MASTER THESIS

2019/2020

STUDENT NAME: Stanislas KIRGENER DE PLANTA

RESEARCH TOPIC

Predicting song popularity combining audio features with metadata

TUTOR'S NAME: Julien FOUQUAU

CONFIDENTIAL No Yes

Table of Contents

Abstract

In this research, we will try to apply several Machine Learning algorithms to see if we can actually classify a song as a Hit or a Non-Hit based on its audio features as well as other external factors. First, we will describe the music market to understand the potential business benefits that Machine Learning could have. Then, through a literature review, we will analyse the methods applied, and progress made, from the use of basic song features to the addition of external factors taking into account other dimensions of song popularity. This will help us drawing up the main questions of our research field. To tackle these questions, a database of Spotify's Top 2018 and 2019 songs will be constructed. We will define a continuous and a categorical variable for popularity to test Linear Regressions with various feature sets as well as the following classifiers: Logistic Regression, K-Nearest Neighbours, Random Forest, Support Vector Machines and a Multilayer Perceptron with one hidden layer. For both tasks, the results are compared to a dummy model. Finally, we will analyse our results as well as the limits of our experimentation. We conclude that the Multilayer Perceptron yield the best results with features selected by Random Forest and that metadata are essential.

Key words

Machine Learning, Hit Song Science, Metadata, Audio Features, Hit, Non-Hit, Classification and Regression.

Acknowledgement

I would like to thank all the people who supported me during my first data project. It gave me the opportunity to deepen my understanding of basic Machine Learning concepts as well as applying them to a field that I am passionate about: the music industry. Then, I would like to thank my tutor Julien Fouquau for allowing me to work on this subject and answering many of my questions. I also thank Spotify for their great API and the huge amount of data they provide for free. Finally, I would like to thank Ulysse Couerbe who gave me a great list of online courses to start learning the basics of Machine Learning, that I was then able to apply on this subject.

Introduction

Since the beginning of the 21st century, the music industry has been evolving a lot, from being troubled by the emergence of P2P sharing platforms to being reinvented through new distribution channels brought by the birth of streaming platforms.

The digital era, characterized by the rapid spread of the Internet worldwide and the development of new compression formats as well as better storage solutions, successfully turned music into an immaterial element, more and more accessible to consumers. Today, it has become quite simple to get millions of songs right at our fingertips.

Moreover, this era translates not only into the distribution but also the production of songs. It has become easier to do so, using software and other digital tools, resulting in a growing number of songs and labels. This raises the question of how an artist, or a song can stand out in the crowd and become popular? In other words, how can good artists be detected in this digital cacophony?

Indeed, every year, the music industry is spending billions in discovering and training new artists with only a few of them reaching the top, allowing labels to recover their financial investments. Being able to leverage advanced analytics tools, such as Machine Learning algorithms, to determine whether or not a song has the potential to become a Hit (and a cash machine) would definitely be valuable for record companies, notably to help them save money and better manage their investments.

Being a cultural product, music tastes are very diverse across the globe and evolve constantly. As explained by (Salganik, Dodds, & Watts, 2006), in cultural markets, success of products is hard to explain and uncorrelated to a better knowledge of the industry because individuals are influenced by their peers. According to this research, the success of a Hit would be explained mostly by social factors, making it difficult to use Machine Learning tools to predict whether or not a song will become popular.

The main research questions that we address here are the following**:**

- **To which extent Machine Learning can be used by labels to determine potential future Hits?**

- **What are the most promising algorithms?**
- **To which degree audio features can be used to explain song popularity?**
- **Which factors have the best predictive power between audio features, artist metadata and song metadata?**

Our work focuses mostly on testing new features that might improve the results and on validating or not the hypothesis that those tools are useful for the music industry. To do so, we will have to reflect on the definition of the popularity of a song as well as choosing which variables to add in our model to improve the results. We will also compare several sets of variables to understand the factors that have the best predictive power and therefore the largest impact on song popularity.

We will start by studying the music market, its evolution and its actors to understand the potential financial benefits of using such tools for the music industry. In particular, we will deal with the flat growth of the market in terms of revenues making it necessary for companies in the sector to improve their cost structures.

The literature review will be useful to conduct an analysis of the statistical methods, the variables and the definitions of a song popularity that have been used in this research field. This will help us orientate our methodology to answer our research questions.

Then, we will describe the methods we used, mainly to create our dataset, select the most promising features and improve the results of our algorithms. We will also analyse and discuss our results to answer the research questions.

Finally, we will conclude by summarizing our experiment, giving our key takeaways and identifying guidelines for further reflection on the subject.

1. The Music market and its evolution

1.1. A brief overview of the recorded music market

1.1.1. The era of digital

The last decades have seen significant changes in the music sector. With the rise of streaming platforms to replace CDs as the new distribution channel and the use of playlists instead of albums to consume music, the industry managed to weather the P2P crisis and has been growing again since 2015.

In 1999, Shawn Fanning, John Fanning and Sean Parker co-founded Napster, a Peer-to-Peer (P2P) file sharing platform. Combined with the development of the mp3 compression format the same year, the network became more and more popular to transfer song files. Even if the company was sued, the concept of illegal P2P music sharing went mainstream and many Napster's copy cats were created such as LimeWire, Morpheus, BearShare or eMule. In 2003, Apple tried to come up with a solution: iTunes, the first legal digital platform allowing users to directly download mp3-formatted songs on their personal computer or other devices (like the iPod). However, the solution proved to be inefficient to counterbalance the negative impact of P2P platforms on the music industry – the constitution a large music library being too expensive. Around 2007, the concept of media streaming emerged with the foundation of YouTube in 2005, the development of Pandora – a web radio created in 2000 – or SoundCloud in 2007. However, the first time people discovered the music streaming model that we are all familiar with, was in 2006, with the establishment of Spotify. It was the first company in the music streaming sector with a catalogue large enough to meet the users' expectations. Since then, many other platforms have been developing competing directly with Spotify – Deezer (2007), Amazon Music (2007), Tidal (2014), Apple Music (2015) or YouTube Music (2015) – others developed for specific geographies with the advantage of local catalogues – KKBox (2004), Yandex Music (2010), Simfy Africa (2010) or Tencent Music (2016) – or operating on a specific market within the music industry, for audiophiles and specific tasks – 8Tracks (2008) or iHeartRadio (2008) – (Soundcharts Blog, 2019).

With this shift towards digital solutions, music has become more and more accessible and consumed. In 2015, according to (Nielsen, 2017), American citizens spent on average 23.5 hours per week listening to music compared to 32.1 hours in 2017. With streaming platforms, users can access millions of songs and discover music from different countries or other genres. Moreover, production has become easier and nowadays, everyone can create its own song without knowing anything about composition and upload it on Soundcloud for instance. Another interesting point is that music listeners are increasingly connected to the artists they follow. Digital platforms can provide users with information regarding concerts, tickets or even specific contents for the biggest fans. The industry seems to focus more and more on the artist condition at the expense of the Major Labels, which is evidenced by Deezer's new compensation solution that aims at fostering less-known artists that are not signed on Major Labels. The idea is that the subscription paid by a given user will be used to compensate the artists one is listening to[1].

However, the conversion of music into a mass consumption product raises issues. With the plurality of distribution channels (streaming platforms, physicals, live concerts, etc.), there has been a rising concern in copyright management and royalties' payments with the artists denouncing an unfair split of the revenues originating from streaming platforms subscriptions – in 2019, Eight Mile Style, Eminem's music publisher decided to sue Spotify for $120m unpaid royalties[2]. As a matter of fact, Spotify is trying to reduce the costs associated with royalties' payments that currently represent approximately 70-74% of the revenues[3], which is a source of conflict between the right holders and the distributor (Spotify) with the artist being in the middle. Right holders request an access to music consumption data in order to better manage their revenues. Big Data technologies are at the heart of those claims with the example of Music Recognition Technology (MRT), which is able to recognize a given track through a sound – a technology used by Shazam.

That last decade saw the rescue of the industry as well as the conversion of music into a mass consumption product with the streaming technology. In the wake of this rebirth, new concerns arose with the artist being in the heart of the reflection around a fair royalties' payment system and Big Data technologies being potential solutions.

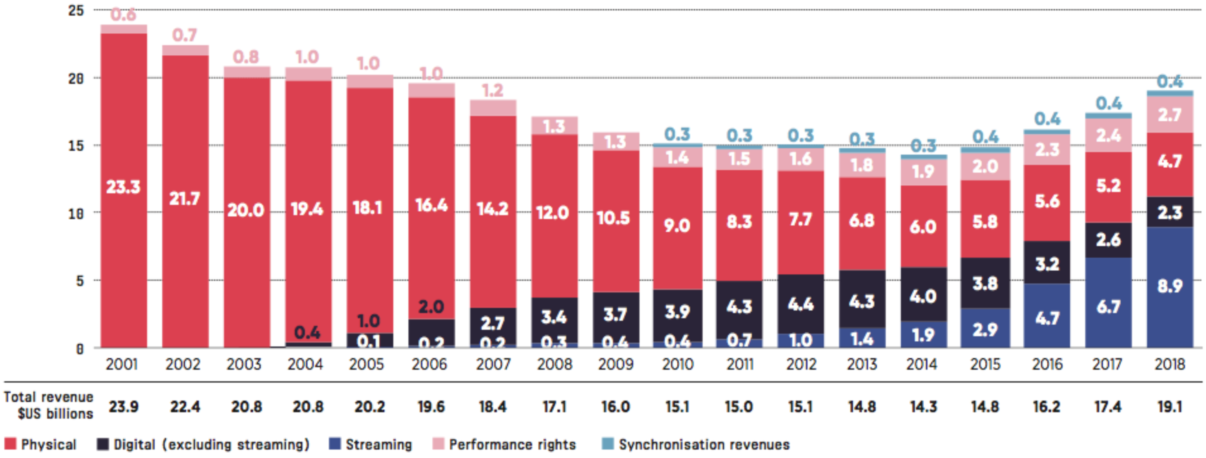
[1] https://www.deezer.com/ucps
[2] https://www.bloomberg.com/news/articles/2019-08-22/spotify-sued-by-eminem-publisher-over-billions-of-unpaid-streams
[3] Spotify's 2019 Anual Report

1.1.2. The expected evolution of the Recorded Music Market

In accordance with what we have detailed in the previous part, streaming is obviously the main driver of growth for the global recorded music industry, representing 47% of record companies' revenues in 2018 (IFPI, 2019). The market is still growing even though growth is slowing down gradually because of a decrease in revenues from previous distribution channels. In 2018, it grew up by 9.7% to reach $19.1bn worldwide, which can be split into the following sub-markets (IFPI, 2019):

- **Paid streaming and ad-based subscriptions** (37% and 10% respectively of 2018's market value): as of today, revenues from streaming platforms have become the main driver of growth for the industry. In 2018, they increased by 34% including a 32.9% increase in paid streaming.
- **Physical channels** (25% of 2018's market value): sales have been decreasing for almost 13 years in favour of digital channels even if the vinyl format seems to be more and more popular nowadays. In 2018, revenues decreased by 10%.
- **Performance rights** (14% of 2018's market value): these are generated when recorded music is used by broadcasters or for public venues (Live music, etc.). In 2018, they increased by 9.8%.
- **Download platforms** (12% of 2018's market value): among the digital segment, download revenues are clearly losing ground to streaming. In 2018, they went down by 21.2%.
- **Synchronisation revenues** (2% of 2018's market value): these are revenues from using music in advertising, films, games and TV. In 2018, they grew up by 5.2%.

Figure 1: Global recorded music industry revenues, 2001-2018 in $bn, (IFPI, 2019)



We will now try to understand how record companies capture value from the streaming distribution channel with the royalty system. Every time a song is played on a platform, the followings have to be paid (Soundcharts Blog, 2019):

- **Mechanical Royalties**: paid to songwriters each time a song is played.
- **Performance Royalties**: paid to songwriters when a song is publicly performed. In the streaming model, a song cannot be owned so each music is considered to be played publicly.
- **Recording Pay-out**: this represents the largest share of what is paid by streaming companies and it is paid to copyright owners (label, distributor or artist).

In addition, it must be noted that payments can be different for a same song. For instance, an ad-based user will generate less revenues for the record company than a paid subscriber. This happens to be the same with the countries where the song is played. Therefore, it is challenging to find an agreement on how much money should be paid or collected by both sides – the record company and the streaming platform. Currently, it is estimated that previously detailed costs represent 60%-70% of Spotify's revenues, which is why streaming companies are constantly renegotiating their contracts with record companies as well as integrating other contents – the example of podcasts with Spotify – to reduce their dependence to Majors. (Soundcharts Blog, 2019).

At the same time, streaming platforms are also facing other challenges on the revenue side. According to (MIDiA, 2019), the global streaming music market is expected to reach a plateau at a 7% growth rate in 2026, which can be explained by the fact that mature countries like the US, the UK, Sweden, the Netherlands or Australia have already reached their maximum penetration rate – 39.2% in 2018 and expected to be 41.8% in 2024 in the US (Statista, 2019). Moreover, ad-based services generate much lower revenues compared to subscription-based services. Therefore, as of today, the main challenge for streaming platforms is to convert free-users into paid-users because there are not many potential users left in developed countries. In the near term, there is still room for horizontal growth with other regions such as South America, Asia or the Middle East, even if it can be quite challenging. In these regions, progress has been made regarding free and legal music streaming services to fight against P2P downloads, but the main issue is to convert those new free users into paid subscribers. We are now going to focus on several countries/regions:

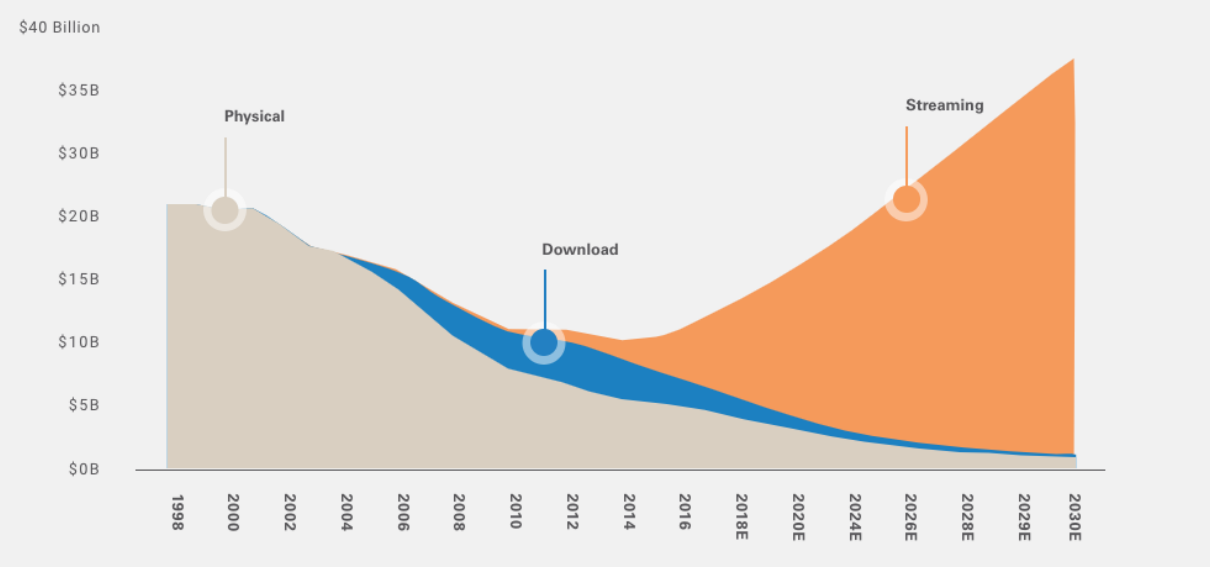
- **China**: the US, Europe and China represent approximately 74% of the global digital recorded music market but when comparing their average revenue per user (ARPU), the difference is striking. From 2019 to 2024, the ARPU in China is expected to increase from $3 to $4, while in the US and in Europe, it will increase from $33 to $37 and $21 to $24, respectively. It shows that Chinese users are not willing to pay for musical contents, which can be explained by different cultural habits or perceptions of the cultural product that music is. (Statista, 2019)

- **Japan**: it is one of the countries where the shift from physical channels towards digital did not happen yet. In Japan, people are still buying a lot of CDs to listen to music. In 2017, 80% of all recording revenues were generated from CD sales. (Soundcharts Blog, 2019).
- **India**: the opportunity is significant since only 10% of the population is using streaming platforms. However, like China, only 1% of users are paying for a subscription and it is challenging to convert free users into paid subscribers. (Soundcharts Blog, 2019).
- **South America**: driven by countries in love with music, like Brazil, the region has been growing fast in recent years. In 2018, revenues increased by 16.8% in the region (IFPI, 2019).

Even if, the streaming market is slowing down, many opportunities remain, mostly outside Western countries. In all those markets with a high potential, it has become even more necessary for main record companies – Universal Music, Sony Music and Warner Music – to sign the best artists and to develop local repertoires. Indeed, to prevent streaming platforms from entering into better agreements with local labels – reducing their costs linked to music distribution – Majors companies must get the exclusivity for local artists because streaming companies can only tackle those new markets by providing users with local catalogues. They represent tremendous opportunities for both record companies and streaming platforms and the end of the fight between both sides is not about to happen.

With those potential improvements, streaming might maintain a strong growth, driving the global recorded music market to high levels but this will be extremely challenging. On the other side, it has become even more significant to control the cost structure if revenues are about to reach a plateau.

Figure 2: Global recorded music industry revenues, 1998-2030 in $bn, (Goldman Sachs, 2018)



1.2. The role of record houses

In the previous part, we have seen that the era of digital implied two changes for record companies. First, music has become a mass consumption product and as a consequence, there are more and more people trying to break into the industry, which increases competition between artists and makes it harder for record companies to select the ones they will support. On the other side, the last decade has seen a growing discontent against main record companies, denouncing the way artists are being treated. With new technologies, they can now record their own songs at home with a personal computer, avoiding expensive studios, and they can even distribute music themselves using streaming services like Spotify or YouTube. It seems that artists can now afford to become independent from Majors.

However, we believe that, even if record companies are frowned upon by artists or streaming platforms, they still play a significant role in the industry, especially when it comes to breaking a new artist worldwide by building its career and developing a brand or other sources of revenues (live concerts, merchandising and sponsorship). In the last few years, they have even increased their investments in the industry.

In a record company, many teams work in partnership with the artist on different aspects of its career, leveraging the label network at every step of the artist's path, (IFPI, 2019):

- **Catalogue**: they are in charge of securing opportunities for a given song for a period of 1.5-2 years.
- **Non Record Income**: they take care of the sales strategy of an artist's merchandising products.
- **Data Insights**: they provide the artist with metrics to support a decision for its career.
- **Artists and Repertoire (A&R)**: this team is in charge of signing new artists on the label, advising them on business and creative perspectives to support their development.
- **Business Affairs & Legal**: they are in talks with all the partners of the artist for specific legal points.
- **Commercial Services**: they are in charge of sales for both digital and physical formats.
- **Marketing**: they organize online and offline campaigns – online, radio, billboards, TV ads, etc. – to promote the artist.
- **Sync & Brand**: they develop the partnerships between an artist and several brands that could help connecting with fans.
- **Video Production**: they are in charge of helping an artist for everything that is linked to video contents.
- **Creative Services**: the team develops the artist's visual identity to communicate.
- **Press & Publicity**: they are in charge of creating contents for the artist through the organization of media interviews, publications on social networks, etc.
- **Promotion**: the team connects fans from all around the world to an artist's music.

When it comes to investments, music labels are the main investors in the music industry. In 2018, according to (IFPI, 2019), approximately 33.8% of global revenues were invested back in A&R and marketing, representing an amount of $5.8bn – $4.1bn and $1.7bn respectively. This can be compared with the $4.5bn invested back in 2015 with $2.8bn in A&R and $1.7bn in marketing (IFPI, 2016). These investments are risky because it is hard to say if a totally new artist will break into a top chart, allowing the record company to recover its investment. On average, the costs associated to break a worldwide artist in a main market – like the UK or the US – are between $0.5 and $2m, which can be split into the followings (IFPI, 2016):

- **Advances** ($50k - $350k): to help the artist in his day-to-day life so that he can focus on music.
- **Recording costs** ($150k - $500k): these costs are variable according to the artists and the projects. For a top producer, it can be around $45k per song.

- **Video production** ($25k - $300k): these are linked to the shooting of a video clip.
- **Tour support** ($50k - $150k): organizing worldwide tours for the artists.
- **Marketing and promotion** ($200k - $700k): online and offline marketing to connect the artist with its fan base.

Then, we need to consider that the success ratio for a new artist is between 1:4 and 1:10 depending on the record companies. It means that most of the time, labels will have to count on the main success of a single artist to recover their investments in the ones that failed. Even for the Majors it is not that simple. For example, 1 500 marketing experts were involved in launching Justin Bieber worldwide (IFPI, 2016).

This ratio of success also indicates that there is still room for improvement in terms of investments following the selection process of a new artist, if the company has more information on whether or not the artist could break into the charts. Indeed, the selection process is performed manually, with A&R managers receiving music every day and having to choose. They use their past experience but most of all, recommendations from music journalists, radio broadcasters or other experts. We believe that technology could help in that process if it exists a recognizable pattern behind Hit songs that algorithms will be able to recognize, helping the A&R department. If so, costs could be drastically reduced.

