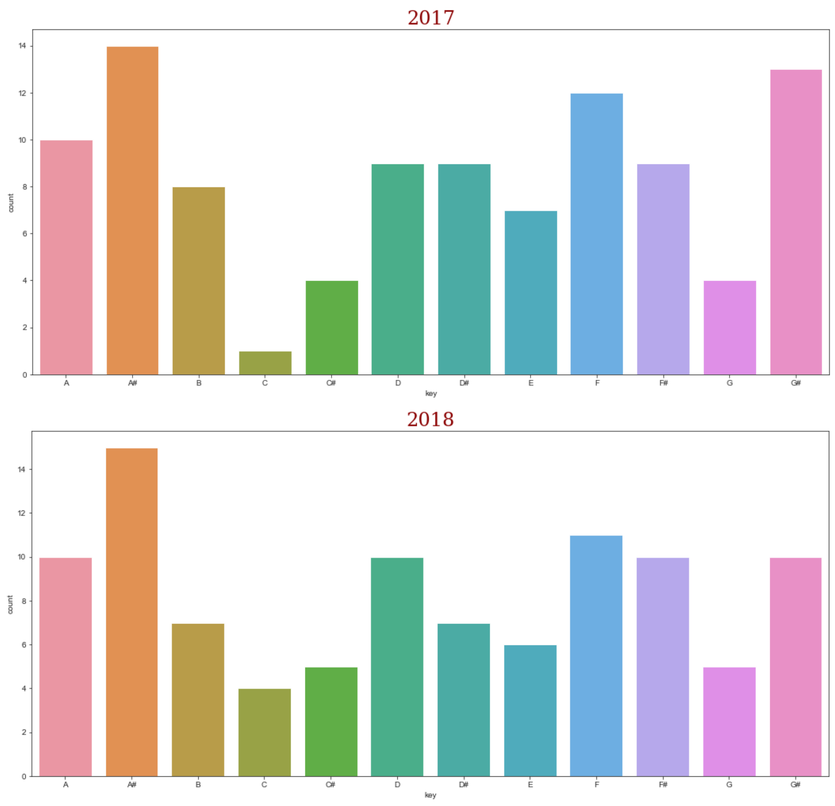

Figure 1. different key counts for 2017 and 2018

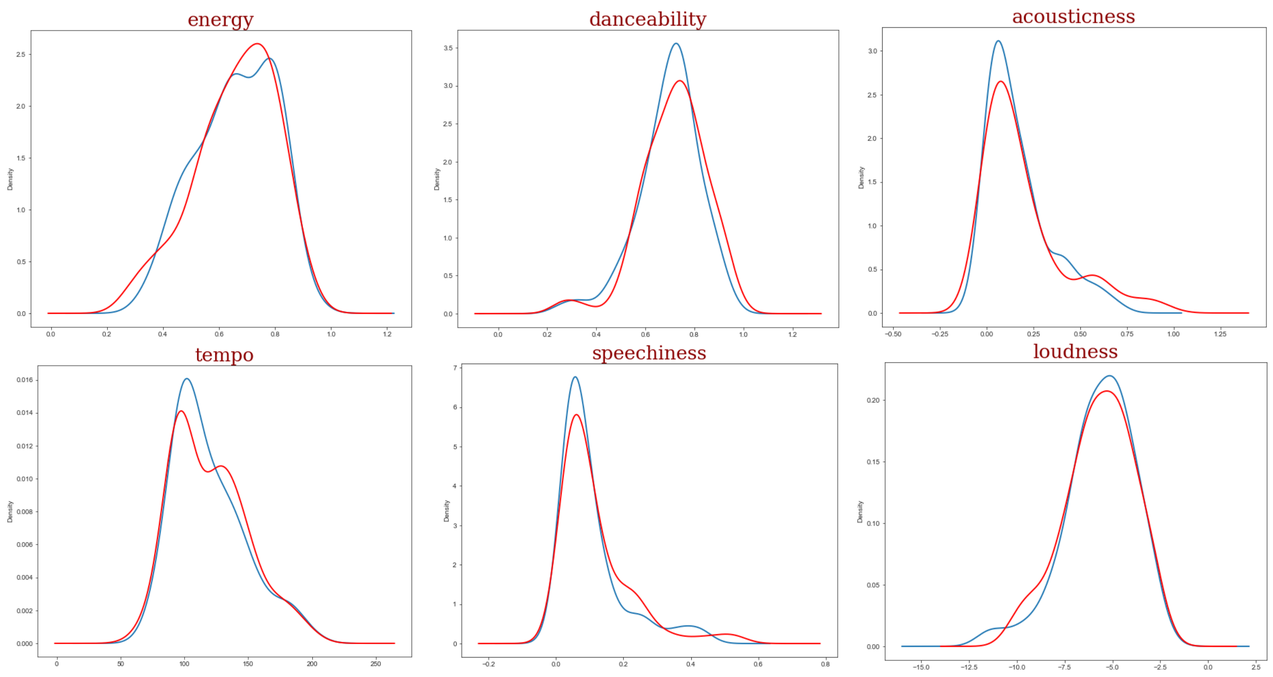

Figure 2. Density plots for different features, 2018 is coloured red, 2017 blue

Machine Learning

The models we used for training were random forest, K nearest neighbors and decision tree. To get better results, we tried different parameters, but parameter tuning didn't improve results as much, so for the most part, we used default parameters, only changing n_neighbors for KNN and n_estimators for RF to 100.

The best results were obtained using the random forest classifier. It had an accuracy of 86.7% on the large dataset and 70% and 84% on the tops of 2017 and 2018 respectively.

	Large dataset	2017 top	2018 top
RF	86.7%	70%	84%
KNN	79.5%	60%	74%
DT	79.5%	66%	76%

Figure 3. Accuracies of the three models we used on different datasets.

Results

- Comparing 2017 and 2018, it was clear that the differences in attributes weren't very large
- For predicting classes, it was best to remove songs that had popularity somewhere in the middle (so 40-60 out of 100)
- The model that got the best results on all datasets that we used was random forest classifier

Source code:
https://github.com/birgitsormus/spotify-predictions