Predicting Song Popularity on Spotify in Spain

Erika Gutierrez & Tobias Pfeiffer

Barcelona School of Economics

December 21, 2022

Contents

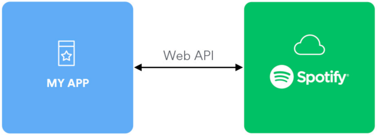
Introduction

Team Organization

- Erika
  - Previous coding experience
  - Set up the library structure
  - Scraped data from Spotify
  - Wrote unit test on the data part
- Tobi
  - No coding experience
  - Created library functions
  - Took care of the modelling
  - Wrote unit test on the feature part

Data Source

- Spotify API
  - Free clean data on albums, artists, songs, genres, etc
- Endpoints used:
  - Get categories used to tag items in Spotify.
  - Get a list of Spotify playlists tagged with a particular category.
  - Get full details of the items of a playlist.
  - Get audio feature information for multiple tracks.



Pipeline

Data Acquisition Flow

1. Get the popular songs
   - We classified songs that are popular in Spain as the ones that are on the Spotify curated ”Top 50 - Spain” playlist.
2. Get the ”not popular” songs.
   - We sampled 10 songs from 190 playlists representing 19 different genres.
3. Create final dataset
   - For each song we have: dancebility, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, duration (in milliseconds), and time signature.
   - Dummy variable for whether it was the in the Top 50 playlist or not.
   - Duplicates were removed.
   - Final dataset has 1940 songs.

Feature Creation

- Happy:
  - Songs with a Valence above 0.5 are classified as a ”happy song”
- Genre:
  - BPM (beats per minute) are an important part of music composition
  - BPM can we used to classify the genre of a song
  - Hip Hop: 85–95 BPM
  - Reggaeton: 90-100 BPM
  - Pop: 100-140 BPM
  - Techno/House/Electro: 120+ BPM
- Duration:
  - Spotify lists the duration in milliseconds which is hard to relate to and inflates the units
  - Our third feature function adds the duration in minutes or seconds and drops the original duration

Machine Learning Model Used

- Random Forest
- Using test data to hyper tune the parameters
  - $max\_depth = 6$,
  - $min\_samples\_split = 4$,
  - else default
- Target: *Top50* $\mathbb{1}\{$ if the song is in the Spanish top 50$\}$

Machine Learning Model Results

- Accuracy Score: 0.97
- BUT no song was predicted to be in the Top50

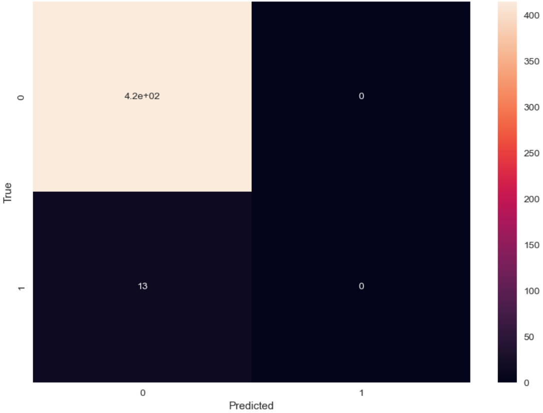
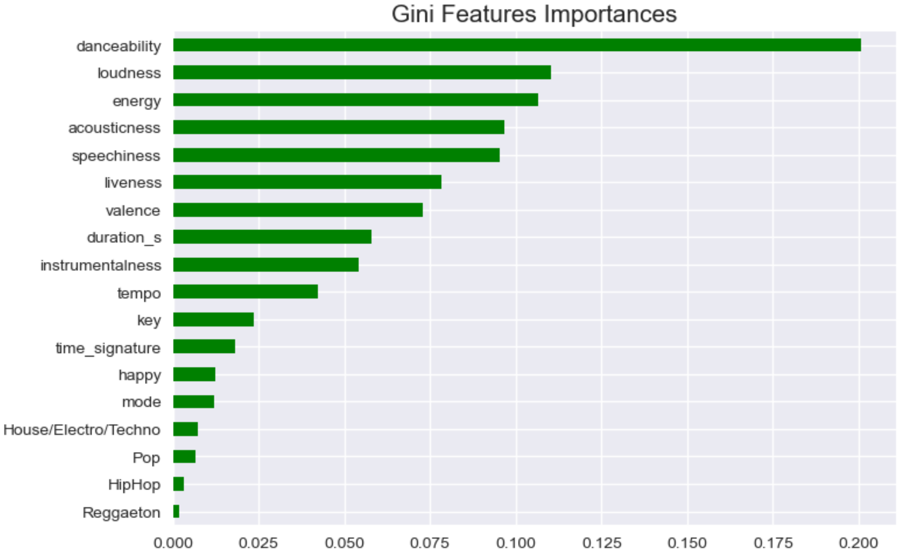


Figure: Confusion Matrix

Machine Learning Model Results



Conclusion

Next Steps

- Adjust the model
  - Change the precision score
  - Use a better tree model (e.g. xgboost)
- Adjust song sampling specifications
- Add new features
  - Scrape Twitter, Reddit etc. for trending songs
  - Scrape TikTok
  - Consider the artist
- Account for previous chart listings
- Test other countries

How a New Member Can Join

- /PopularityContest
  - /Load
    - spotify.py
    - data.py
    - config.py
  - /Process
    - feature1.py
    - feature2.py
    - feature3.py
  - /Model
    - split.py
    - bestmodel.py
- test.ipynb
- /test
  - /Load
    - test_spotify.py
  - /Process
    - test_features.py

- Clone our repository on github
- Test other countries: edit data.py
- Adjust song sampling specifications: edit spotify.py
- Add new features: create a feature.py file in the Process folder

Thank you!