Machine Learning algorithms are already applied in several research fields in the music sector, mostly for streaming platforms' recommendation algorithms, which is why we decided to research on the use of such algorithms to help A&R managers in their decision process.

1.3. The applications of Machine Learning in the music market

With the shift towards digital platforms, the actors of the music market have been able to leverage available user data to completely reshape the listening experience. Available data led to the development of recommendation algorithms and the emergence of a hyper-personalized music listening experience as well as its contextualisation. In today's society, the main challenge is to provide the users with the best songs according to their tastes, moods or even contextual information such as the weather.

The term Machine Learning was first coined by (Samuel, 1959) with his checkers-playing program that was one of the first self-learning programs developed at that time. It is a sub-field of Artificial Intelligence and it became widely popular a few years ago. It is now used in a wide range of areas – automobile, computer vision, fraud detection, financial markets, sentiment analysis, speech recognition, etc.

In the music sector, the research field of Music Information Retrieval (MIR) (Downie, 2003), that comprises the works related to the retrieve and analysis of information from large music datasets, gathers various subjects such as:

- **Genre classification**: musical genre attribution was done manually, and this work area focused, with success, on automating the process using audio features. In (Tzanetakis & Cook, 2002), one of the first papers on the subject, they used rhythmic variables, timbral variables and pitch information and they achieved a correct classification for 61% of the dataset, across 10 music genres. The genre classification area was one of the first in the MIR field.
- **Mood classification**: the idea here is to automatically classify songs into various mood categories based on audio analysis, with the addition of lyrics that can improve the overall results (Laurier, Grivolla, & Herrera, 2008). This research field is based on several works, including (Russell, 1980) which defined a 2-dimensional matrix to classify a whole list of emotions according to arousal and valence[4]. This was applied to music classification as the valence and arousal values for a song can be computed.
- **Recommendation systems**: these algorithms aim at predicting what preference a given user would assign to a particular song to then suggest music that the user is supposed to like. They became more and more predominant with the development of services like Netflix, Amazon and Spotify. These systems combine various methods and can become extraordinary competitive advantages for a company. Spotify’s recommendation algorithm is well-known for being one of the most advanced in the music industry. It combines collaborative filtering – recommendations based on similar users – content based – use of raw audio features to find similar songs – and natural language processing to screen blogs and articles on the web. (Ciocca, 2017).
- **Contextual listening**: the idea here is to provide the user with a song that is adapted to the context of the current situation. As an example, suppose that it is raining, a song

[4] Valence in music describes the musical positiveness of a given song

related to the rain would be recommended to the user. Several start-ups were created based on this field like Endel, a German-based company, which creates personalized environments to help a user to focus and relax.

Machine Learning’s applications are numerous in the music industry but consists mainly of automating certain processes (also to produce music), improving the general listening experience of the user to discover new songs or using music for certain contexts. We will now focus on the research field that analyses the use of Machine Learning to determine whether or not a song will be a Hit.

2. Literature review

In the first part, we described the evolution of the music market, its actors (in particular A&R professionals in charge of discovering new artists and helping them to refine their music) as well as other Machine Learning applications in the Music Information Retrieval field (Downie, 2003) such as genre classification, mood classification or recommendation algorithms (content-based filtering and collaborative filtering).

Inside this research field that is MIR, the study of what it takes for a song to be a Hit is often referred to as "Hit Song Science", which claims that "cultural items would have specific, technical features that make them preferred by a majority of people, explaining the non-uniform distribution of preferences" (Pachet & Roy, 2008). Many societal, cultural or other qualitative factors can influence the popularity of a song and this research field is exploring the use of quantitative data to explain it.

First, we will study the research papers that focused mainly on what we call intrinsic factors (title, duration or more acoustic features such as timbre). Then, we will list the papers that decided to add external factors to their models to take into account other dimensions of a song popularity: mainly lyrics, artist metadata, social factors and time.

2.1. The use of audio features

In this part, we will tackle the literature that focused on predicting song popularity using mainly audio features (but not only). Something interesting is that the variables used in the models evolved significantly over time. In the beginning, authors had to compute themselves the values taken by the variables they wanted to use in their models, deriving them from signal processing operations and unsupervised methods. Examples include the Mel-frequency cepstral coefficients (MFCCs) that represent a song timbre and the audio signal spectral centroid of a song.

In June 2005, the company The Echo Nest[5], a music intelligence data platform was founded and became a leader. The company used to work with many famous companies such

[5] The Echo Nest: http://the.echonest.com/company/

as Twitter, Nokia, SiriusXM, MTV, Yahoo and others providing developers with the design of complex audio features. In 2014, the company was bought out by Spotify. Researchers in the field of Hit Song Science were then able to use those new features to get better results with their models.

2.1.1. From simple audio features (…)

The first research on the subject was made by (Dhanaraj & Logan, 2005). They gathered a collection of 18 500 songs, dating from 1956 to 2004 and used unsupervised clustering methods to compute the MFCCs and other acoustic attributes of songs. They also extracted lyric features from the Astraweb Lyrics Search Engine2[6] website. Then, they applied Support Vector Machines algorithms (SVM) and Boosting Classifiers on a set composed of 1700 songs – including 91 hits defined as top 1 song in the US, the UK or Australia – that combined both lyric and acoustic features. Even with simple acoustic features, that are not described in the paper, they managed to get a better performance than a random classifier – AUC (Area Under the Curve)[7] of 0.69 when combining audio and lyric features – making Hit Song Science a reality. However, the size of the dataset balances the results.

As an answer to (Salganik, Dodds, & Watts, 2006), an experiment was made by (Pachet & Roy, 2008). They used 3 feature sets with 32 000 songs: a generic one composed of 49 spectral, temporal and harmonic features, a second one composed of 98 Blackbox features developed by Sony Music and a third one with human features (difficult to quantify or measure) gathering 632 Boolean values. They divided song popularity into 3 sub-categories to avoid a binary definition, which does not take into account the intermediary steps to become popular. They trained SVM algorithms on the 3 sets and compared their results to a random classifier. They concluded their experiment stating that some subjective labels might be learnt by Machine Learning algorithms, but Popularity cannot. However, they qualified their conclusion, saying that further research should focus on determining additional features, but that current acoustic classifiers were not able to model song popularity.

[6] http://lyricsearch.net/
[7] https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

We have just seen two different works both using SVM classifiers and acoustic features derived from traditional digital signal processing theory. They had different conclusions but even (Dhanaraj & Logan, 2005) had poorer results using only the acoustic features.

2.1.2. (…) To advanced audio feature engineering

As an answer to (Pachet & Roy, 2008), the work of (Ni, Santo-Rodríguez, Mcvicar, & De Bie, 2011) concluded positively to the question "*Hit Song Science Once Again a Science?"* They used the UK top 40 charts from the past 50 years and got a dataset of 5947 unique songs with features extracted from The Echo Nest's API: Tempo, Time Signature, Duration, Loudness, Loudness variance among audio segments and Harmonic Simplicity. They decided to distinguish the top 5 from the top 30-40 songs in order to get equally balanced classes and they trained a time-weighted shifting perceptron model to take into account evolving music tastes. As a result, the accuracy metric was between 56% and 62% – with balanced classes – leading them to conclude that with better audio features and a different definition of song popularity, Hit Song Science might be feasible. However, they noticed that performing a classification task with such a definition of Popularity was probably easier. Indeed, the songs composing the dataset are "similar" in a sense that they already belong to a chart, which reduces noise in the dataset.

As an attempt to further develop the audio features used in Hit Song Science, the work of (Herremans, Sörensen, & Martens, 2014) was to do advanced audio feature engineering to improve the results of the Machine Learning algorithms. They constructed a database composed of dance hit songs from 1985 to 2013 using The Billboard Top 40 charts and the Official Charts Company (OCC) to get 21 692 songs with 4 features: Song Title, Artist Name, Positions in the charts, Dates of the Positions, Peak Chart Position. Then, they used The Echo Nest's API to extract other audio features that can be split into 3 categories:

- Metadata, which are descriptive information, not related to the audio signal: Artist Location, Artist Familiarity, Artist Hotness, Song Hotness or Duration.
- Audio features: Tempo, Time Signature, Mode, Key, Loudness, Danceability and Energy.
- Temporal features, which take into account the variations within a same song: Timbre (12-dimension vector for each audio segment), Beatdiff (time difference between the elements of a series of beats).

They also decided to test 3 definitions of popularity:

- Comparison of the Top 10 versus the Top 30-40: biggest gap between Hits and Non-Hits.
- Comparison of the Top 10 versus the Top 20-40: smaller gap between Hits and Non-Hits.
- Split of the dataset into 2 at position 20: no gap between Hits and Non-Hits.

Then, they tried Decision Tree classifier, RIPPER Ruleset, Naïve Bayes, Logistic Regression and Support Vector Machines on those 3 datasets, keeping only the data from 2009 to 2015, to compare the results. They found out that several components of the timbre vectors have a good predictive power, meaning that going deeper into the definition of audio features (segmental analysis of an audio signal) might be promising. Moreover, the definition of song popularity with the largest gap gave the best results, proving that even the Top 10 has a pattern that makes it different from the bottom of the chart. Our understanding of this conclusion is that their method to define popular songs as a smaller share of a chart rather than having a binary approach (being in a chart / not being in a chart) increases the performance of the algorithms. In terms of models, Decision Tree and Ruleset did not perform well given their simplicity (but they are easy to interpret). SVM did not perform that well but Naïve Bayes and Logistic Regression gave the best results with both an AUC of 0.65 on the first dataset, using 10-fold cross validation method. Finally, compared to other past projects, they used a database composed of recent songs to train the algorithms and focused on one genre of music to reduce noise in the dataset because they observed a significant evolution of Hit song features over time, even for the same music genres. We can see here that with advanced audio feature engineering, better results can be achieved on a significant enough dataset, while there is still room for improvement adding Lyrics and Social information to the model, according to the authors.

In this research project too, (Garcia, Kala, & Barajas, 2017) tried to have a deeper approach on the audio features used, with different levels:

- Macro-level: Key, Tempo, Time Signature and Loudness.
- Micro-level with features for various segments of a song: Pitch and Timbre (compute also an average difference in Timbre values to understand how the Timbre evolves progressively).
- TF-IDF Bag-Of-Words: a 300-dimension vector is created using the most common words found in a subset.

- Creation of a dummy variable for the location of the artists.

They used the Million Song Dataset[8], which was created in 2011 using features provided by The Echo Nest's API, and they decided to define popularity using the Hotness value of a song. This is interesting because the variable is continuous, and the authors can now use a regression model to explain the popularity of a song. Compared to previous works, we do not have a binary definition and it now takes into account the progressive steps between the first and the last song of a chart. Moreover, regression models allow for a better interpretation of which features explain the best the popularity of a song. They also applied unsupervised algorithms such as K-Means clustering and Principal Component Analysis, which did not yield significant insights. A higher Hotness seems to be associated with louder songs and with short titles. However, the regression model gave really interesting results on a correlation between Key and a song popularity, explained by the fact that popular songs are written with a simple key value. In terms of lyrics, the following words had a positive impact on the Hotness value: Guitar, Acoustic, Hardcore, Instrumental, Soundtrack. In the end, once again, the best result was obtained combining all the features. The authors got an MSE (Mean Squared Error) of 2% for a Hotness value oscillating between 0 and 1, which is very good. This work proves again that a segmental analysis of audio features leads to better results.

In another work, the authors also used the Hotness value of a song to define the popularity of a music. But what is interesting in the work of (Pham, Kyauk, & Park, 2016) is that they defined a binary label Hits/Non-Hits, splitting their dataset according to the distribution of the Hotness value. The first 25% were labelled Hits and the rest non-Hits with a splitting value of 0.623 (Hotness value is between 0 and 1). They used the Million Song Dataset with audio features like: Duration or Key. And metadata (extracted previously from The Echo Nest's API): Danceability, Energy or Song Hotness. Also, they had array data containing segmental measures of the audio. In the end, the dataset was reduced from 10 000 songs to 2 717 songs – deleting tracks with missing values – and from 977 features to 45 – reducing array variables using mean and variance values in addition to Forward Stepwise Selection, Backward Stepwise Selection. They used several classifiers – Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, SVMs and a Multilayer Perceptron – and a linear regression in the end. They concluded that with a classification approach, information was lost due to the definition of the popularity of a song. Indeed, it is hard to say

[8] http://millionsongdataset.com/

that a song is famous or not because there are different popularity degrees. Moreover, metadata (labels, artist familiarity), for both classifiers and regression models had more predictive power than acoustic features. This makes us believe that external factors seem to bear more importance than acoustic features in explaining popularity.

Finally, another research work concluded that audio features are not sufficient to explain the popularity of a song. (Borg & George, 2011) decided to define song popularity using YouTube's video counts – for the most viewed video corresponding to a specific music. They also used the Million Song Dataset with the typical audio features that we already talked about: Song Title, Artist, Duration, MFCCs (for Timbre), Loudness, Tempo, Time Signature, Key or Mode. They averaged the MFCCs and computed the variance of these vectors. First, they proceeded in a correlation analysis between the audio features and their popularity variable. It appeared that most metadata feature, even the song Hotness value, were not good predictors of YouTube's video counts – i.e. weak correlation. Once again, the conclusion was that audio features do not have sufficient predictive power to explain a song popularity. However, in this work, we believe that the weak results were due to the dataset composed of many different genres of music as well as this definition of song popularity. Indeed, we think that many other factors explain video counts such as the video clip, a same person watching the video more than once or social factors (friends, etc.). It means that we need to pay attention to the features we choose and the definition of Popularity.

Here, we have seen that using advanced audio feature and audio segmentation may produce better results, mostly because with the apparition of The Echo Nest, researchers have been able to easily integrate complicated audio features in their models compared to what was used at the beginning of Hit Song Science. However, many works concluded that audio features were not sufficient to explain song popularity and that external factors should be added to improve the results such as indicators of the artist popularity, lyrics, etc. Moreover, we found out that the definition of popularity must be taken seriously. It seems that formulating the problem of Hit Song Science as differentiating the top of a chart from the bottom or using song Hotness as a proxy (and continuous variable) produces good results.

2.1.3. And better performing algorithms

In this part, we are presenting the work of (Yang, Chou, Liu, Yang, & Chen, 2017), who applied Convolutional Neural Networks to improve the results compared to typical Machine Learning algorithms. They used listening data from KKBox, a music streaming application for Asian countries, for 30 000 users and 125 000 songs. In this work, popularity was defined as the play counts of a given song and two subsets were created: Western songs and Mandarin songs. Then, different Convolutional Neural Networks are trained on the two datasets. It resulted that deeper and more complicated networks returned better results than traditional Machine Learning algorithms for the Hit Song Science problem. Also, an interesting point is that the authors got better results with the Mandarin subset, indicating that cultural origin of songs can impact song popularity. They explained that by the higher diversity of genres in Western countries, which is why deeper structures performed better because they can capture diversity in acoustic features.

Once again, popularity is affected by external factors like cultural origins. Deeper models appear to perform better on noisy datasets compared to traditional methods.

2.2. Adding external factors to improve the model

In the previous part, we have seen that advanced audio features now computed and provided by The Echo Nest allowed models to perform better at explaining song popularity. However, many research works indicated that adding external factors – that are not derived from the audio signal – would definitely improve the results.

2.2.1. Lyrics information

Even the oldest work in the field of Hit Song Science, (Dhanaraj & Logan, 2005), explored the addition of Lyrics information in their models – using the Bag-Of-Words method – and they obtained better results when combining both acoustic and lyrics features. But most importantly, they stated that lyrics features were even more important than acoustic ones to identify popular songs. This must be taken into account event if the dataset used in that experiment was small.

In another experiment, (Isogawa, Masling, & Pan, 2019) tried to understand the main factors of popularity for a given song among its audio attributes, the lyrics and the artist in itself. They used songs from the Billboard Hot 100 chart as Hits and a Kaggle dataset mostly composed of songs that did not make it to the Top 100 as Non-Hits. The combined dataset was equally balanced between each class and audio features were extracted from The Echo Nest's API. In this work, the authors used the Bag-Of-Words method to integrate Lyrics components in their models. They then trained a Binary Linear Classifier, Logistic Regression and SVMs and got the following results: with lyrics features only, Logistic Regression got an accuracy of 87% and a F1 score of 0.89, and combined with audio features, the Binary Classifier yielded an accuracy of 89%. Therefore, the addition of audio features only improved by 2% the best result achieved with lyrics features only meaning that they seem to bear more importance than audio features to explain popularity.

2.2.2. Artist information

To assess the impact of P2P networks on an album survival time in a chart, (Bhattacharjee, Gopal, Lertwachara, Marsden, & Telang, 2007) compared survival time in the post and pre era of P2P. They found out that those networks impacted negatively albums that debuted at a low rank (less popular) mainly due to 3 factors:

- The Superstar effect: coined by (Adler, 1985), artists that are already popular tend to remain popular because consumers desire to minimize the time they spend looking for information on a new artist. They have to weigh the pros and cons in terms of costs between looking for new songs or staying in their comfort zone.
- Popular/Unpopular labels: small labels offering less popular catalogues are more impacted by P2P networks because they can be the victims of negative word-of-mouse very quickly.
- The initial position in the chart: this is the result of the experiment in this research. Songs released by labels with higher marketing spending prior to the release date tend to perform better because they will benefit from the positive aura of the label and start higher in a particular chart.

In the light of this work, it seems that song popularity is impacted by complex factors, including the popularity of an artist. One method to assess this factor is to create a dummy variable indicating whether or not an artist is already popular. Indeed, the Superstar effect indicates that well-known artists are much more listened to by consumers than unknown ones.

In the context of P2P technologies, (Hann, Oh, & James, 2011) analysed how networks of people could be used by music companies to assess the future popularity of a given song. They used various variables for album characteristics such as: Genre, Number of Daily Comments on YouTube (before the release of the song – marketing proxy), Artist Gender, Total Number of Past Albums. And they extracted sales data from the Billboard Top 200 albums as well as downloading data from the Ares P2P network. It appeared that the features had a significant predictive power for the sales of an album with an adjusted R_2 increasing from 56% a month before the release of the album to 73% a week before. We can conclude that external artist information should be considered in a model.

More recently, the work of (Georgieva, Suta, & Burton, 2018) intended to predict the ability of a song to reach out the Billboard Hot 100 based on Spotify data. The authors used songs from 1990 to 2018 to get a balanced dataset (Hits and Non-Hits) composed of 4 000 songs. In addition to the typical features extracted from The Echo Nest (bought out by Spotify in 2014), they created a dummy variable – Artist Score – indicating if an artist already had a song part of the Billboard Hot 100. They got a 75% accuracy with Logistic Regression and an Artificial Neural Network with one hidden layer. Also, the Artist Score was considered to be the most important feature in the models in addition to other acoustic features: Instrumentalness, Danceability, Acousticness, Speechiness, Valence and Energy.

In this part, we have seen that artist information such as popularity, gender or label could significantly improve the results combined with audio features.

2.2.3. Social factors

P2P networks are a good proxy to assess the social outreach of a song. Indeed, at that time, digital platforms were not developed as much as today, and most sales were still physical. Those platforms were a good way to obtain information about a song interest (counting queries). The work of (Koenigstein, Shavitt, & Zilberman, 2009) analysed the relationships between a song popularity on a P2P network and its rank in the Billboard chart. They found a strong correlation between the two, proving that social factors could impact popularity. They managed to predict the appearance of a song in the Billboard Hot 100 based on P2P queries before release with more than 86% of precision. As P2P information are sometime available before the official release date of an album, labels could use such information to take advantage of those platforms

– the authors concluded that the discovery of a Hit song could be done 2 or 3 weeks before the release of the track based on P2P queries.

Another interesting social factor, that we already talked about, is the geographic origin of Hit songs. The work of (Fan & Casey, 2013) compared 40 weeks of data for UK's and China's Pop songs Top 40 with features that were extracted from The Echo Nest's API: Danceability, Duration, Energy, Key, Time Signature, Loudness, Mode or Tempo. They defined popularity with an equal chance of being a Hit or a Non-Hit – the 20 first songs are Hits and the others are Non-Hits. Then, they used a Time-Weighted Linear Regression – more weight for recent data – with the benefit of coefficient analysis, as well as SVMs, which are not great to provide information regarding the features used. The linear model performed better on Chinese songs compared to UK's, which is explained by a less diversified basket of genres for China. The SVMs models performed well on both datasets. This study helped define a difference between Hit features in the UK and in China. It seemed that: Liveness, Speechiness, Mode, Time Signature, Energy and Danceability were the most important ones in the UK (ordered), while for China, it was: Speechiness, Liveness, Danceability and Energy. Therefore, it appears that cultural differences remain between Hit songs from various countries, indicating that song popularity is not equally defined in every geographical area. This difference between cultures was also analysed by (Yang, Chou, Liu, Yang, & Chen, 2017) who compared a Western Dataset and a Mandarin's. They concluded that Western music had more diverse genres and that the predominance of Pop music in China explained why Hit song prediction was easier.

Here, (Montecchio, Roy, & Pachet, 2019) tried to understand if the structure of a song could influence what they called the « Skipping Behaviour » of users, which is defined by the action of skipping to another song while listening to music on a digital platform. Indeed, with streaming platforms, consumption evolved significantly, and music tended to become a mass-consumption product. They analysed a set of 100 popular songs with 12 in the top 100 and 40 in the top 1000 with a high diversity in terms of genres and artists. They found out that users were more sensitive to particular audio structures, which could be used to produce music that would catch the attention with a higher chance of becoming popular. However, they concluded that Hit song prediction was rather unfeasible because listeners' subjective preferences were not uniform over time, due to social interactions. It means that a Hit can be a Non-Hit for the same user. This is why other researchers in the Hit Song Science field decided to integrate a Temporal dimension in their works – also because features evolved over time.

We have seen here that social factors such as culture, people networks and individual interactions are some of the main factors that explain popularity. There are many others that could be integrated but the idea here was to show that social factors also played a significant role to explain popularity.

2.2.4. Temporal dimension

We have seen before that Hit song features have been evolving over time mostly because music is a fashion product, as explained by (Salganik, Dodds, & Watts, 2006), and users' tastes are changing. From our analysis of previous works, it seems that two methods can be applied: considering only recent songs or adding a temporal dimension to the model, taking into account those evolving trends.

The work of (Chon, Berger, & Slaney, 2006) was one of the first which analysed the impact of an album starting position on its chart trajectory. They used 3 years of published bi-weekly sales from the Billboard for Jazz music and they discovered that most albums started at their peak position or at least close to that peak to then go down the chart. It means that there is a high correlation between the starting position of an album and the time it remains in the chart. Popularity evolves over time and a temporal dimension must be taken into account.

In addition to the evolution of Hit song features, (Askin & Mauskapf, 2017) showed that time dimension matters in the explanation of what makes music popular because they found out that tracks sounding too much like previous hits were less likely to succeed, meaning that the distant past cannot be used and that artists cannot just copy patterns of Hit songs. In their experiment, they used the Billboard Hot 100 from 1958 to 2016 and included the following audio features: Key, Mode, Tempo, Acousticness, Energy and Danceability. In addition to feature temporal dimension, they also found out that features mattered for cultural products, meaning that audio features play a role in the explanation of popularity: Danceability, Liveness, 4/4 Time Signature were associated to Hit song features. Therefore, a Hit must be a combination of features that follow a frame defined by past Hits but also novelty to differentiate itself. According to that study, a Hit pattern does not exist or at least evolves over time. We can define borders for some audio features but what makes a song popular is its "optimal differentiation".

To take into consideration this temporal dimension statistically, the work of (Perrie, 2019) focused on creating a target variable to model song popularity as well as its evolution over time (how many listeners and how long it would stay in the charts), comparing trajectories of Hit songs. He used the Billboard Hot 100 for its target variable as well as Spotify's API to extract audio features: Danceability, Energy, Acousticness, Instrumentalness, Liveness, Speechiness, Tempo, Valence, Duration, Key, Mode, Time Signature and Loudness. Better results were obtained with binary classification compared to multiclassification. The author explained that most previous works in the field of Hit Song Science had trouble explaining popularity only with audio features because song performance has multiple dimensions that need to be taken into account. This work is interesting because instead of stating that more variables should be added to improve the results, the author explained that the definition of popularity could not be taken as a 1-dimension feature because it could be measured in different ways: survival time in a top chart and the position in the chart.

Given the fact that Hit song features have been evolving and that consumer's behaviours can change quickly, adding a temporal dimension to a model may improve the predictions. But what we need to remember from those studies is that the definition of popularity is complex.

2.2.5. Marketing spending

We have seen in previous studies that the starting position in a chart is highly correlated to its trajectory. Therefore, labels should pay attention to their marketing strategy and invest significantly if they believe that an artist has the potential to reach the top. This is what explained (Silber, 2019) in her thesis. She compared recommendation algorithms from Spotify and Soundcloud and explained that an important factor impacting the recommendation of a song was the label paying the streaming platform to promote it. It implies that a song can become a Hit thanks to other factors than its audio characteristics.

This is exactly the dimension of popularity that (Borg & George, 2011) tried to model with YouTube video counts. Indeed, the video clip is the final product that is delivered by the label to promote the track and reflects marketing spending.

Obviously, marketing spending has a significant influence on the ability of a song to become a Hit. In the era of digital streaming platforms, it is even more important as it has become easier for distribution channels to target potential consumers of a given song and to offer that service to labels and music companies. Therefore, we believe that integrating a marketing proxy in a model may improve the results.

2.3. Conclusions

In this literature review, we have seen that most of the research on the subject of Hit Song Science focused obviously on: defining song popularity, which variables to integrate and which algorithms to train.

Popularity has been defined in 3 different ways:

- **Binary**: in this scheme, a song is considered to be popular or unpopular and this can be modelled differently. Some authors translated that into the ability to be part of a chart (the Billboard mostly) or not. Others divided a chart to define classes: splitting it in the middle or with a gap between the two classes (Top 10 versus Last 10).
- **Continuous**: use a continuous/discrete variable as an indicator of popularity such as song Hotness (The Echo Nest), video counts, number of comments or rank in a chart.
- **Multiclass**: this case is similar to the first one, but the authors considered a split into 3 or more classes, following the same methods.

Variables can be classified into two categories:

- **Audio features**: the first works in the field of Hit Song Science started by using signal processing methods to define audio features – like MFCCs – to then adopt the advanced features provided by The Echo Nest's API. Recent works focused also on exploring segmented audio signals to analyse the variation of audio features in a same song.
- **Metadata**: as song popularity cannot only be explained by audio features only, the authors started integrating other variables to take into account the multi dimensionality of popularity. The main factors studied were: lyrics – with a great predictive power – label and artist metadata – the popularity of an artist is highly correlated with the song popularity – cultural aspects of geographical regions, marketing spending, people networks or temporal dimension – with the evolution of tastes and therefore Hit features over time.

When it comes to algorithms, the typical supervised Machine Learning algorithms were used in most cases. Good results were obtained with Support Vector Machines and Logistic Regression for classification tasks and Linear Regression performed well for a continuous/discrete popularity variable. Several works implemented Artificial Neural Networks and got better results than traditional algorithms with deeper structure, being able to take into consideration more diversity in the datasets. Some unsupervised algorithms – K-Means clustering and Principal Components Analysis – were used to determine general patterns in the data without tremendous success.

Another point that must be noted, is that even though the research started at the beginning of the 21st century, authors are still differing on the results they obtain. There are still no clear answers to the question: Is Hit Song Science a science? (Pachet & Roy, 2008), which is why we would like to further dive into the subject.

From the past researches we define the following research questions:

- Is Hit Song Science feasible?
- Which are the most efficient Machine Learning algorithms?
- How could we define song popularity?
- Which features have the best predictive power? Audio features? Lyrics? Artist information? Label information? Social factors?

3. Methodology

Now that we have an idea of how the Hit Song Science research question has been tackled in previous works, we will define here the methods we used for our experimentation. Mainly how we built our database, gathering new features to try and how we conducted the process of training and testing Machine Learning algorithms.

3.1. Database and features

3.1.1. Sources

Going through past works, we noticed that the main data sources used were the Billboard Magazine's API (Askin & Mauskapf, 2017), the Million Song Dataset (Pham, Kyauk, & Park, 2016), the UK Top 40 (Ni, Santo-Rodríguez, Mcvicar, & De Bie, 2011), The Echo Nest's API (before being bought by Spotify) and later on, Spotify's API[9] (Georgieva, Suta, & Burton, 2018).

We decided to try a different approach using Spotify's Top 2018 and 2019 playlists as the main sources for our songs (compared to the Billboard charts for instance) for two reasons. First, we wanted to tackle the Hit Song Science topic using only well-known songs (Hits) as several previous works showed that for an experiment, it was better to reduce noise in the dataset by selecting songs which made it to a chart (UK Top 40, Billboard, etc.). In (Herremans, Sörensen, & Martens, 2014), the authors defined as Hits the top 10% of UK Top 40 charts and as Non-Hits, the bottom 10%. The reason for such an approach is that the papers which used true Hits (chart lists) and then random songs as Non-Hits were therefore considering that the song number 101 on a top 100 was not a Hit, as much as another one that would have been far from reaching out to such a position. Second, Spotify bought The Echo Nest in 2014 and made available most of the features that were developed by The Echo Nest and used in past researches. Although we could have used the Million Song Dataset, we chose Spotify playlists to get recent Hits as (Herremans, Sörensen, & Martens, 2014) stated that the main features defining popular songs were evolving over time.

[9] Spotify's API: https://developer.spotify.com/documentation/web-api/

We used Spotipy[10], a Python library to work with Spotify's API, to collect a list of 150 songs (50 from the Top 2019 and 100 from the Top 2020) with 27 features. Then, we removed duplicates (songs being twice in a Spotify's Top), checking if they were true duplicates, to end up with 144 songs. Also, we would like to underline that we extracted metadata for each artist involved in a given song, keeping information into list objects and aggregating it into features that we will describe later on.

Then we got other features from various sources. For the lyrics, we used the Genius' API[11] to obtain an URL for each song and extract them from the web page. Indeed, lyrics are unfortunately unavailable directly from the Genius's API. We also wanted to monitor web activity for an artist name, prior to the release date of an album or a single. To do so, we used pytrends[12], an unofficial API for Google Trends, and got values for each artist over a 3-month period, ending 1 week after the release date of the album. Spotify already provides information regarding the artists but no basic features like age or country. Therefore, we gathered data from the MusicBrain's API working with the musicbrainzngs Python library[13]. Finally, we used LastFM's API[14] to get top tags for each song and use them as genres. Indeed, Spotify has a genre by track feature, but it was always missing for our dataset. In the end, our dataset was composed of 144 songs and 40 features.

3.1.2. Variables

We obtained those 40 features from the various sources presented above and they can be divided into 3 sets: song metadata, artist metadata and audio features, as well as 3 types: integers, floats and strings. For Spotify's features, the description can be found on the website Spotify for Developers. In our work, we decided to use general audio features provided by Spotify (there are more complex audio features that can be accessed using the API and that were used in several works of our literature review) and to test the addition of metadata obtained through other APIs. Among those features, some were recommended by previous studies, including Lyrics by (Pham, Kyauk, & Park, 2016) and (Isogawa, Masling, & Pan, 2019), Labels by (Bhattacharjee, Gopal, Lertwachara, Marsden, & Telang, 2007), Genres by (Reiman &

[10] Spotipy: https://spotipy.readthedocs.io/en/2.12.0/
[11] Genius: https://docs.genius.com/#/getting-started-h1
[12] Pytrends: https://pypi.org/project/pytrends/#description
[13] MusicBrain: https://python-musicbrainzngs.readthedocs.io/en/v0.7.1/
[14] LastFM: https://www.last.fm/api/

Örnell, 2018) or Artist Popularity by (Hann, Oh, & James, 2011). Others are completely new like our Google Trends features.

Table 1: Variables

Name	Source	Type	Method	Description
Target Variables				
Song Popularity	Spotify	Integer - Quantitative	Directly available.	A value between 0 and 100 mostly based on the number of plays the track received and if they are recent or not.
Hit / Non-Hit	Spotify	Integer - Quantitative	Differentiating the top 10% from the rest using the playlist ranking.	A value of 0 or 1 indicating the position of a track in the Top 2018 and 2019. It equals 1 if the song was among the top 10% in terms of ranking in the playlist.
Popularity Level	Spotify	Integer - Quantitative	Differentiating the top 20.8% from the rest using Song Popularity values (i.e. values > quartile 3).	A value of 0 or 1 indicating if a track was part (strictly) of the third quartile of the Song Popularity distribution.
Dropped Variables				
Playlist	Spotify	String	Directly available.	The name of the playlist a track belongs to (Top 2019 or Top 2018).
Track ID	Spotify	String	Directly available.	The Spotify's ID for a given track.
Album	Spotify	String	Directly available.	The name of the album.
Name	Spotify	String	Directly available.	The name of the song.
Artist	Spotify	String - List	Directly available.	A list of the names of the artists.
Markets	Spotify	String - List	Directly available.	A list of countries where the song is available (alpha-2 codes).
Year	Spotify	Integer – Qualitative	Extract the year from the release date feature provided by Spotify.	A value indicating the release year of the track. We dropped the feature because we did not have enough historical data (by choice).
Time Signature	Spotify	Integer – Qualitative	Directly available.	An indicator of how many beats are in each bar (4 time signature for most of our songs).
Song Metadata				
Duration	Spotify	Integer - Quantitative	Directly available.	The duration of a song in milliseconds.
Name Length	Spotify	Integer – Quantitative	Cleaning the name of a song (removing useless part like: "feat…" or "remixed by…").	The number of characters in the name of a given track.
Explicit Lyrics	Spotify	Boolean - Qualitative	Directly available. (convert Boolean values into 0 and 1).	Binary variable indicating if a track has explicit lyrics.
Lyrics	Genius	String - Qualitative	Extracting lyrics from the Genius's webpage of a song. Cleaning the string of the following string patterns: "\n",	The lyrics of the song. We transformed them using the bag-of-words approach (see below).

			“[…]” and “(…)”. We also excluded punctuations.	
Album Type	Spotify	String - Qualitative	Directly available.	Binary variable indicating if the song was released on an album or a single.
Record Company	Spotify	String - Qualitative	We had 81 unique values that we reduced to 9 (Universal (61), Warner (30), Sony (19), Independent (18), Independent & Warner (4), Independent & Sony (4), Universal & Warner (4), Independent & Universal (3), Universal & Sony (1). To do so, we identified the labels belonging to a Major and classify those that are not financed by one of the three as independent (including labels financed by conglomerates).	The record company behind a given track.
Genres	LastFM	String - Qualitative	Extract top tags and map unique values to a list of 9 different genres: r&b, dance, hip-hop/rap, latin/reggaeton, rock, pop, country, indie and soundtrack.	The genre of a track.
Month	Spotify	Integer – Qualitative	Extract the month from the release date feature provided by Spotify.	A value from 1 to 12 indicating the release month of the track.
Holiday Effect	Spotify	Integer – Qualitative	Assign a value of 1 if the release month is 5, 6, 7 or 8.	A binary value indicating whether the song was released during the summer period.
Artist Metadata				
Artist Popularity Mean	Spotify	Float – Quantitative	Mean value of the Artist Popularity score for each artist involved in a same song.	The score is computed by Spotify as a function of the popularity value of all the songs of an artist. We take the average value of the artists involved in a same song.
Artist Popularity Max	Spotify	Integer – Quantitative	Max value of the Artist Popularity score for each artist involved in a same song.	The score is computed by Spotify as a function of the popularity value of all the songs of an artist. We then take the maximum value. Compared to the mean, the idea is to take into account the combination of a very famous artist and a completely unknown artist who would be driven by the popularity of the famous artist.
Followers Mean	Spotify	Float – Quantitative	Mean value of the number of followers for each artist involved in a same song.	The average of the number of followers the artists of a same song have on Spotify.
Followers Max	Spotify	Integer – Quantitative	Max value of the number of followers for each artist involved in a same song.	The maximum number of followers the artists of a same song have on Spotify. The idea for using the maximum is the same as above.
GT Mean Value	Google	Float – Quantitative	Mean of the mean values for the Google trends of each artist over a 3-month period, ending one week after the release date of an album.	Google Trends values are computed for a defined time period with a maximum value of 100 – being the peak of

				interest for the key word, over the period. Low interests for a key word imply that the other values are low (one 100 and many low values). The mean is an indicator of whether or not the trends belong to the upper range.
GT Std Value	Google	Float – Quantitative	Mean of the standard deviation values for the Google trends of each artist over a 3-month period, ending one week after the release date of an album.	The standard deviation is an indicator of the volatility of trend values over the period. It allows us to understand if the values are fluctuating a lot around the mean or tend to be regular over the period.
GT Range Value	Google	Float – Quantitative	Mean of the range values for each artist. A range is the difference between the maximum value (100) and the minimum.	The range value is an indicator of the dispersal of trend values over the period.
GT Peak	Google	Integer – Qualitative	Assigning a value indicating the position of the peak (trends = 100) with respect to the release date. 0 for before, 1 for at (plus-minus 1 day) and 2 for after.	A list of integer values indicating for each artist of a given song if the Google Trends peak was before, at or after the release date of the album. It is an indicator of whether the artist already had a peak of interest before the official release date.
Age	MusicBrain	Integer – Quantitative	Transform the birthdate into an age value. For Groups, look manually for the birthdate of each member. Average the values.	Mean of each artist's age, involved in a same song.
Solo / Group	MusicBrain	String – Qualitative	Mix gender and Person Group information to create a feature taking the values: solo_male, solo_female, solo_other, group_male, group_female, group_mixed.	Categorical feature indicating if the song was performed by a group or a person, as well as the gender. Mixed stands for a group where we have different genders and other for people who do not identify themselves as male or female. Also, we considered to be a group a collaboration of several artists.
Country Code	MusicBrain	String - Qualitative	Extract the country code for each artist of a given song and same order, reduce each value to a string of unique country codes	Country codes (alpha-2 format) indicating the artist origin.
Audio Features				
Acousticness	Spotify	Float – Quantitative	Directly available.	Value between 0.0 and 1.0, indicating the confidence of the track being acoustic or not (1.0 means highly confident).
Danceability	Spotify	Float – Quantitative	Directly available.	Value between 0.0 and 1.0, indicating if a track is adapted to dancing (based on other elements such as tempo, rhythm stability or beat strength). 1.0 means very danceable.

Energy	Spotify	Float – Quantitative	Directly available.	Value between 0.0 and 1.0, representing the intensity and activity of a given song. A fast, loud and noisy song has a value close to 1.0.
Instrumentalness	Spotify	Float – Quantitative	Directly available.	Value between 0.0 and 1.0, indicating the total absence of vocals. Above 0.5, it is considered to be an instrumental track and a value of 1.0 means the track contains no vocals.
Liveness	Spotify	Float – Quantitative	Directly available.	Value between 0.0 and 1.0, detecting the presence of an audience in a track. Above 0.8, there is a great chance that the track was performed live.
Loudness	Spotify	Float – Quantitative	Directly available.	Values between -60 and 0 db. It is an average of loudness values across the entire song. It is highly correlated with the physical strength of a song.
Speechiness	Spotify	Float – Quantitative	Directly available.	Values between 0.0 and 1.0, indicating the presence of spoken words. 1.0 indicates speech recording (talk show, audio book or poetry for instance).
Valence	Spotify	Float – Quantitative	Directly available.	Values between 0.0 and 1.0, indicating the musical positiveness of a song. A value close to 1.0 refers to happy and cheerful tracks.
Tempo	Spotify	Float – Quantitative	Directly available.	Overall estimated tempo in beats per minute (BPM).
Key	Spotify	Integer – Qualitative	Directly available.	The key associated to a track mapped to integers between 0 and 11 using the Standard Pitch Class notation.
Mode	Spotify	Integer – Qualitative	Directly available.	A binary variable indicating the modality of a song. 0 corresponds to minor and 1 to major.

3.1.3. Defining popularity

The definition of song popularity is one of the main topics in the Hit Song Science research field. It is hard to define a method to measure such a subjective thing. In our literature review, we found out that it could be quantified using the following methods:

- **Binary variable**: there are several possibilities for this definition. A song is considered to be popular (=1) if it was part of a chart (like the Billboard or UK Top charts) and unpopular otherwise. For others, the Hit class is defined as the top 10% of a chart and the problem of Hit Song Science is tackled by differentiating the top of the chart from the bottom. (Ni, Santo-Rodríguez, Mcvicar, & De Bie, 2011) and (Herremans, Sörensen, & Martens, 2014) also used a gap between classes.

- **Multiclassification**: several classes define various degrees of popularity. This method introduces a smoother definition compared to a binary variable, as recommended by (Askin & Mauskapf, 2017) and experimented in (Pachet & Roy, 2008).
- **Continuous variable**: popularity is quantified using a continuous value like the number of YouTube counts (Borg & George, 2011) or the song hotness variable provided by The Echo Nest (Garcia, Kala, & Barajas, 2017) (similar to the popularity feature now available on Spotify's API).

Also, some of the previous definitions do not take into account the time dimension of popularity as evidenced by (Perrie, 2019). Indeed, a track can become more or less popular over time and old popular songs can suddenly come back onto the charts (this is the example of Queen - Bohemian Rhapsody in our dataset).

We chose to test continuous and binary definitions of popularity because we did not identify a single solution better than the others in our literature review. We used Spotify's track popularity as a continuous value. Also, we defined as Hit (=1) the top 10% of each Top playlist (5 songs for 2019 and 10 songs for 2018). However, we created and used a second binary variable for classification because we found out during our experiment that with the previous one, a traditional and simple algorithm like Logistic Regression was not able to classify a single Hit. However, we did not have time to test other algorithms on this popularity feature. We defined as popular all the tracks belonging strictly to the 3rd quartile of Spotify's popularity feature (top 20.83% of our dataset), a definition similar to the one used by (Pham, Kyauk, & Park, 2016).

3.2. Data analysis and pre-processing

3.2.1. Data analysis

Before diving into the use of Machine Learning models, we performed an analysis of our data to get our first insights: Who are the most successful artists in the dataset? What are the main genres? Which countries are the most represented? What are the most famous words in the lyrics? Also, we focused on univariate and bivariate analyses to visualize the distribution of our features as well as the correlations between them but most importantly with the target variables. Unfortunately, we did not have time to use our third definition of popularity (the one we used in the models) because it was created at a later stage of the project.

For the univariate analysis, we plotted the distribution or count of all our features and we used basic descriptive statistics to better understand the variables. Then, in the bivariate analysis, we plotted a correlation matrix for the quantitative features and a focus on correlations with Song Popularity, with and without threshold to remove outliers. We also performed Pearson's and Spearman's correlation coefficient statistical tests between our continuous target variable and the other quantitative features. We also tried to perform Chi-Squared tests for our qualitative features (using the binary definition of popularity) but according to scipy, our samples were not representative (there is still a discussion regarding the minimum number of observations per class). Finally, we visualized the links between our features with regression plots, kernel density estimation plots, distribution or count plots with a hue parameter.

In the end, we used Principal Component Analysis, excluding lyrics, to visualize our data using 2 components in addition to a colour parameter and see if the two classes (3rd target variable) were separable. We did so for the audio feature subset, the artist metadata subset, the song metadata subset and the whole dataset. For each Principal Component Analysis, we extracted the main coefficients in absolute value (in the eigen vectors) and the variance explained by the components (eigen values). Note that this step of the analysis was performed after pre-processing data and converting categorical features into dummy variables.

3.2.2. Data pre-processing

To use Machine Learning models for our project, we had to clean data, fill in missing values, pre-process and perform some feature engineering (lyrics for instance).

For missing values, most of the work was done manually. Every feature extracted from Spotify's API did not have any missing values except for the track genre, which is why we ended up using LastFM's API to approximate genres. While extracting lyrics from Genius website we were missing 15 values, mostly due to a mismatch between our song and artist names with the Genius' catalogue. Therefore, we looked manually for the missing values directly on the website to copy and assign them. Regarding the Google Trends, 3 artists were missing: Queen (because the true release date of the album was too old), Panic! At the Disco (too complex or long name) and Greatest Showman Ensemble (group name for a soundtrack in a musical play – the interests for the song were mostly driven by the main singer or the movie name). For Queen, we changed the date and used the release date of the movie (2018-10-23), for Panic! At the Disco, we looked manually on Google Trends website trying what was proposed by the search bar and for the last one, we kept the values found for the main singer of the song (Keala Settle). Regarding the features extracted using MusicBrain's API, 71 values (6 features with several artists for each of the 144 songs) were missing. It was often caused by groups which therefore had no gender information. We manually looked for the missing pieces of information on the internet and we also corrected the birthdate values for groups. Indeed, they matched with the inception date of the groups so they could not be converted into a true age value. We looked for each member's birthdate on the internet. In the end, we were also missing 5 values for LastFM's top tags, which we filled in manually with genre values.

Cleaning data was not a main part of this project because there was not that much to do. We cleaned lyrics as explained in table 1: we replaced "\n" (new line) characters by spaces and then we used regular expressions to remove everything that was between parenthesis or brackets ((…) and […]). Also, during the engineering of the Name Length feature, we cleaned song names to avoid counting useless characters like (feat. … or remixed by …).

Finally, we had to pre-process data to use our features with Machine Learning algorithms. We dropped the useless features: Name, Playlist, Track ID, Album, Markets, Year (popularity is evolving over time), Artist and Time Signature (probably a useful variable but our dataset is almost composed only of tracks with a time signature of 4). Then, we converted categorical features (Album Type, Record Company, Genres, Solo Group, Peak and Country code) into dummy variables. For the last two features, we had multiple values per song (list of strings or integers). Therefore, we used Scikit-Learn's MultiLabelBinarizer[15] which can handle such cases. For instance, a song with the value "US, CA" (country codes) will be converted into a vector of 0 and 1 taking the values 1 for the US and CA columns and 0 for the others. Finally, we also dropped some of the features that were highly correlated due to their

[15] Scikit-Learn: https://scikit-learn.org/stable/

construction: Artist Popularity Max (we kept Artist Popularity Mean), Followers Max (same) and GT Mean Value. We decided to delete them based on our data analysis.

Most of the work was done while dealing with lyrics. First, we used Googletrans library[16] to translate all of our lyrics to English and avoid counting twice a same word (same meaning). Then we used the NLTK library[17] to remove stop words and to perform stemming using SnowballStemmer. Stemming is a process that reduces words to their root form so that similar words are not counted several times. By doing so, we ended up with a vocabulary of 1424 words. Then, we used the bag-of-words method[18] to convert each song lyrics into a 1424-dimensional vector of integers (count for each word of the vocabulary in the given lyrics). Finally, we used TF-IDF method (term frequency – inverse document frequency)[19], which assigns a weight to each count reflecting how important a word is with respect to our corpus of lyrics. For a given word in a given lyrics, the value increases if the word appears frequently in the lyrics and is offset by the number of lyrics that contain the word in our dataset to adjust for the fact that some words are more frequently used.

3.3. Models

For Regression and Classification tasks, we went through a systematic process of trying quick and dirty models on feature subsets (song metadata, artist metadata and audio features) and/or on the whole set. Then, we used various feature selection methods to finally optimize some of the hyperparameters of our models (mostly for regularization) using cross validation. In the end, we experimented the addition of lyrics features to our best models and see if it could improve the overall performance or not. Each model was trained and tested on different subsets of our dataset (using the holdout cross validation method[20]) as well as on a reduced dataset to exclude outliers by selecting tracks with a Song Popularity value above 40 (indeed, the median for that feature is 82 and Q1 is 77). Regression models were trained and tested on the same training and testing sets for comparison. We did the same for classifiers. All models were build using Scikit-Learn, Statsmodels[21] and Keras[22] libraries.

[16] Googletrans: https://pypi.org/project/googletrans/
[17] NLTK: https://www.nltk.org/
[18] Bag-of-words: https://en.wikipedia.org/wiki/Bag-of-words_model
[19] TF-IDF: https://en.wikipedia.org/wiki/Tf%E2%80%93idf
[20] Holdout cross-validation: https://en.wikipedia.org/wiki/Cross-validation_(statistics)#Holdout_method
[21] Statsmodels: https://www.statsmodels.org/stable/index.html
[22] Keras: https://keras.io/

3.3.1. Multiple Linear Regression

Linear Regression[23] is linear solution to model the relationship between a continuous variable (the target or dependent variable) and several other variables (explanatory or independent variables). This model is easily understandable and allows for a good understanding of the role played by each feature, looking at the coefficients. Due to scale differences between our variables, we scaled and normalized our features to compare coefficients and improve convergence.

Features were selected through the following methods:

- **Lasso Regularization**[24]: this method consists in adding the L1 norm to our cost function with the effect of shrinking the coefficients towards 0. Increasing the parameter lambda results in decreasing the complexity of the model (adding bias) to prevent from overfitting. What is interesting with this method is that it constrains some of the coefficients to be equal to 0, which is the same as selecting features. The parameter lambda is tuned using 5-fold cross-validation on the training set with the R2 metric as a scorer.
- **Forward Selection with adjusted R2**[25]: we start with no variables in the model and we test the addition of features. The variable that gives the best improvement in adjusted R2 (measured on the training set) is added to the model. The operation is repeated until no further improvement. We chose to improve the adjusted R2 on the training set instead of a validation set because the idea was to get something on the training set before generalizing the model. Also, out dataset is quite small and cross-validation sets can return very different results.
- **Backward Elimination with adjusted R2**[26]: the process is similar to what we have described above. We start with every feature and by removing each feature one by one, we choose to withdraw the one that gave the best adjusted R2 while being removed. The process is then repeated until no further improvement.

[23] Linear Regression: https://en.wikipedia.org/wiki/Linear_regression
[24] Lasso Regularization: https://en.wikipedia.org/wiki/Lasso_(statistics)
[25] Forward selection: https://en.wikipedia.org/wiki/Stepwise_regression
[26] Backward elimination: https://en.wikipedia.org/wiki/Stepwise_regression

We tried various models on various subsets also implementing Lasso and Ridge Regularizations (L1 and L2) to prevent our models from overfitting. Lambda parameters were tuned using 5-fold cross-validation:

- Ordinary Least Squares (OLS) using the song metadata subset without and with a threshold.
- OLS using the artist metadata subset without and with a threshold and then removing features that are highly correlated.
- OLS using the audio feature subset without and with a threshold and then removing features that are highly correlated.
- Lasso Regression using the whole feature set without and with a threshold. We used Scikit Learn's LassoCV[27], which can automatically select the parameter lambda using a validation method.
- OLS using the features selected by Lasso.
- Ridge Regression using the whole feature set with a threshold.
- Ridge Regression using the features selected by Lasso.
- OLS using the features selected by forward selection and backward elimination.
- Ridge Regression using the features selected by both previous methods.
- OLS, Ridge Regression and Lasso Regression only using the lyrics set.
- Lasso Regression using the whole feature set combined with a selection of words.

We evaluated our models using various approaches and metrics. For our first models, we used an inferential statistics approach, looking at the followings on the training set:

- **R2 – Coefficient of determination**[28]: the proportion of the variance of our target variable explained by the linear model (and the independent variables).
- **Adjusted R2**[29]: it adjusts the previous metric for the number of variables in the model relatively to the number of examples.
- **Fisher's f-test p-value**[30]: for regression problems, the Fisher's f-test can assess the statistical significance of our regression (Is the group of selected variables jointly significant?).

[27] LassoCV: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LassoCV.html
[28] R2 – Coefficient of determination: https://en.wikipedia.org/wiki/Coefficient_of_determination
[29] Adjusted R2: https://en.wikipedia.org/wiki/Coefficient_of_determination
[30] Fisher's f-test: https://en.wikipedia.org/wiki/F-test

- **Student's t-test p-value**[31]: for regression problems, the Student's t-test can assess the statistical significance of a variable in the model (Is the associated coefficient significantly different from 0?).

These metrics were useful to assess the relevance of using linear regression with a particular subset before trying to generalize. To compare the models built with hyperparameter tuning (lambda parameter for regularization) and feature selection, we used the following metrics on the training and testing sets:

- **Root-mean-square error (RMSE)**[32]: it measures the square root of the average squared difference between the estimated values and the true values. The benefit of using the square root is that the result is in the same unit as the target variable.
- **R2 – Coefficient of determination**

Also, we compared our results to a dummy linear model, which always returns the mean value of the target feature on the training set.

3.3.2. Logistic Regression

Logistic Regression[33] can be used to model the probability of a given class (binary dependent variable) from the image of a linear combination of independent variables under the sigmoid function. A cut-off value is then defined to determine if the output probability corresponds to the class 1 or 0. For multiclassification tasks, the one-against-all[34] method can be used. Due to scale differences between our variables, we scaled and normalized our features.

Features were selected through the following methods:

- **Lasso Regularization**: as for linear regression, a L1 norm penalty is added to our cost function. That time, the parameter lambda was tuned using stratified 5-fold cross-validation, which is similar to what we have used before, but each partition of the training set contains approximately the same proportions of classes. This is helpful in the context of unbalanced datasets. We also defined our own scoring function, that returns the F1 score for our Hit class. Indeed, the default scorer in Scikit-Learn is the accuracy metric, which is not suited for unbalanced classes and the default F1 scorer

[31] Student's t-test: https://en.wikipedia.org/wiki/Student%27s_t-test
[32] Root-mean-square error
[33] Logistic Regression: https://en.wikipedia.org/wiki/Logistic_regression
[34] One-against-all: https://en.wikipedia.org/wiki/Multiclass_classification#One-vs.-rest

returns the weighted average F1 score of each class. That scorer was used every time we had to use cross-validation.

- **Random Forest**: during the training of a RandomForest model, we can compute the decrease in Gini impurity[35] resulting from a particular feature. This allows us to classify features and then select the most significant ones looking at our scoring function combined with a validation strategy.

We tried models on different subsets also implementing Lasso and Ridge Regularizations (L1 and L2) to prevent from overfitting. Lambda parameters were tuned using a stratified 5-fold cross-validation method:

- Logistic Regression (LR) using the song metadata subset without a threshold.
- LR using the artist metadata subset without a threshold.
- LR using the audio feature subset without a threshold.
- LR combining the song metadata and artist metadata subsets, without a threshold.
- LR combining the song metadata and audio feature subsets, without a threshold.
- LR combining the artist metadata and audio feature subsets, without a threshold.
- LR using the 3 subsets, without and with a threshold.
- Logistic Ridge Regression using the 3 subsets, without a threshold.
- Logistic Ridge Regression using the features selected by the Lasso method, without a threshold. We used Scikit Learn's LogisticRegressionCV[36], which can automatically select the parameter lambda using a validation method.
- Logistic Ridge Regression using the features selected by the Random Forest method, without a threshold. The lambda parameter is tuned using the same method as above.
- Logistic Lasso Regression using features selected by both methods in addition to a selection of words (performed with Logistic Lasso Regression on the lyrics set without TF-IDF), without a threshold.

The models were evaluated on a test set that was created using a stratified strategy (same method as for the tuning of hyperparameters but splitting into two the entire dataset). We looked at the following metrics that are computed using the confusion matrix[37]:

[35] Gini Impurity: https://en.wikipedia.org/wiki/Decision_tree_learning#Gini_impurity
[36] LogisticRegressionCV: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegressionCV.html
[37] Confusion matrix: https://en.wikipedia.org/wiki/Confusion_matrix

- **Precision**[38]: the precision metric measures the proportion of predicted positive values that were actually true positive values. It answers the question how many selected items are relevant?
- **Recall**[39]: the recall metric measures the proportion of true positive values that were actually predicted positive. It answers the question how many relevant items are selected?
- **F1 score**[40]: this metric is the harmonic mean of the precision and recall metrics. The harmonic mean is used to penalize for very low values of precision or recall. Therefore, the F1 score can only reach its maximum of 1 if both metrics are equal to 1. It is especially useful to measure the performance of a classifier with unbalanced classes. Indeed, the accuracy, which the default metric for classification, is no longer appropriate in that context.

Also, we compared our results to a dummy classifier using Scikit Learn. We chose a classifier that adopts a stratified strategy, meaning that it predicts classes based on the proportions of the training set.

3.3.3. K Nearest Neighbours

K Nearest Neighbours is a method that can be used in the context of classification. It outputs a given class based on the k nearest datapoints of the training set. The class is determined the majority among the neighbours. Due to scale differences between our variables, we scaled and normalized our features (it is required for this model).

The resulting selected features that we obtained previously can be used for any classifiers. Therefore, we decided to keep 3 lists: selected by Lasso, selected by Random Forest and the intersection of the two selections.

We tried models on different subsets of features:

- K Nearest Neighbours (KNN) using the whole feature set without and with a threshold.
- KNN using the features selected by both methods (intersection), without a threshold.
- KNN using the features selected by both methods (intersection), with a threshold

[38] Precision: https://en.wikipedia.org/wiki/Precision_and_recall
[39] Recall: https://en.wikipedia.org/wiki/Precision_and_recall
[40] F1 score: https://en.wikipedia.org/wiki/F1_score

Then, we decided to use the Scikit Learn's GridSearchCV[41] which is useful for hyperparameter tuning. This tool allows us to test a grid of hyperparameters and to get the best combination of parameters using a validation strategy (we used stratified 5-fold cross-validation).

- We tuned the following hyperparameters: the number of neighbours, the use of weights to rank the neighbours and the type of distance to use (Manhattan distance or Euclidean distance)
- Resulting KNN using the whole feature set with a threshold

The models were evaluated using the same testing strategy and the same metrics as Logistic Regression.

3.3.4. Random Forest

Random Forest[42] is an ensemble learning method that can be used for classification (it means that it uses several classifiers to end up with a prediction – Decision Trees in that case). This method is robust to outliers and it does not need feature scaling. Each decision tree is used to output a value (a class) and the most recurrent class is predicted by the model. Each tree does not use the same combination of samples from the training set nor the same features to produce a decision. There is a high risk of overfitting the training set using that method.

We did not perform a selection of features for that model. Indeed, we used the set that was previously selected using Random Forest during Logistic Regression.

We tried models on different subsets of features:

- Random Forest (RF) with 100 estimators (Decision Trees) using the whole feature set, without a threshold.
- RF with 100 estimators (Decision Trees) using the selected features, without a threshold.

Then, we decided to tune the hyperparameters of our model with the same method as above.

- We tuned the following hyperparameters: the number of estimators, the maximum number of features to consider to look for the best split, the maximum depth of a tree,

[41] GridSearchCV: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
[42] Random Forest: https://en.wikipedia.org/wiki/Random_forest

the minimum number of samples to split an internal node, the minimum number of samples for a leaf node, the use of bootstrap samples to build a tree and various class weight combinations.

- Resulting RF using the selected feature set without a threshold.
- RF using the selected features in addition to the words that were selected during Logistic Regression.

The models were evaluated using the same testing strategy and the same metrics as Logistic Regression.

3.3.5. Support Vector Machines

Support Vector Machines is a model that determines a separating hyperplane (decision boundary) that maximizes the nearest points of each class to that hyperplane. A non-linear decision boundary can be created through the use of a kernel function. We used a gaussian kernel (gaussian radial basis function) and a linear kernel. In practice, the linear kernel is used when the number of features is large compared to the size of the training set.

We used the following selection methods:

- **Lasso Regularization** applied to an SVM with linear kernel: for the SVM, the regularization parameter works as the inverse of the lambda parameter (increasing C results in a more complex model and increases variance).
- We also used the features that were selected by the **Random Forest** model.

We tried models on different subsets of features:

- Support Vector Machines (SVM) with linear kernel using the whole feature set, without and with a threshold.
- SVM with gaussian kernel using the whole feature set, without and with a threshold.
- SVM with linear kernel using the features selected by Lasso Regularization, without a threshold
- SVM with gaussian kernel using the features selected by Lasso Regularization, without a threshold

- SVM with linear kernel using the features selected by Random Forest, without a threshold

Then, we tuned the hyperparameters of SVM, still using the same validation strategy and both sets of selected features:

- We tuned the following hyperparameters: The C parameter, the kernel of the model (linear or gaussian), the use of class weights and the gamma parameter (for the gaussian kernel)
- The resulting models were then used on their respective feature sets
- Improvement of the best-case model by further tuning the parameters
- SVM with linear and gaussian kernel using features selected by lasso in addition to the words that were selected during Logistic Regression.

The models were evaluated using the same testing strategy and metrics as Logistic Regression.

3.3.6. Multilayer Perceptron

The Multilayer Perceptron[43] is an example of feedforward neural network that can be used for regression and classification tasks. It is composed of at least an input and output layers in addition to one or more hidden layers. Each layer is composed of a given number of units. In the Perceptron model, all layers are fully connected. We chose to train a model with one hidden layer of 40 units, an input layer of 33 units (the number of features selected to train the model on) and an output layer of 1 unit. Due to scale differences between our variables, we scaled and normalized our features.

We wanted to try out different subsets, but we ended up using only the Random Forest's one and tried to improve the network as much as possible given that feature subset.

In a neural network, there are many parameters that can be tuned to further improve the performance of the model. A validation set was created from the training set to be used during the fitting process. It allows us to compare for each epoch the loss on the training and validation sets. In addition to that, we created a F1 scoring metric that we could also monitor while fitting

[43] Multilayer Perceptron: https://en.wikipedia.org/wiki/Multilayer_perceptron

the model. Indeed, for classification tasks Keras does not provide a built-in F1 score. The various parameters we tried were:

- **The number of hidden layers and units**: to reduce or add complexity to the model.
- **Optimization algorithm**: RMSprop[44] and Adam[45] are two stochastic gradient descent methods, which perform well in the context of large datasets. Given the size of our dataset, we could have used the original Batch Gradient Descent[46] algorithm but we could not find it on the library.
- **L2 Regularization**: once again, a penalization can be used to prevent our model from overfitting, which is a high risk given the size of the dataset. A parameter lambda has to be selected.
- **Class weights**: for neural networks, this parameter is essential in the context of unbalanced classes.
- **Kernel initializer**: this parameter sets the method used to initialize the weights/parameters of the model.
- **Dropout**: this can be used to randomly cut off a given percentage of the units of a layer. It also helps to prevent overfitting.
- **Activation function**: it is the function that determines the output of each unit of the network. For the output layer we used the sigmoid function, but we tried out the rectified linear unit[47] and the hyperbolic tangent[48] functions for the other units.

The models were evaluated using a validation set and the same metrics as Logistic Regression. We also plotted the learning curves for our model to monitor the fitting process.

[44] RMSProp: https://en.wikipedia.org/wiki/Stochastic_gradient_descent#RMSProp

[45] Adam: https://en.wikipedia.org/wiki/Stochastic_gradient_descent#Adam

[46] Batch Gradient Descent: https://en.wikipedia.org/wiki/Gradient_descent

[47] Rectified Linear Unit: https://en.wikipedia.org/wiki/Rectifier_(neural_networks)

[48] Hyperbolic Tangent: https://en.wikipedia.org/wiki/Hyperbolic_functions

4. Results

4.1. Data Analysis

4.1.1. Visualization

From data analysis, we tried to answer some of our questions and to understand our dataset. We will present below the main artists of 2018 and 2019, the main genres, the main countries and the most used words in the lyrics.

Figure 3: Most Famous Artists of Spotify's Top 2018 and 2019

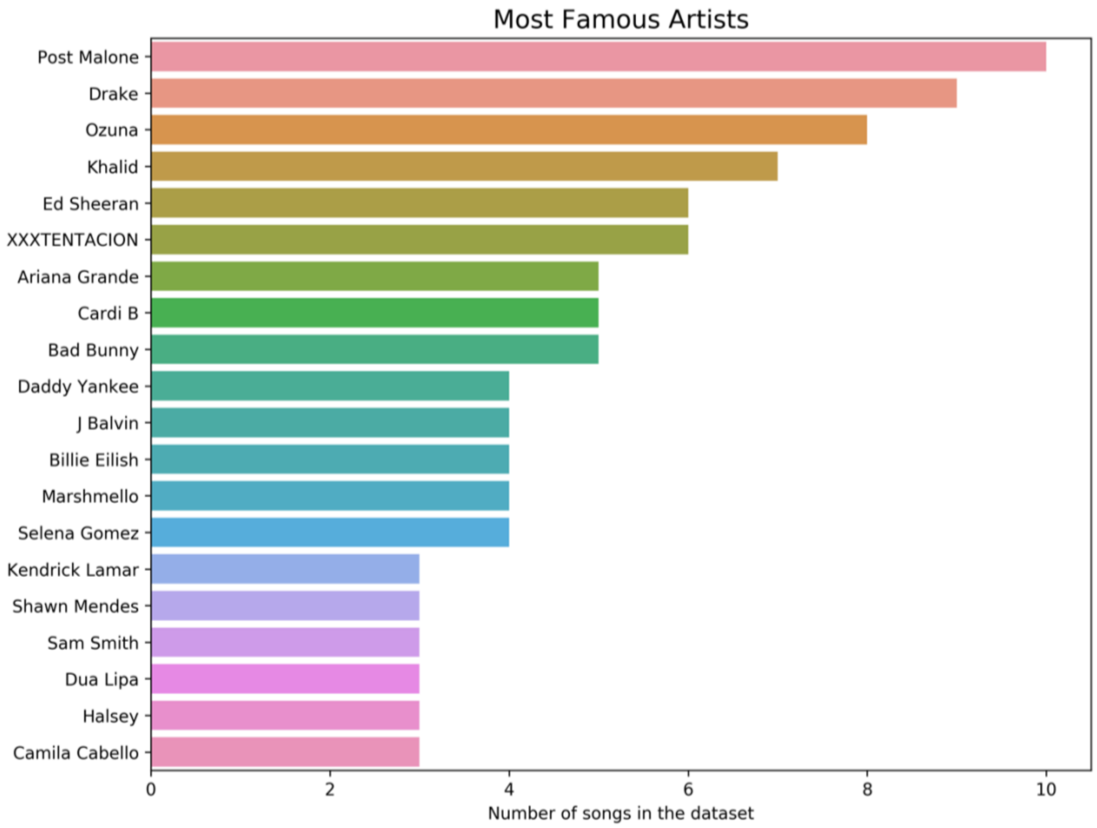


Figure 4: Main genres of music of Spotify's Top 2018 and 2019

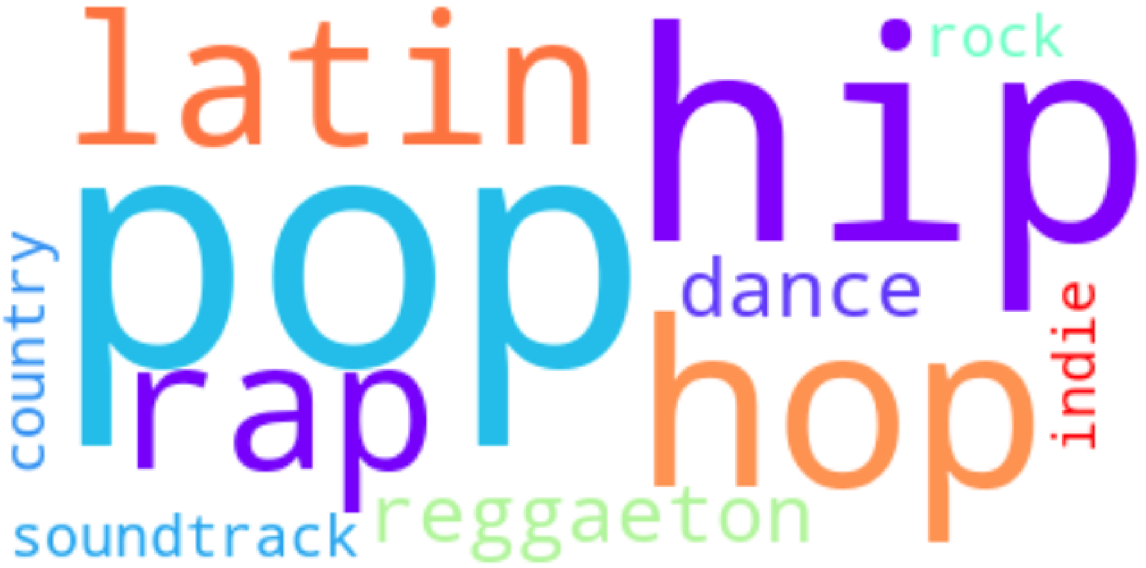
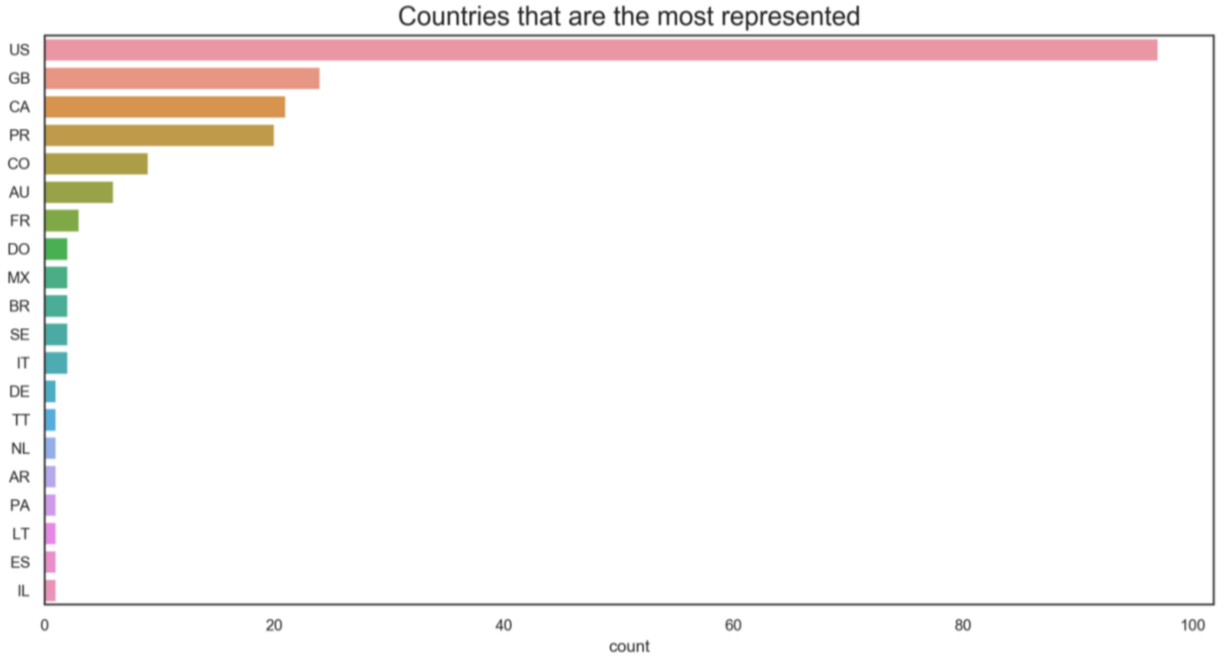
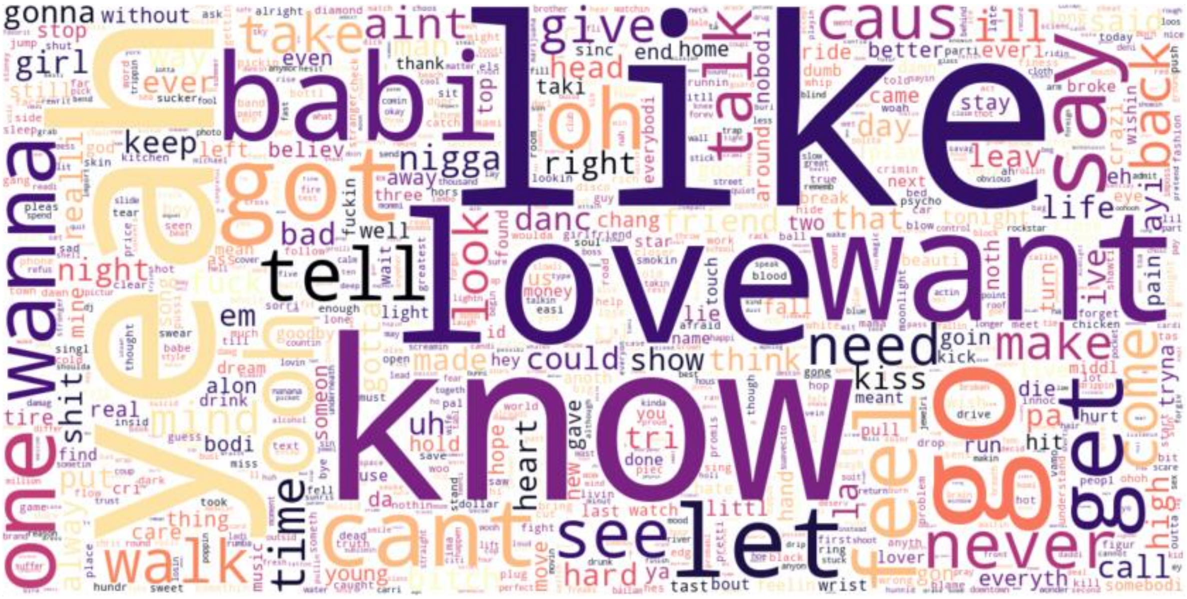


Figure 5: Main artist countries of Spotify's Top 2018 and 2019



From these figures, we can state that the dominant genres of today's Hit songs are Pop and Hip-Hop/Rap, which is confirmed by the main artists of our dataset. Obviously, many famous artists come from anglophone countries, with a clear dominance of the United States. It is interesting to see the dominance of one Central American "country": the Free Associated State of Puerto Rico, which is an unincorporated territory of the United States. Alongside with Colombia and The Dominican Republic.

Figure 6: Wordcloud for the lyrics of Spotify's Top 2018 and 2019



Interesting words are: like, know, love, want, wanna, babi, cant, feel, never, need, give, friend, girl, man, hard, life, heart, fuck, shit, bad, kiss, good, better, lie, feel or ill. These words do not refer to a specific lexical field, but they are linked to some general ideas such as: love, sex, negative feelings (missing or need for something) or positive feelings. Maybe the lyrics of a Hit song should not refer to specific ideas because they need to concern a maximum amount of people. In addition to that, evoking positive of negative (general) feelings is an easy way to catch up the attention. Further and deeper sentimental analysis should be performed but this is not the main subject. It could be interesting to compare predominant Hit and Non-Hit words.

4.1.2. Univariate analysis

Our features can be summarized into the following table. We can observe that our dataset is constituted mostly of highly popular songs, which might be an issue for a regression analysis. Also, some variables have extremely low variance and they might not be good candidates for our Machine Learning project (duration for instance). Our Google Trends features are normally distributed (slightly skewed). Other features have outliers: followers or age (maybe this is because our dataset is small). Regarding the audio characteristics of our songs, it appears that they are not acoustic (as of today, the music industry often use electronic instruments or acoustic songs which are transformed), they seem to be danceable, energetic and to contain vocals (low values of instrumentalness).

Figure 7: Distributions of quantitative features

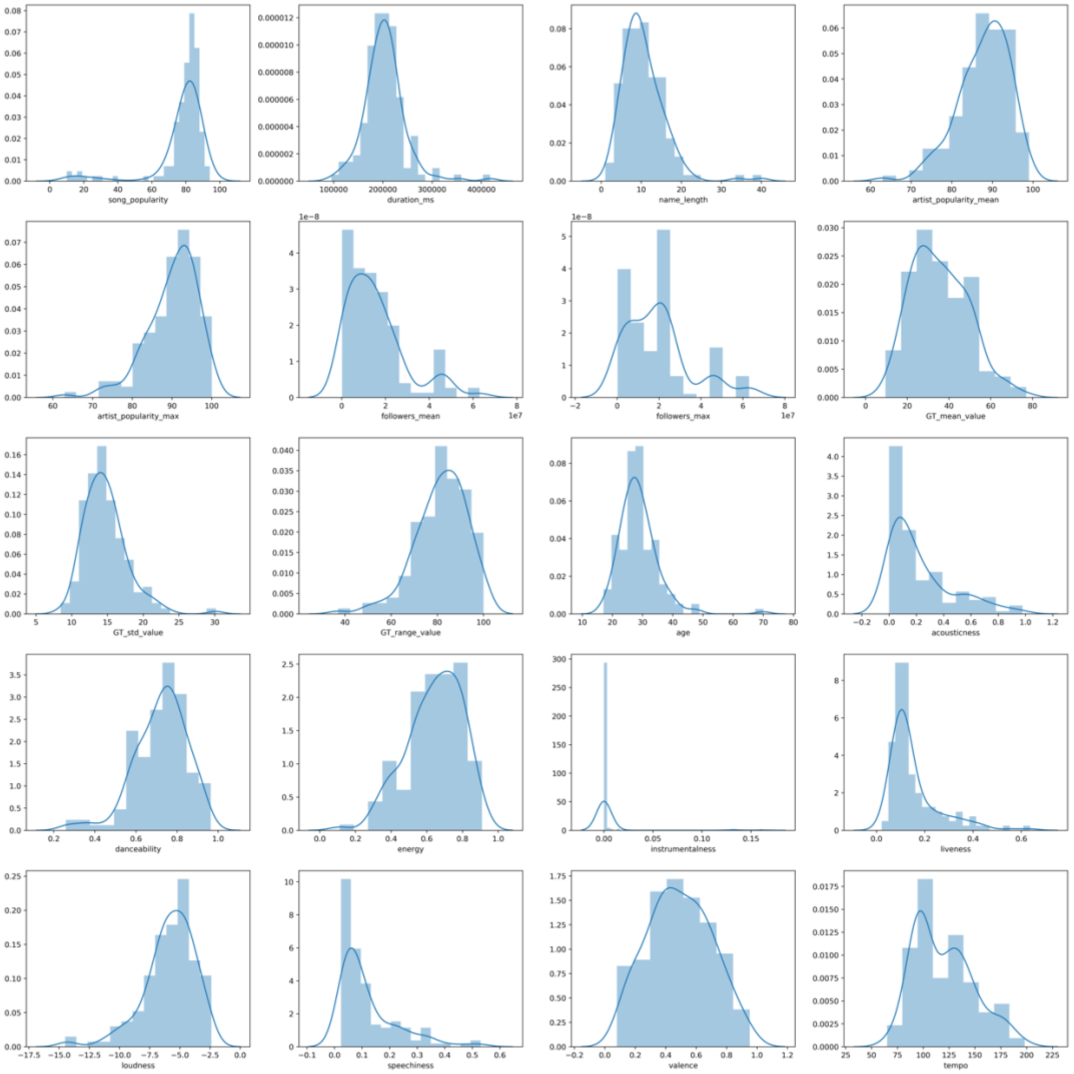
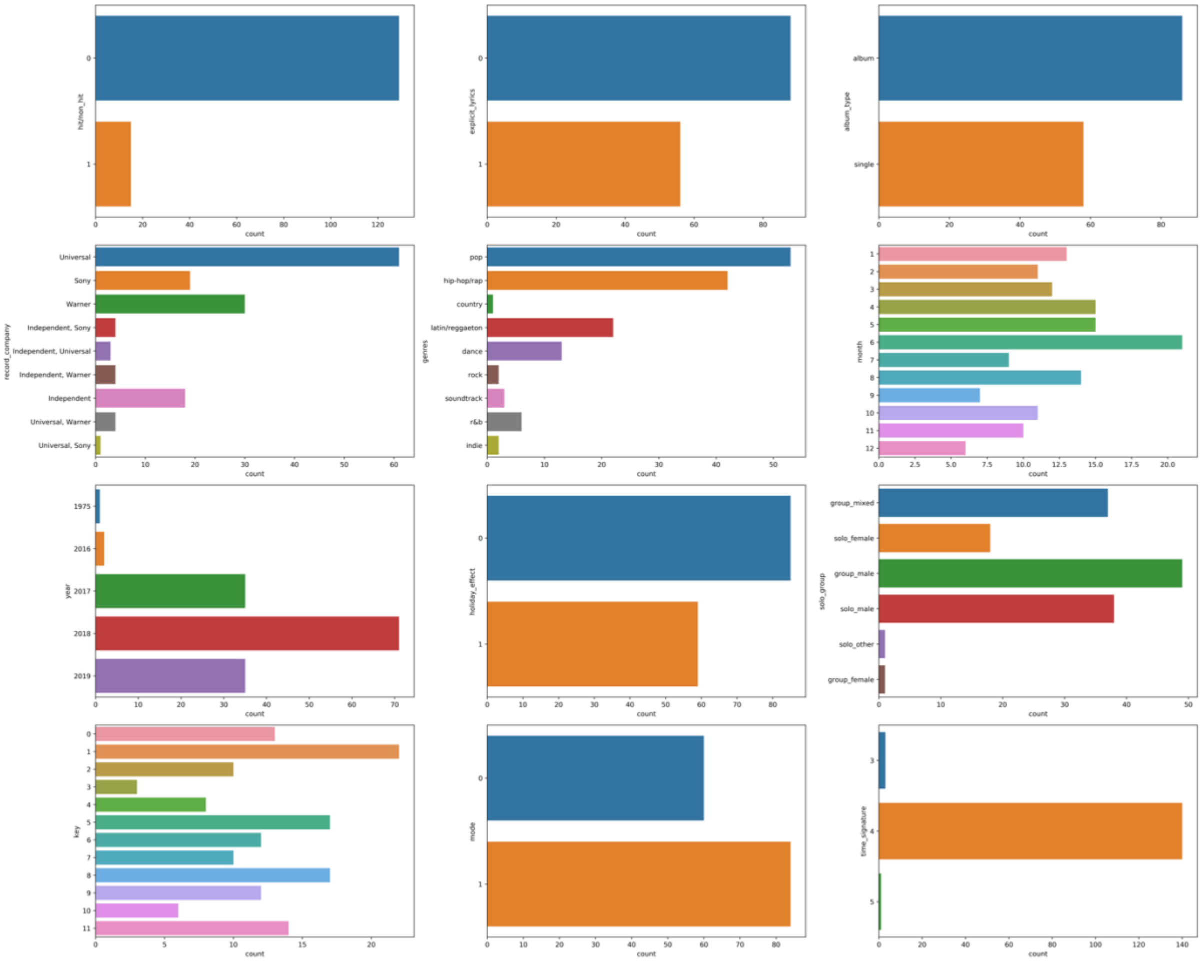


Table 2: Descriptive Statistics for quantitative features

	count	mean	std	min	25%	50%	75%	max
song_popularity	144	77,3	16,1	10,0	77,0	82,0	85,0	94,0
duration_ms	144	203 200,0	41 010,0	95 470,0	180 700,0	201 300,0	220 200,0	417 900,0
name_length	144	10,5	5,3	1,0	7,0	10,0	13,0	40,0
artist_popularity_mean	144	87,8	6,3	63,0	84,0	89,0	92,6	99,0
artist_popularity_max	144	90,1	6,4	63,0	87,0	91,5	95,0	100,0
followers_mean (in thousands)	144	15 520,0	13 590,0	127,9	5 668,0	12 690,0	20 280,0	62 940,0
followers_max (in thousands)	144	19 400,0	15 510,0	245,7	7 028,0	19 240,0	24 650,0	62 940,0
GT_mean_value	144	35,7	13,5	9,5	25,4	34,4	45,3	77,0
GT_std_value	144	14,7	3,0	8,5	12,7	14,3	16,2	30,2
GT_range_value	144	81,7	11,3	37,0	75,4	83,0	90,0	100,0
age	144	28,7	6,5	17,0	25,0	28,0	32,0	70,0
acousticness	144	0,2	0,2	0,0	0,0	0,1	0,3	1,0
danceability	144	0,7	0,1	0,3	0,6	0,7	0,8	1,0
energy	144	0,6	0,2	0,1	0,5	0,7	0,8	0,9
instrumentalness	144	0,0	0,0	-	-	-	0,0	0,2
liveness	144	0,2	0,1	0,0	0,1	0,1	0,2	0,6
loudness	144	(5,9)	2,1	(14,5)	(6,8)	(5,7)	(4,4)	(2,4)
speechiness	144	0,1	0,1	0,0	0,0	0,1	0,1	0,5
valence	144	0,5	0,2	0,1	0,3	0,5	0,6	1,0
tempo	144	119,8	28,7	64,9	95,9	118,5	137,2	198,1

Regarding our quantitative features, we noticed that: albums are dominant compared to singles, Universal Music is the main recording house in 2018 and 2019, pop music dominates the market and most songs were released during what we defined as the "Holiday Period" (those 4 months represent more than 1/3 of the dataset).

Figure 8: Countplot of quantitative features

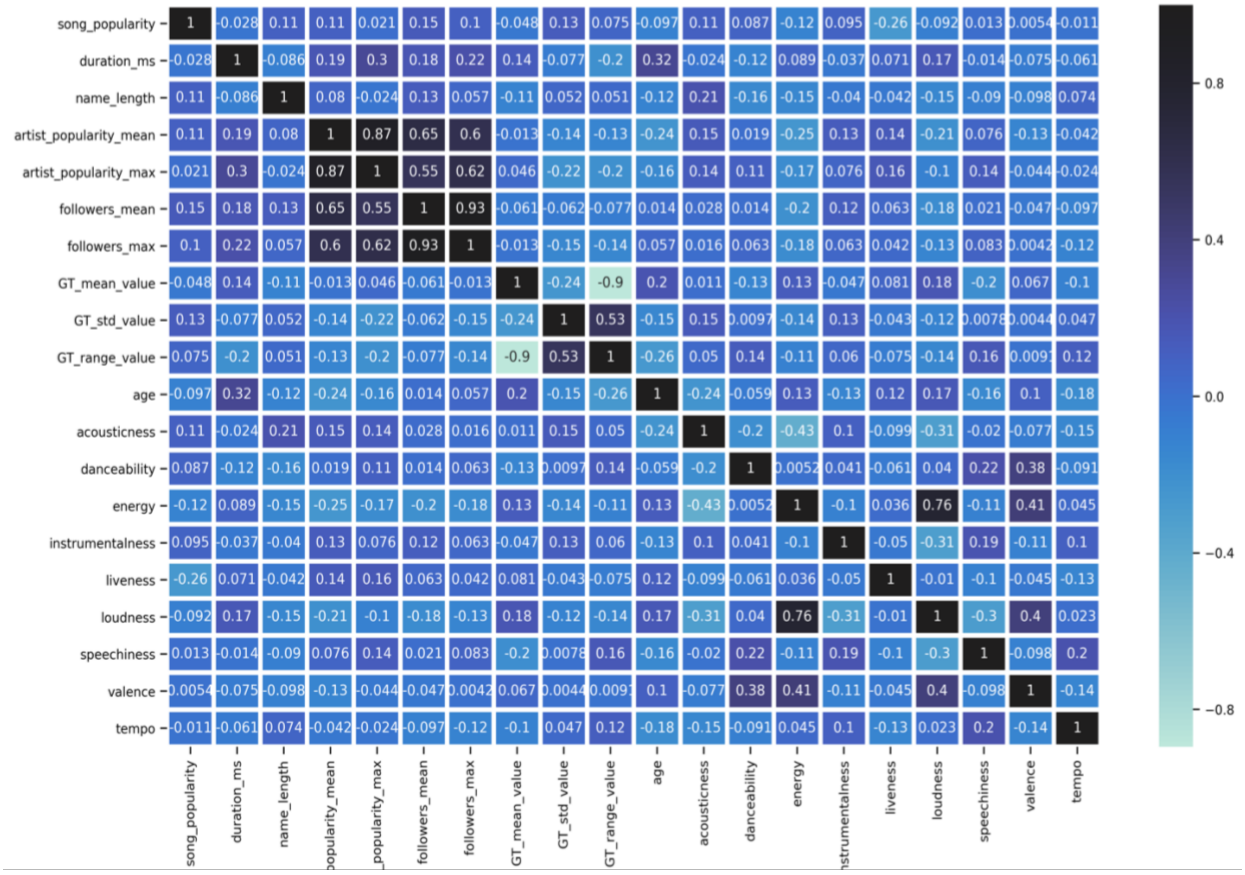


4.1.3. Bivariate analysis

While looking at the linear relationship between our quantitative features and paying attention to our target variable (Song Popularity), we get poor results. What is strange is that the correlations between our continuous target and Artist Popularity features are low while they are supposed to be derived from it. Indeed, the feature is computed by Spotify based on the popularity value for all the artist's songs. We would have expected a higher value for those features. We also notice some high correlations between our independent variables – above the obvious correlations between the features that are derived from the same variable. This is the case for:

- Artist Popularity and Followers: max has a correlation of 0.62 and mean of 0.65
- Energy and Acousticness: -0.43
- Energy and Valence: 0.41
- Energy and Loudness: 0.76
- Valence and Loudness: 0.4

Figure 9: Correlation matrix



	song_popularity	duration_ms	name_length	popularity_mean	_popularity_max	followers_mean	followers_max	GT_mean_value	GT_std_value	GT_range_value	age	acousticness	danceability	energy	nstrumentalness	liveness	loudness	speechiness	valence	tempo
song_popularity	1	-0.028	0.11	0.11	0.021	0.15	0.1	-0.048	0.13	0.075	-0.097	0.11	0.087	-0.12	0.095	-0.26	-0.092	0.013	0.0054	-0.011
duration_ms	-0.028	1	-0.086	0.19	0.3	0.18	0.22	0.14	-0.077	-0.2	0.32	-0.024	-0.12	0.089	-0.037	0.071	0.17	-0.014	-0.075	-0.061
name_length	0.11	-0.086	1	0.08	-0.024	0.13	0.057	-0.11	0.052	0.051	-0.12	0.21	-0.16	-0.15	-0.04	-0.042	-0.15	-0.09	-0.098	0.074
artist_popularity_mean	0.11	0.19	0.08	1	0.87	0.65	0.6	-0.013	-0.14	-0.13	-0.24	0.15	0.019	-0.25	0.13	0.14	-0.21	0.076	-0.13	-0.042
artist_popularity_max	0.021	0.3	-0.024	0.87	1	0.55	0.62	0.046	-0.22	-0.2	-0.16	0.14	0.11	-0.17	0.076	0.16	-0.1	0.14	-0.044	-0.024
followers_mean	0.15	0.18	0.13	0.65	0.55	1	0.93	-0.061	-0.062	-0.077	0.014	0.028	0.014	-0.2	0.12	0.063	-0.18	0.021	-0.047	-0.097
followers_max	0.1	0.22	0.057	0.6	0.62	0.93	1	-0.013	-0.15	-0.14	0.057	0.016	0.063	-0.18	0.063	0.042	-0.13	0.083	0.0042	-0.12
GT_mean_value	-0.048	0.14	-0.11	-0.013	0.046	-0.061	-0.013	1	-0.24	-0.9	0.2	0.011	-0.13	0.13	-0.047	0.081	0.18	-0.2	0.067	-0.1
GT_std_value	0.13	-0.077	0.052	-0.14	-0.22	-0.062	-0.15	-0.24	1	0.53	-0.15	0.15	0.0097	-0.14	0.13	-0.043	-0.12	0.0078	0.0044	0.047
GT_range_value	0.075	-0.2	0.051	-0.13	-0.2	-0.077	-0.14	-0.9	0.53	1	-0.26	0.05	0.14	-0.11	0.06	-0.075	-0.14	0.16	0.0091	0.12
age	-0.097	0.32	-0.12	-0.24	-0.16	0.014	0.057	0.2	-0.15	-0.26	1	-0.24	-0.059	0.13	-0.13	0.12	0.17	-0.16	0.1	-0.18
acousticness	0.11	-0.024	0.21	0.15	0.14	0.028	0.016	0.011	0.15	0.05	-0.24	1	-0.2	-0.43	0.1	-0.099	-0.31	-0.02	-0.077	-0.15
danceability	0.087	-0.12	-0.16	0.019	0.11	0.014	0.063	-0.13	0.0097	0.14	-0.059	-0.2	1	0.0052	0.041	-0.061	0.04	0.22	0.38	-0.091
energy	-0.12	0.089	-0.15	-0.25	-0.17	-0.2	-0.18	0.13	-0.14	-0.11	0.13	-0.43	0.0052	1	-0.1	0.036	0.76	-0.11	0.41	0.045
instrumentalness	0.095	-0.037	-0.04	0.13	0.076	0.12	0.063	-0.047	0.13	0.06	-0.13	0.1	0.041	-0.1	1	-0.05	-0.31	0.19	-0.11	0.1
liveness	-0.26	0.071	-0.042	0.14	0.16	0.063	0.042	0.081	-0.043	-0.075	0.12	-0.099	-0.061	0.036	-0.05	1	-0.01	-0.1	-0.045	-0.13
loudness	-0.092	0.17	-0.15	-0.21	-0.1	-0.18	-0.13	0.18	-0.12	-0.14	0.17	-0.31	0.04	0.76	-0.31	-0.01	1	-0.3	0.4	0.023
speechiness	0.013	-0.014	-0.09	0.076	0.14	0.021	0.083	-0.2	0.0078	0.16	-0.16	-0.02	0.22	-0.11	0.19	-0.1	-0.3	1	-0.098	0.2
valence	0.0054	-0.075	-0.098	-0.13	-0.044	-0.047	0.0042	0.067	0.0044	0.0091	0.1	-0.077	0.38	0.41	-0.11	-0.045	0.4	-0.098	1	-0.14
tempo	-0.011	-0.061	0.074	-0.042	-0.024	-0.097	-0.12	-0.1	0.047	0.12	-0.18	-0.15	-0.091	0.045	0.1	-0.13	0.023	0.2	-0.14	1

However, linear correlation coefficients indicate a linear correlation between two variables and can be heavily influenced by outliers and the sample used. By removing "outliers", we obtain better correlations with the Song Popularity (Energy has a correlation of -0.35 and Artist Popularity Mean of 0.3), but still nothing great. It would probably be interesting to repeat the same analysis with a larger dataset and a normally distributed Song Popularity variable.

Figure 10: Correlations of independent variables with Song Popularity – with outliers

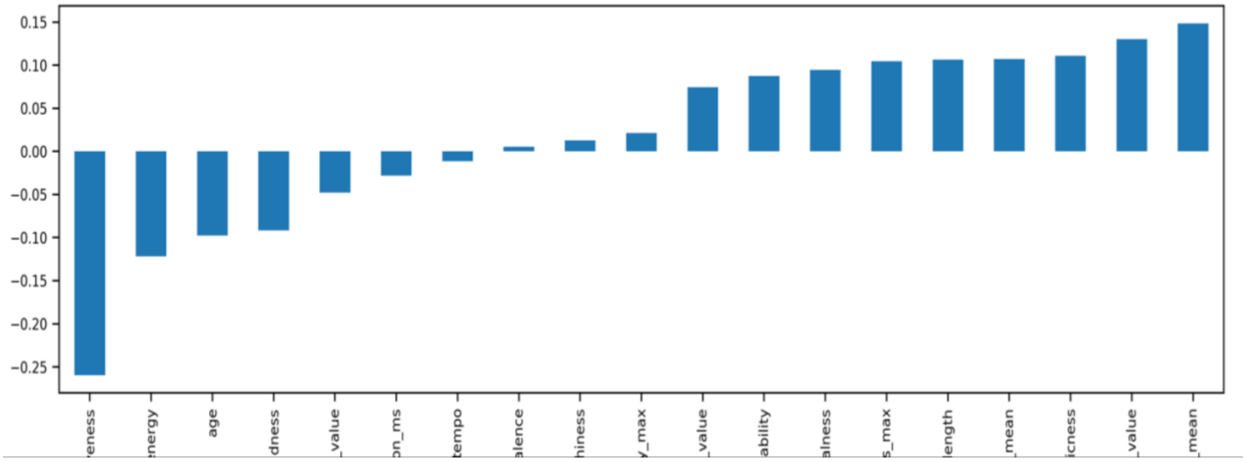
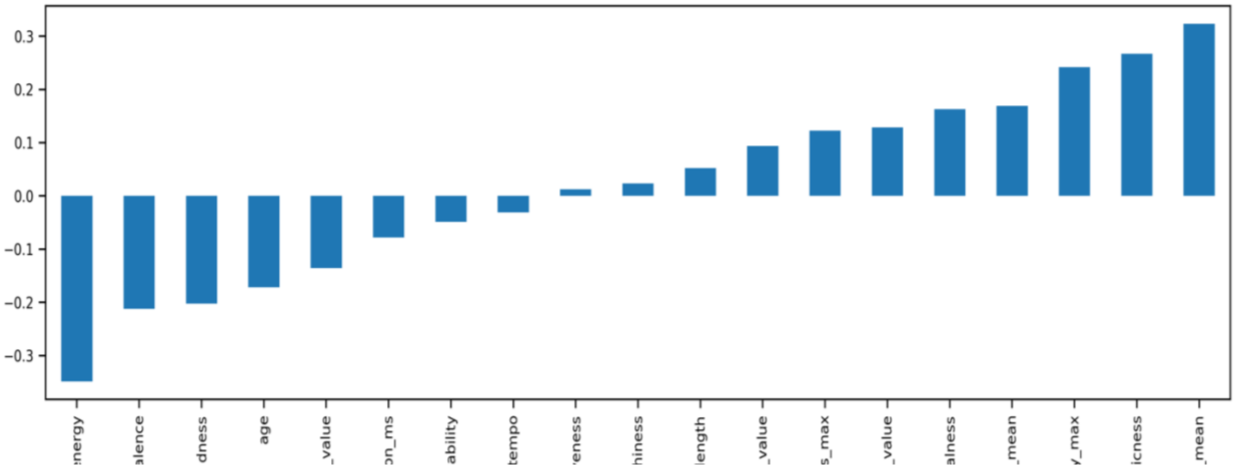


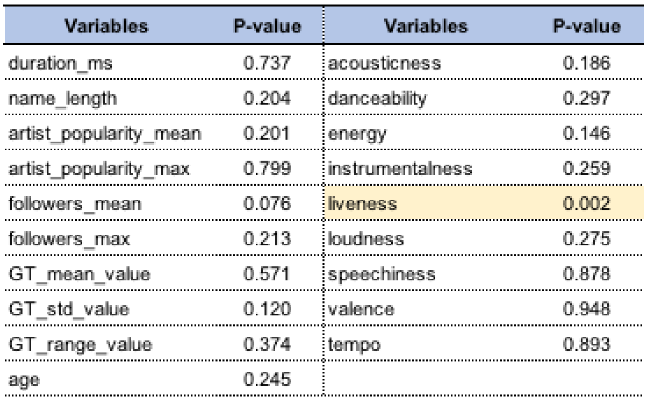
Figure 11: Correlations of independent variables with Song Popularity – without outliers



The Pearson’s correlation coefficient statistical test is established as follow:

- H0: The two samples are independent / H1: There is a dependency between the samples.
- Limits: Assume a Gaussian distribution and a linear relationship.
- Results: the only quantitative features that is statistically dependent with Song Popularity is Liveness with a p-value of 0.002.

Table 3: Pearson’s test results



Variables	P-value	Variables	P-value
duration_ms	0.737	acousticness	0.186
name_length	0.204	danceability	0.297
artist_popularity_mean	0.201	energy	0.146
artist_popularity_max	0.799	instrumentalness	0.259
followers_mean	0.076	liveness	0.002
followers_max	0.213	loudness	0.275
GT_mean_value	0.571	speechiness	0.878
GT_std_value	0.120	valence	0.948
GT_range_value	0.374	tempo	0.893
age	0.245		

To account for non-linear relationships, the Spearman correlation coefficient gives better results (but not great), especially for the Spearman's correlation coefficient statistical test (same method as above). It accounts for monotonic relationship. We get dependence for Artist Popularity Mean, Followers Mean, GT Std Value, GT Range Value, Age and Energy.

Figure 11: Spearman correlations with Song Popularity – with outliers

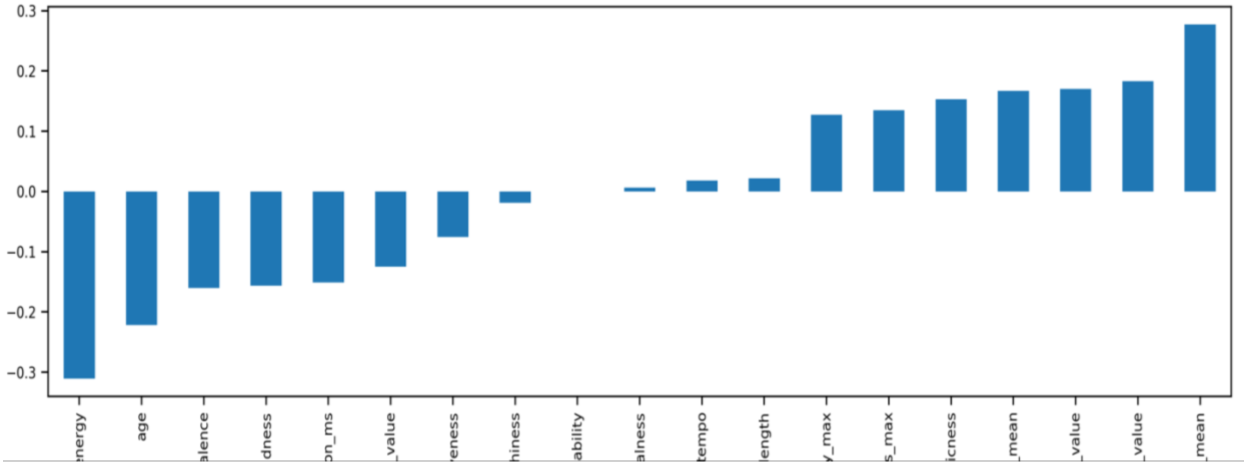


Figure 12: Spearman correlations with Song Popularity – without outliers

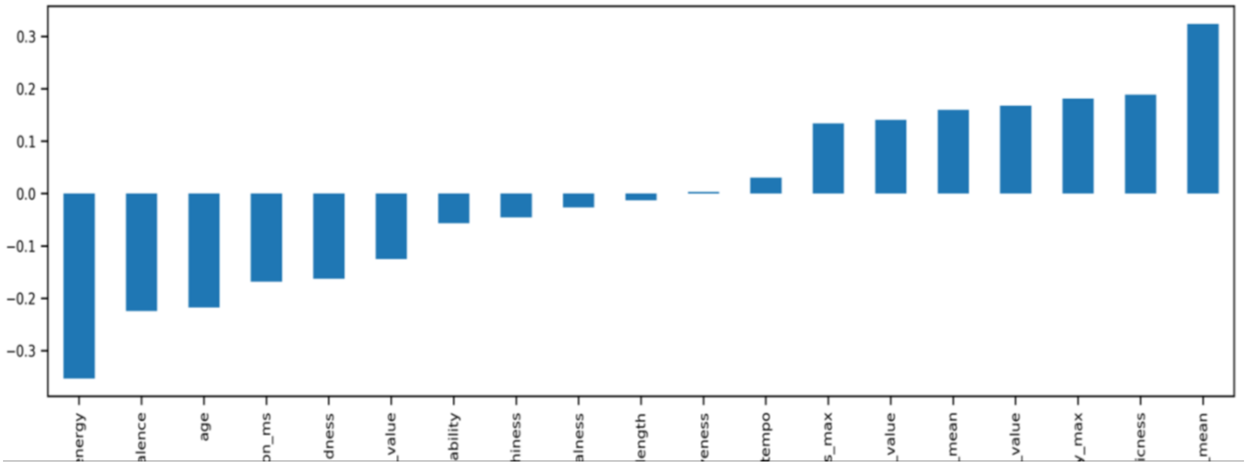
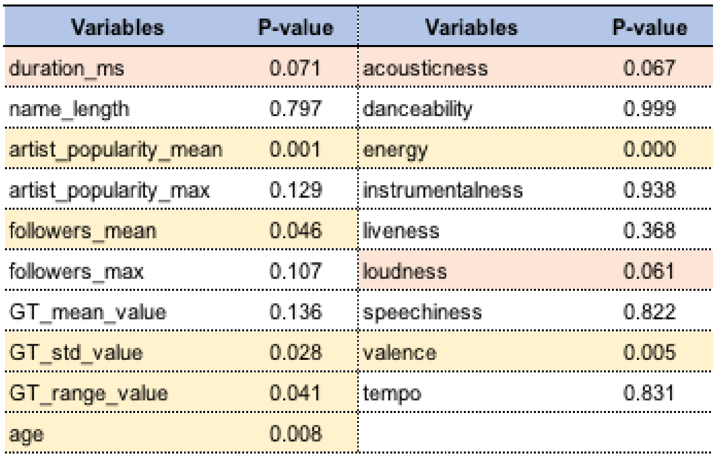


Table 4: Spearman's test results



Variables	P-value	Variables	P-value
duration_ms	0.071	acousticness	0.067
name_length	0.797	danceability	0.999
artist_popularity_mean	0.001	energy	0.000
artist_popularity_max	0.129	instrumentalness	0.938
followers_mean	0.046	liveness	0.368
followers_max	0.107	loudness	0.061
GT_mean_value	0.136	speechiness	0.822
GT_std_value	0.028	valence	0.005
GT_range_value	0.041	tempo	0.831
age	0.008		

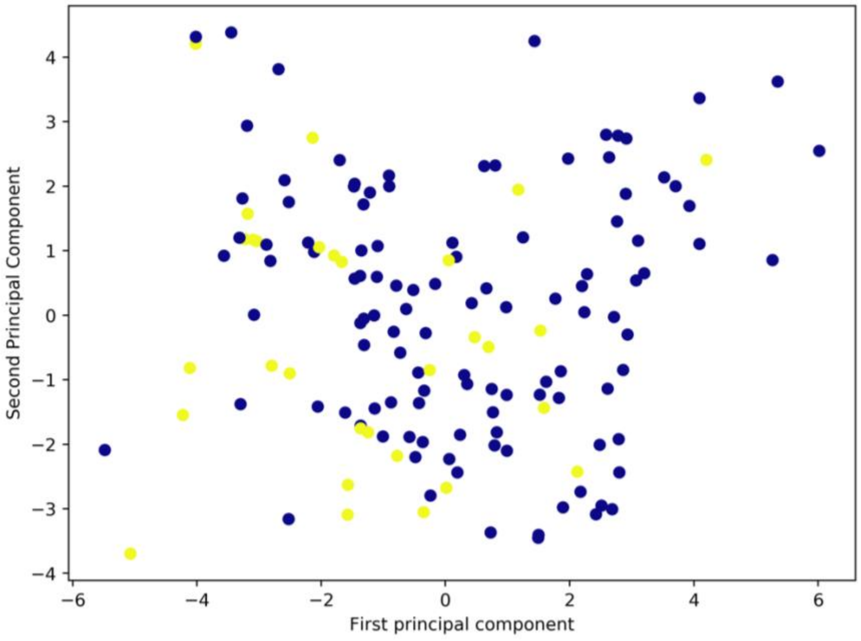
From that bivariate analysis, we can say that audio features do not have a linear correlation with the popularity of a song and that only some of them seem to have a relationship with our target: energy, valence, loudness and acousticness. Overall, with our tests, we obtain that metadata features have a stronger correlation (statistically) with Song Popularity than the audio features. However, those results should be investigated on a larger dataset.

4.1.4. Principal Component Analysis

On the following graph, the yellow colour matches Hits and the blue colour Non-Hits, based on the popularity feature that we created using the distribution of Song Popularity (top 25% is composed of Hits). The Song Popularity feature is not taken into account.

For the whole feature set, the two components explained 13.78% of the variation among our individuals (for 66 features). Looking at the absolute value of the component coefficients, the main variables to model the inertia of our dataset are: Loudness, Energy, Danceability, Explicit Lyrics, Genre Latin/Reggaeton, Single Type, Genre Hip-Hop/Rap and Genre Pop. In both components, among the 15th main coefficients in absolute value, we have genre features (Latin/Reggaeton, Hip-Hop/Rap), Group Gender features, Explicit Lyrics and GT Peak Values. Overall, we observe an absence of strong multicollinearity between our features because of the low variation explained by our first components and the potential homogeneity of our dataset.

Figure 13: Principal Component Analysis – whole feature set



Then, we performed Principal Component Analysis on our 3 subsets of features. We do not display the results here (see the notebook in Appendix) but we look at the main coefficients for each feature set. For the audio features, the 4 main variables of the 1st component are Energy, Valence, Loudness and Acousticness – the exact same audio features for which we found a dependency with the Spearman's statistical test. For the artist metadata subset, Artist Popularity Mean, GT Peak (before value), Group/Gender (solo male, mixed), GT Std Value and Range Value are the main features (looking at both components). And for the song metadata subset, we have Genre Hip-Hop/Rap, Genre Pop, Explicit Lyrics, Company Universal, Single Type, Name Length and Genre Latin/Reggaeton.

4.2. Selected features

4.2.1. Linear Regression

Table 5: Selected features for regression

Model	Parameter(s)	Feature Selected
Lasso Regularization (with "outliers")	Lambda = 2.41	Liveness, MX, Genre R&B, Energy
Lasso Regularization (without "outliers")	Lambda = 0.45	AU, Genre Latin/Reggaeton, Company Warner, FR, Genre Soundtrack, Energy, BR, Group Gender Solo Female, Genre R&B, Genre Rock, GT Std Value, Holiday Effect, GT Range Value, Group Gender Solo Male, Artist Popularity Mean, Company Universal, Acousticness
Forward Selection		Company Warner, Energy, AU, Genre Latin/Reggaeton, Acousticness, Loudness, FR, PR, Genre Soundtrack, Genre R&B, Group Gender Solo Female, Genre Hip-Hop/Rap, Peak After, Genre Dance, GT Range Value, ES, Artist Popularity Mean, CA, LT, Company Universal, Holiday Effect, Company Universal & Warner, Liveness, Instrumentalness, IT
Backward Elimination		Explicit Lyrics, Single Type, Genre Hip-Hop/Rap, Genre Latin/Reggaeton, Genre R&B, Genre Soundtrack, Company Independent & Sony, Company Universal, Company Universal & Sony, Company Universal & Warner, Company Warner, Peak After, Age, Group Gender Solo Female, Group Gender, Solo Other, AR, AU, CA, FR, GB, IT, LT, NL, PR, SE, TT, US, Key, Mode, Danceability, Energy, Instrumentalness, Liveness, Loudness, Valence, Tempo
Intersection of Previous Methods		AU, FR, Company Universal, Company Warner, Energy, Genre Latin/Reggaeton, Genre R&B, Genre Soundtrack, Group Gender Solo Female
Lasso Regularization for Lyrics	Lambda = 59.99	None

It is interesting to observe that many selected features do not belong to the audio subset. Only Energy was selected by each method. We therefore confirm that audio features are not the most important ones to explain the popularity of a track. Obviously, the label, the composition of the artist(s) involved in a given song or the genre are essential. However, this is completely relative to our dataset and two of our methods (Forward and Backward) are not suited to generalize results. Looking at the first method, which used a validation strategy, we notice the presence of genre and group composition information as well as the features computed using Google Trends and an indicator of artist popularity.

When it comes to the lyrics, no words are selected by our Lasso method. We believe that the vocabulary is too large compared to our sample size. We select manually some words: Like, Know, Love, Want, Wanna, Babi, Cant, Feel, Never, Need, Give, Nigga, Friend, Girl, Man, Hard, Life, Heart, Fuck, Shit, Bitch, Bad, Kiss, Good, Better, Lie. We use as features the count of each word without TF-IDF. The results are detailed later on.

4.2.2. Classification

Table 6: Selected features for classification

Model	Parameter(s)	Feature Selected
Lasso Regularization (no threshold)	Lambda = 2329	Energy, Peak After, Company Sony, Age, CA, Danceability, AU, Company Warner, Month, Speechiness, Single Type, Genre R&B, Group Gender Group Mixed, Instrumentalness, Company Independent & Universal, Genre Soundtrack, Group Gender Solo Other, Genre Latin/Reggaeton, Genre Dance, Peak Before, Loudness, IL, Valence, BR, PR, Duration, MX, Liveness, Name Length, Tempo, Group Gender Solo Female, CO, Group Gender Group Male, Peak At, Explicit Lyrics, Genre Hip-Hop/Rap, Key, Mode, Company Universal & Warner, Company Independent & Warner, GB, Followers Mean, Company Universal, US, Holiday Effect, GT Range Value, Company Independent & Sony, Genre Rock, IT, Artist Popularity Mean, Genre Pop, Group Gender Solo Male, Acousticness, GT Std Value
Random Forest	Number of estimators = 100 (by default)	Energy, Peak After, Age, Danceability, Speechiness, Month, Instrumentalness, Type Single, Group Gender Group Mixed, Peak Before, Name Length, Duration, Liveness, Valence, Loudness, Tempo, Peak At, Explicit Lyrics, Mode, Key, Company Independent & Warner, Followers Mean, Company Universal, Holiday Effect, GT Range Value, US, Company Independent & Sony, Artist Popularity Mean, Genre Rock, Group Gender Solo Male, Genre Pop, Acousticness, GT Std Value
Intersection of Previous Methods		Duration, Explicit Lyrics, Name Length, Holiday Effect, Type Single, Genre Pop, Genre rock, Month, Company Independent & Sony, Company Independent & Warner, Company Universal, Artist Popularity Mean, Followers Mean, GT Std Value, GT Range Value, Peak Before, Peak At, Peak After, Age, Group Gender Group Mixed, Group Gender Solo Male, US, Key, Mode, Acousticness, Danceability,

		Energy, Instrumentalness, Liveness, Loudness, Speechiness, Valence, Tempo
Lasso Regularization for Lyrics	Lambda = 50	40, Alcohol, Bathroom, Bein, Blowin, Bodi, Class, Come, Complic, Damag, Deep, Dive, Drop, Easi, Fashion, Favorit, Fear, Fine, Give, Grace, Hate, Head, Hear, Juic, Jumpin, Kinda, Left, Lesson, Love, Low, Move, Much, Nobodi, One, Pay, Poison, Pool, Put, Ride, Sun, Sad, Seat, Shawti, Shirt, Sinc, Sit, Sky, Slip, Sometim, Soul, Spirit, Straight, Stranger, Street, Sublimin, Suicid, Summer, Talk, Tatoo, Tell, Time, Today, Told, Travel, Uh, Weve, World

Compared to Regression, the whole audio feature subset was selected by both methods, in addition to each Google Trends features. Also, countries of origin do not seem to be important to classify our songs: only the US feature was selected by both methods. This is not surprising given the fact that most top artists are from the US, coming from another country clearly reduces the probability of getting a Hit. To summarize, selected features are: audio features, genre, label, artist popularity, group gender information and Google Trends features.

Overall, different features are selected regarding the task (regression or classification). It seems that audio features are much more important for classifying tracks while countries and other metadata are dominant for regression. We also notice that the features we created using Google Trends are often selected (for both tasks). It is hard to clearly understand what they measure but we can say that:

- GT Std Value: it is an indicator of the volatility of trends for an artist over a 3-month period. High volatility means more extreme values and stronger variations of the series. The fact that there are several steepest peaks of interest during the period could be explained by various marketing events and/or events that draw the attention on the artist more than usual. Therefore, this variable is quite useful, even if the artist is unknown or famous, as it takes into account the dispersion around the mean (unknown artist will have a lower mean).
- GT Range Value: this indicator does not make sense in terms of business for us but it is useful to analyze the previous feature. Indeed, if we have a high volatility, it tells us if it is due to very low trend values or several peak of interests as the maximum trend value is 100.
- GT Peak: this indicates when the interest for an artist was at its maximum over the 3-month period. If the peak is before, it could be explained by other tracks released or a significant promotion event. If the peak is at the release date of the track, it could be the sign that the artist is not that famous at the moment or that nothing was released in the

previous months or that it is a new artist. And a peak after is probably the sign that the artist was not drawing the attention that much before and that the track was not expected by a majority of people.

Finally, it is hard to tell which factors have the best predictive power between audio features, artist metadata and song metadata. It clearly depends on the method used to tackle the Hit Song Science problem but, in the end, a mix of the three improves results (see later on).

4.3. Models

4.3.1. Linear Regression

The results for each regression are available in the appendix: access to coefficients, p-value and other metrics.

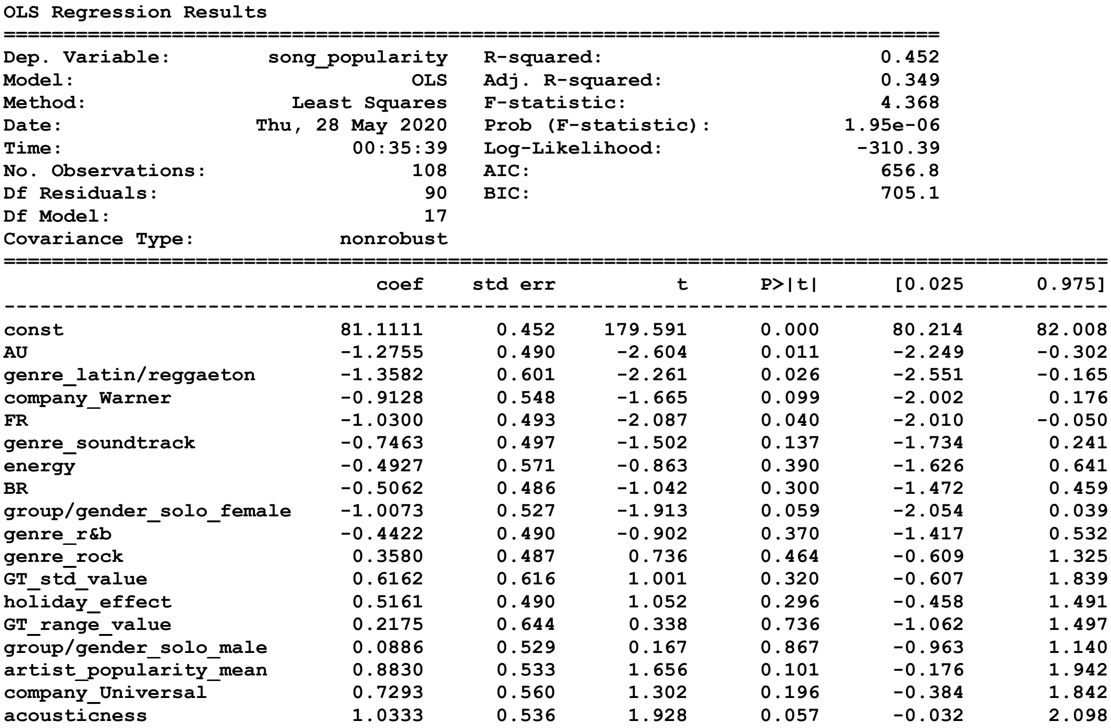
Table 7: Summary of linear regression models

Model Description	With Outliers	Hyper Param.	R2	Adj. R2	Fisher p-value	RMSE
OLS – song metadata	Yes		Train = 0.213	Train = 0.024	0.332	
OLS – song metadata	No		Train = 0.267	Train = 0.077	0.135	
OLS – artist metadata	Yes		Train = 0.181	Train = -0.098	0.905	
OLS – artist metadata	No		Train = 0.349	Train = 0.119	0.078	
OLS – artist metadata removing correlated features	No		Train = 0.333	Train = 0.118	0.070	
OLS – audio features	Yes		Train = 0.191	Train = 0.105	0.019	
OLS – audio features	No		Train = 0.160	Train = 0.064	0.094	
OLS – audio features removing correlated features	No		Train = 0.126	Train = 0.036	0.192	
Lasso – whole feature set	Yes	2.41				Test = 20.39
Lasso – whole feature set	No	0.45	Train = 0.374 Test = 0.205			Train = 4.57 Test = 5.78
OLS – lasso selected features	No		Train = 0.452 Test = 0.117	Train = 0.349	1.95e-06	Test = 6.09
Ridge – whole feature set	No	59.98	Train = 0.487 Test = 0.194			Train = 4.15 Test = 5.82
Ridge – lasso selected features	No	59.98	Train = 0.416 Test = 0.178			Train = 4.42 Test = 5.88

OLS – forward selection	No		Train = 0.580 Test = 0.073	Train = 0.458	4.84e-08	Test = 6.24
OLS – backward elimination	No		Train = 0.604 Test = -0.018	Train = 0.435	2.96e-06	Test = 6.54
Ridge – forward selection	No	17.47	Train = 0.530 Test = 0.208			Train = 3.97 Test = 5.77
Ridge – backward selection	No	8.81	Train = 0.560 Test = 0.176			Train = 3.84 Test = 5.88
Ridge Regression – lyrics	No	59.99	Train = 0.031 Test = -0.002			Train = 5.70 Test = 6.49
Lasso – whole feature set + personal selection of words	No	59.99	Train = 0.320 Test = 0.155			Train = 4.77 Test = 5.96
Dummy Model	No		Train = 0.0 Test = 0.0			Train = 5.79 Test = 6.48

Overall, regressions yield very poor results. While comparing the models to the dummy one, results are almost similar (recall that Song Popularity is a value between 0 and 100). We believe that this is mostly due to the selection of the tracks composing the dataset. For regressions, it is probably better to select examples to get a Song Popularity variable normally distributed around 50. We get our most “promising” results with a Lasso Regression: the results are similar to other models tried but Lasso is the one using the smallest feature set (curse of dimensionality). Also, lyrics does not help improving the results. Bellow we will provide the results for two models that we tried out: OLS with lasso selection subset (no regularization applied on coefficients) and OLS with forward selection subset.

Figure 14: OLS – lasso selection – no outliers



```
OLS Regression Results
==============================================================================
Dep. Variable:        song_popularity   R-squared:                       0.452
Model:                            OLS   Adj. R-squared:                  0.349
Method:                 Least Squares   F-statistic:                     4.368
Date:                Thu, 28 May 2020   Prob (F-statistic):           1.95e-06
Time:                        00:35:39   Log-Likelihood:                -310.39
No. Observations:                 108   AIC:                             656.8
Df Residuals:                      90   BIC:                             705.1
Df Model:                          17
Covariance Type:            nonrobust
==============================================================================================
                               coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------------
const                       81.1111      0.452    179.591      0.000      80.214      82.008
AU                          -1.2755      0.490     -2.604      0.011      -2.249      -0.302
genre_latin/reggaeton       -1.3582      0.601     -2.261      0.026      -2.551      -0.165
company_Warner              -0.9128      0.548     -1.665      0.099      -2.002       0.176
FR                          -1.0300      0.493     -2.087      0.040      -2.010      -0.050
genre_soundtrack            -0.7463      0.497     -1.502      0.137      -1.734       0.241
energy                      -0.4927      0.571     -0.863      0.390      -1.626       0.641
BR                          -0.5062      0.486     -1.042      0.300      -1.472       0.459
group/gender_solo_female    -1.0073      0.527     -1.913      0.059      -2.054       0.039
genre_r&b                   -0.4422      0.490     -0.902      0.370      -1.417       0.532
genre_rock                   0.3580      0.487      0.736      0.464      -0.609       1.325
GT_std_value                 0.6162      0.616      1.001      0.320      -0.607       1.839
holiday_effect               0.5161      0.490      1.052      0.296      -0.458       1.491
GT_range_value               0.2175      0.644      0.338      0.736      -1.062       1.497
group/gender_solo_male       0.0886      0.529      0.167      0.867      -0.963       1.140
artist_popularity_mean       0.8830      0.533      1.656      0.101      -0.176       1.942
company_Universal            0.7293      0.560      1.302      0.196      -0.384       1.842
acousticness                 1.0333      0.536      1.928      0.057      -0.032       2.098
```

Figure 15: OLS – forward selection – no outliers

```
OLS Regression Results
==============================================================================
Dep. Variable:        song_popularity   R-squared:                       0.580
Model:                            OLS   Adj. R-squared:                  0.458
Method:                 Least Squares   F-statistic:                     4.774
Date:                Thu, 28 May 2020   Prob (F-statistic):           4.84e-08
Time:                        00:49:45   Log-Likelihood:                -296.04
No. Observations:                 108   AIC:                             642.1
Df Residuals:                      83   BIC:                             709.1
Df Model:                          24
Covariance Type:            nonrobust
=============================================================================================
                                coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------
const                        81.1111      0.412    196.967      0.000      80.292      81.930
company_Warner               -0.9995      0.531     -1.882      0.063      -2.056       0.057
energy                       -3.1391      0.794     -3.953      0.000      -4.719      -1.559
AU                           -1.0618      0.479     -2.215      0.030      -2.015      -0.108
genre_latin/reggaeton        -3.1116      0.764     -4.075      0.000      -4.630      -1.593
acousticness                  0.4087      0.514      0.795      0.429      -0.614       1.431
loudness                      2.2260      0.736      3.026      0.003       0.763       3.689
FR                           -1.3292      0.495     -2.683      0.009      -2.315      -0.344
PR                            1.8598      0.710      2.620      0.010       0.448       3.272
genre_soundtrack             -1.1595      0.480     -2.413      0.018      -2.115      -0.204
genre_r&b                    -0.8652      0.485     -1.785      0.078      -1.829       0.099
group/gender_solo_female     -1.9525      0.513     -3.809      0.000      -2.972      -0.933
genre_hip-hop/rap            -1.3401      0.619     -2.165      0.033      -2.571      -0.109
peak_after                   -0.9220      0.467     -1.976      0.051      -1.850       0.006
genre_dance                  -0.8978      0.510     -1.759      0.082      -1.913       0.117
GT_range_value                0.8759      0.520      1.684      0.096      -0.159       1.911
ES                            0.8159      0.487      1.675      0.098      -0.153       1.785
artist_popularity_mean        0.8540      0.594      1.438      0.154      -0.327       2.035
CA                           -1.0217      0.508     -2.010      0.048      -2.033      -0.011
LT                            0.3255      0.235      1.386      0.170      -0.142       0.793
company_Universal             0.7993      0.546      1.465      0.147      -0.286       1.884
holiday_effect                0.6089      0.479      1.272      0.207      -0.343       1.561
company_Universal, Warner     0.8749      0.565      1.550      0.125      -0.248       1.998
liveness                     -0.6562      0.493     -1.330      0.187      -1.638       0.325
instrumentalness              0.4747      0.464      1.024      0.309      -0.447       1.397
IT                            0.3255      0.235      1.386      0.170      -0.142       0.793
```

From those figures we can see that even if the model is statistically significant (Fisher test), we still have “high” p-value for some features. However, the size of our dataset needs to be taken into account.

4.3.2. Classification

Table 8: Summary of best scores

Model	Supp. 1/0	Precision Class 1	Recall Class 1	F1 Test Class 1
Logistic Regression	8 / 28	0.47	0.88	0.61
K Nearest Neighbours	8 / 26	0.67	0.50	0.57
Random Forest	8 / 28	0.50	0.12	0.20
Support Vector Machines	8 / 28	0.57	0.50	0.53
Multilayer Perceptron	8 / 28	0.58	0.88	0.70

Logistic Regression

Table 9: Logistic regression results

Model Description	Hyper param.	Supp. 1/0	Precision Class 1	Recall Class 1	F1 Test Class 1	F1 Train Class 1
LR – song metadata		8 / 28	0.00	0.00	0.00	0.22
LR – artist metadata		8 / 28	0.75	0.38	0.50	0.39
LR – audio features		8 / 28	0.33	0.12	0.18	0.28
LR – song metadata + artist metadata		8 / 28	0.50	0.38	0.43	0.75
LR – song metadata + audio features		8 / 28	0.14	0.12	0.13	0.59
LR – artist metadata + audio features		8 / 28	0.43	0.38	0.40	0.61
LR – whole feature set		8 / 28	0.29	0.25	0.27	0.84
LR – whole feature set - with threshold		8 / 26	0.20	0.12	0.15	0.95
Ridge LR – lasso selected features	0.0045	8 / 28	0.47	0.88	0.61	0.62
Ridge LR – whole dataset	0.007	8 / 28	0.47	0.88	0.61	0.64
Ridge LR – random forest selected features	0.01	8 / 28	0.60	0.75	0.67	0.59
Lasso LR – features selected by both methods + addition of lasso selected words	0.0007	8 / 28	0.67	0.50	0.57	0.95
Dummy classifier		8 / 28	0.21	0.38	0.27	

The results with logistic regression are promising. While testing various subsets, it appears that the addition of artist metadata increases the precision of the model and not in the expense of the recall, meaning that those features are increasing the odds that when the model predicts that a track is a Hit, there is a better chance that it is actually a Hit. The Ridge Regularization with the use of features selected by Lasso or Random Forest gives us our best result. We tend to prefer the model with Lasso features because the result on the test set with features selected by Random Forest is probably due to random chance (small test set – better result than on training set). The addition of lyrics is promising but the model needs further regularization because it is overfitting, but the C parameter is already very low.

K Nearest Neighbours

Table 10: K Nearest Neighbours results

Model Description	Hyper param.	Supp. 1/0	Precision Class 1	Recall Class 1	F1 Test Class 1
KNN – whole feature set – no threshold	Neighbours=4 Weight=distance	8 / 28	0.43	0.38	0.40
KNN – whole feature set – with threshold	Neighbours=4 Weight=distance	8 / 26	0.67	0.50	0.57
KNN – features selected by both methods – no threshold	Neighbours=4 Weight=distance	8 / 28	0.50	0.12	0.20
KNN – features selected by both methods – with threshold	Neighbours=4 Weight=distance	8 / 26	0.50	0.25	0.33
KNN – tuned – whole feature set – with threshold	Neighbours=3 Weight=uniform Distance=euclidean	8 / 26	0.50	0.50	0.50
Dummy classifier		8 / 28	0.21	0.38	0.27

The K Nearest Neighbours algorithm provides better results than a random classifier. The benefit of this model is that it is easy to implement and to understand. However, it is not performing well when it comes to generalizing. It would be worth testing it on a larger dataset with a larger test set.

Random Forest

Table 11: Random Forest results

Model Description	Hyper param.	Supp. 1/0	Precision Class 1	Recall Class 1	F1 Test Class 1
RF – whole feature set – no threshold	Trees=100	8 / 28	0.50	0.12	0.20
RF – selected features – no threshold	Trees=100	8 / 28	0.33	0.12	0.18
RF – tuned – selected features – no threshold	Trees=100 Bootstrap=False Class weight=None Max depth=10 Max features=auto Max samples=None Min samples leaf=1 Min samples split=5	8 / 28	0.33	0.12	0.18
RF – same parameters – selected features + words		8 / 28	0.25	0.12	0.17
Dummy classifier		8 / 28	0.21	0.38	0.27

Our results with random forest are the worst. The Dummy classifier is even better.

Support Vector Machines

Table 12: Support Vector Machine results

Model Description	Hyperparameters	Supp. 1/0	Precision Class 1	Recall Class 1	F1 Test Class 1	F1 Train Class 1
SVM – linear kernel – whole feature set – no threshold	C=1	8 / 28	0.30	0.38	0.33	1.00
SVM – linear kernel – whole feature set – with threshold	C=1	8 / 26	0.40	0.50	0.44	0.98
SVM – gaussian kernel – whole feature set – no threshold	C=1 Gamma=auto	8 / 28	0.00	0.00	0.00	0.37
SVM – gaussian kernel – whole feature set – with threshold	C=1 Gamma=auto	8 / 26	0.00	0.00	0.00	0.37
SVM – linear kernel – lasso selected – no threshold	C=1	8 / 28	0.14	0.12	0.13	0.84
SVM – gaussian kernel – lasso selected – no threshold	C=1 Gamma=auto	8 / 28	0.00	0.00	0.00	0.48
SVM – linear kernel – random forest selected – no threshold	C=1	8 / 28	0.57	0.50	0.53	0.56
SVM – tuned – lasso selected – no threshold	C=1000 Class weight=balanced Gamma=0.0001 Kernel=rbf (Gaussian)	8 / 28	0.32	0.75	0.44	Mean score best model during validation = 0.53
SVM – tuned – random forest selected – no threshold	C=100 Class weight=balanced Gamma=0.0001 Kernel=rbf	8 / 28	0.32	0.75	0.44	Mean score best model during validation = 0.43
SVM – improved – lasso selected – no threshold	C=700 Class weight=balanced Gamma=0.00001 Kernel=rbf	8 / 28	0.35	0.88	0.50	Mean score best model during validation = 0.49
SVM – tuned – lasso selected + words	C=10 Class weight=None Gamma=0.001 Kernel=rbf	8 / 28	0.38	0.38	0.38	Mean score best model during validation = 0.86
Dummy classifier		8 / 28	0.21	0.38	0.27	

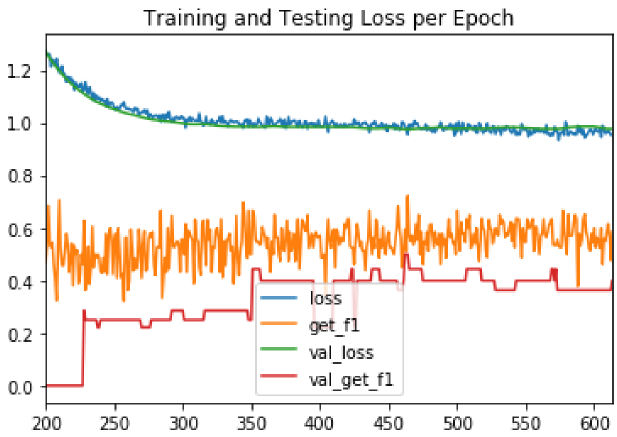
Results are average and we believed that Support Vector Machines would have yielded better F1 scores than Logistic Regression. We noticed during the hyperparameter tuning for the model with features selected by Lasso Regularization and combined with lyrics, that some extremely good results were achieved on the validation sets. The average F1 score (over 5 cross-validation sets) for our best combination of parameters was 0.86. This could be explained by an undersized validation set. But it means that Support Vector Machines can yield very good results. Support Vector Machines perform well on high dimensional dataset and they can produce highly complex decision boundaries in comparison with other models that expect data to be linearly separable. The Gaussian kernel was always chosen during cross-validation meaning that our data probably have non-linear relationships.

Multilayer Perceptron

Table 13: Multilayer Perceptron results

Model Description	**Parameters**	**Supp. 1/0**	**Precision Class 1**	**Recall Class 1**	**F1 Test Class 1**
One hidden layer – 40 units – Ridge Regularization – random forest selected	Activation fct.=ReLU Activation output=sigmoid Kernel Initializer=He uniform Kernel regularizer=3.5 Bias=yes Dropout=0.35 Optimizer=Adam Learning rate=0.0009 Class weight={0:1, 1:3.45} Batchsize=200 (exclude cases where a batch would not contain Hit examples)	8 / 28	0.58	0.88	0.70
Results on validation set		4/18	0.50	0.50	0.50

Figure 14: Learning curves



This is our best model so far. During the process of testing various set of parameters for the model, we noticed that the use of class weight clearly improved the results. Before that, our cost function (binary cross entropy) was converging towards a lower value (around 0.5) but this was mainly caused by a strong penalization of Hit examples (i.e. the model chose to concentrate on class 0 to reduce even more the value returned by the cost function). We also used a smaller learning rate to start with.

4.3.3. Conclusion

From the numerous tests that we tried out, given our dataset, the models performed better than a dummy one, except for Random Forest. We believe that Machin Learning can indeed be used by the industry to help A&R teams determining potential future Hits. According to our work, the most promising algorithms are the Multilayer Perceptron and the Logistic Regression. Our results are similar to the ones of (Herremans, Sörensen, & Martens, 2014) who used a similar method to define popularity. They had better results with Logistic Regression and poor ones with Random Forest.

5. Discussions

In this part, we would like to present various discussions regarding our dataset, the definition of popularity and our results. Here, we analyse the limits of our work and expose some ideas that we have.

5.1. Dataset discussion

The construction of a database for the Hit Song Science research field is something quite challenging. When we first started, we thought that tackling the subject using only Hits (Top 2018 and Top 2019 playlists) was better than using Hits and random songs for Non-Hits (true Non-Hits mixed with tracks that are close to being part of a chart). However, we realized during the Regression task that this was probably not a good method either because our dataset was then only composed of tracks with similar values in Popularity. Indeed, being part of a Spotify's Top is already a huge achievement for an artist. Therefore, we believe that a good database should be composed of a significant number of tracks with a normally distributed song popularity. To obtain such a database, methods like undersampling or oversampling could be used in addition to Spotify's API search point with a list of common genres of music.

Also, some of our features are evolving over time (Artist Popularity, Followers, Age and Song Popularity). Except for Song Popularity, we included a bias in our analysis with the others because we could not retrieve their value before the release of the track. Indeed, the historical values are unavailable on Spotify. However, for the Song Popularity, it is interesting to get the present value for each song because then it includes a kind of temporal dimension, which is recommended by (Perrie, 2019).

Additionally, our dataset was probably too small to get consistent results. We did not think about it when we built it because it was our first Machine Learning project. However, compared to previous works, our database is the smallest one. We could have included more Top playlists but then we would have encountered a second issue: old Hits in our database. As a matter of fact, music tastes are constantly evolving, and a database should therefore be constituted of a majority of recent tracks. Another solution could be to discount older hits to account for time evolution. The small size of our dataset led to several troubles while training and analysing our models. For instance, the hyperparameters tuned on cross-validated sets could be very different when running the algorithm several times: the first time a C value of 10 would be chosen and the second time, it would return a C value of 1000. We tried to obtain a larger dataset but the other sources we used are much less performant than Spotify's. Taking the example of Google Trends, an error 429 was returned several times for a list of 150 tracks only. We would need much more time to retrieve the same features for more songs.

5.2. Song Popularity discussion

We chose to test the two main methods (continuous and binary) used in the Hit Song Science research field to define the popularity of a track. Regarding the continuous variable we used, there is a debate because it is based on the listening counts of the track. Therefore, a song could reach a high value of popularity if it is streamed many times by a small amount of people. In general, should we consider a song to be popular because it reaches people globally or because it is extremely famous but only in some countries. We tend to opt for the second option, as evidenced by (Fan & Casey, 2013), also because in today's society, artists earn money if a song is streamed a lot. Listening counts are a good solution to account for that second option. Regarding the binary approach of popularity, we operated similarly to previous works by differentiating the top of a chart from the rest of the chart. This approach tends to work better in terms of numerical results, but it cannot be used with a business perspective. Indeed, being part of a chart, regardless of the position, is already a main accomplishment. Therefore, we believe that it would be interesting on a business perspective to tackle the problem using a dataset with the Song Popularity feature being normally distributed (mean approx. 50). Several classes (0-25, 25-50, 50-75, 75-100) could then be defined to include smoothness in the definition and avoid the problem of the songs at position 101 on a top 100 chart. In terms of business perspective, it is more useful to have a rough idea of where the track would be positioned on a popularity scale rather than having a two-fold response with a significant error rate.

5.3. Feature discussion

When looking at the features that we selected, it is interesting to see that we got very few audio features during the regression task while they were all selected for the classification. However, for every model we tried on individual subsets (song metadata, artist metadata and audio features), the results on audio features were often very poor. We therefore confirm that the success of a track is not only explained by audio features but mostly by external factors. It seems that metadata features are necessary for our project. It makes sense given the fact that a track is always associated to one or several artists who play a main role in making a track popular as they are its physical representatives. The popularity of a track is explained by a mix of many things among which we have the figure of the artist and its perception by others, the basic song metadata (the environment in which the record label is evolving or the release period for instance) and the audio characteristics.

We found it hard to analyse the coefficients of our model given the features that were selected. For instance, during the regression task, we noticed that the Lasso method did not select many country features compared to the other two methods, which is probably explained by the fact that it was the only one using cross-validation and selecting features that could help a model to generalize. Moreover, we believe that a magic recipe for Hit songs cannot be found because a track needs something new compared to what has been done previously to become popular (Salganik, Dodds, & Watts, 2006). We also found several “inconsistent” results with our coefficients. With linear regression using song metadata, every genre coefficient was negative (meaning that the last genre which is not included in the model – Country – would be a “popular” genre). With Logistic Regression, Peak At, Peak After and Peak Before had negative coefficients. This can be explained by multicollinearity, in the first case, or the use of Regularization methods that can invert the signs of the coefficients. We can still analyse the signs of the coefficients that are not close to 0. We observe for Logistic Regression that GT Std Value increases the probability of having a Hit while the Energy audio feature decreases it (these observations are probably consistent because the associated coefficients are clearly different from 0). Among the drivers of popularity (positive coefficients) we also have the acousticness of a song, the fact that there is only one male artist associated, that the song genre is Pop or that the artists are popular. On the contrary, it seems that a Google Trends peak after the release date of the track decreases the odds of being a Hit, which supports the idea that a strong marketing campaign before the release is necessary. Also, older artists are less likely to be associated with a popular track. Those results are consistent with what we found for the regression using features selected by Lasso, without applying a regularization, even if some of the associated p-values are higher than our risk level alpha (0.05).

5.4. Result discussion

For this project, we wanted to test the addition of metadata to audio features and observe if we could improve the performance of the models that were used in previous researches. We could not find the datasets that were previously used (to have the exact same tracks) and in any case we noticed while working on our dataset that we would have needed more time to extract our features for a larger list of songs. Relatively to our dataset, the addition of metadata clearly improved our results and we managed to get interesting scores for the Logistic Regression and the Multilayer Perceptron. For another project, we would like to try out those models on a larger dataset.

In terms of business perspectives, several features could not be used in real time. For example, the Google Trends' would only be accessible after the release of the track. As monitoring web activities prior to the release of a song returns interesting results (Koenigstein, Shavitt, & Zilberman, 2009), it could be probably more interesting for record companies to use Machine Learning techniques to monitor the potential evolution of a track in the charts (Bhattacharjee, Gopal, Lertwachara, Marsden, & Telang, 2007). Regarding the performance of our models, it seems easier to increase recall at the expense of precision. Our best F1 scores (Logistic Regression and Multilayer Perceptron) were achieved with a recall of 0.88 and precision of 0.47 and 0.58 respectively. It means that predicted Hits include almost all the true Hits and almost the same amount of Non-Hits. For a record company, the ratio of success is between 1:4 and 1:10 (IFPI, 2016), meaning that 10%-25% of the supported artists will become very famous. Let us pretend that a model could be generalized and achieve similar results to ours with features that can be used by record companies. If the company decided to invest only in the artist/track returned by the model, the ratio would become approximately 1:2, (precision of 0.47, 0.58) but the company would not miss any opportunities (rejecting an artist that could have become famous – which is not included in the 1:4, 1:10 ratios). It would be interesting to ask music professionals what the best trade-off for them would be. We tend to think that a higher recall (without a precision of 0.00) would be better but we do not know if record companies often miss opportunities by rejecting artists.

When it comes to the algorithms or methods used, we think that the classification method yields better results, even if it cannot be compared to regression. However, the classifiers we trained, had significantly better results than a dummy model contrary to our regressions.

Conclusion

Our objective was to test the combination of metadata with audio features to improve the results. We decided to focus on metadata that have never been used in previous works: Google Trends, artist age, genres of music or labels for instance. We conclude that indeed, the audio features are not sufficient to explain the popularity of a track and that metadata are essential. The selection methods we used gave us a set of influential features for both kinds of tasks, with a majority of metadata (even if we mostly had song/artist metadata in our dataset). It was interesting to see that Google Trends features exerted a high influence on the outcome of our classifiers, increasing the probability for a track to be a Hit. We also found the classification task more promising, due to the comparison of our models with a dummy classifier and a dummy regressor. The use of a Neural Network yielded our best result with a F1 score of 0.70 on our test set and we would recommend trying to use it on a larger dataset composed of the features we selected by Random Forest.

In terms of business perspective, what we have tried would not suit given the fact that some features are available only at the release date of the album. Trend evolution for an artist is an interesting factor to monitor but that should be used differently compared to what we have done, to account for available information prior to the release date or the recruiting of artist.

This project was very interesting, and we learnt a lot as it was our first application of the content we had worked on with online courses since the beginning of the year. We discovered many other resources and books to deepen our understanding of Data Science and we look forward to improving ourselves and evolving in the field of data.

We would like to end this project by presenting some ideas that could be tried for further work on the Hit Song Science research field:

- Try to constitute a dataset per genre and to train models on each dataset. (Herremans, Sörensen, & Martens, 2014) worked on dance songs and found that models could performed better on a specific genre.
- Build a large dataset, thinking about the distribution of Spotify’s popularity value. There is now an offset limit with Spotify’s API for the search point so we would

recommend using lists of Spotify's track ID that can be found on Kaggle for instance, paying attention to the release date of the tracks.

- Use detailed audio features which are available on Spotify's API. The ones we used summarize that information but probably simplify it.
- Use the market feature to create a feature indicating the number of countries where the song is available. Maybe we could interpret various strategies for launching an artist: is it better to focus on some markets or to promote it worldwide (use historical data)?
- Many tags can be obtained from LastFM's API, which can be useful to understand how the song is perceived by the listeners.
- Tackle the Hit Song Science problem as a multiclassification task to smooth the definition of popularity.
- Use a combination of Spotify's followers, LastFM's subscribers, Deezer's subscribers, Genius's page views or Instagram's followers to analyse the popularity of an artist.

Bibliography

Adler, M. (1985, February). Stardom and Talent. *American Economic Review, 75*(1).

Askin, N., & Mauskapf, M. (2017, April). What Makes Popular Culture Popular?: Product Features and Optimal Differentiation in Music. *American Sociological Review, 82*(5), 910-944.

Bhattacharjee, S., Gopal, R. D., Lertwachara, K., Marsden, J. R., & Telang, R. (2007, September). The Effect of Digital Sharing Technologies on Music Markets: A Survival Analysis of Albums on Ranking Charts. *Management Science, 53*, 1359-1374.

Borg, N., & George, H. (2011). What makes for a hit pop song? What makes for a pop song? *Stanford University*.

Chon, S. H., Berger, J., & Slaney, M. (2006, October). Predicting success from music sales data: a statistical and adaptive approach. *1st ACM workshop on Audio and music computing multimedia*, (pp. 83-87). New York.

Ciocca, S. (2017, October 10). *How Does Spotify Know You So Well?* Retrieved from Medium: https://medium.com/s/story/spotifys-discover-weekly-how-machine-learning-finds-your-new-music-19a41ab76efe

Dhanaraj, R., & Logan, B. (2005). Automatic Prediction of Hit Songs. *ISMIR*, (pp. 488-491). London.

Downie, J. (2003). Music information retrieval. *Annual review of information science and technology, 37*(1), 295-340.

Fan, J., & Casey, M. (2013, October). Study of Chinese and UK Hit Songs Prediction. *10th International Symposium on Computer Music Multidisciplinary Research*, (pp. 640-652). Marseilles.

Garcia, A., Kala, C., & Barajas, G. (2017). *Rage Against the Machine Learning: Predicting Song Popularity.* Retrieved from Stanford University: http://cs229.stanford.edu/proj2017/final-reports/5231342.pdf

Georgieva, E., Suta, M., & Burton, N. (2018). *Hitpredict: Predicting Hit Songs Using Spotify Data.* Retrieved from Stanford University: http://cs229.stanford.edu/proj2018/report/16.pdf

Goldman Sachs. (2018, September). *The Music in Air*. Retrieved from https://www.goldmansachs.com/insights/pages/infographics/music-streaming/

Hann, I.-H., Oh, J., & James, G. (2011). *Forecasting the Sales of Music Albums: A Functional Data Analysis of Demand and Supply Side P2P Data.* Working paper.

Herremans, D., Sörensen, K., & Martens, D. (2014, July). Dance hit song prediction. *Journal of New Music Research, 43*, 291-302.

IFPI. (2016). *Investing In Music - The Value of Record Companies.* IFPI.

IFPI. (2019). *Global Music Report 2019.*

IFPI. (2019). *Record companies: Powering the music ecosystem.* Retrieved from IFPI: https://powering-the-music-ecosystem.ifpi.org/download/Powering_the_Music_Ecosystem_poster.pdf

Isogawa, M., Masling, S., & Pan, M. (2019, July). *Predicting the Popularity of Song.* Retrieved from Maikaisogawa - CS_221: http://www.maikaisogawa.com/wp-content/uploads/2019/07/CS_221_Predicting_the_Popularity_of_Top_100_Billboard_Songs.pdf

Kaminskas, M., & Ricci, F. (2012, April). Contextual music information retrieval and recommendation: State of the art and challenges. *Computer Science Review, 6*(2-3), 89-119.

Koenigstein, N., Shavitt, Y., & Zilberman, N. (2009). Predicting Billboard Success Using Data-Mining in P2P Networks. *ISM*, (pp. 465-470). San Diego.

Laurier, C., Grivolla, J., & Herrera, P. (2008). Multimodal Music Mood Classification using Audio and Lyrics. *Machine Learning and Applications, 2008* (pp. 688-693). IEEE.

MIDiA. (2019, March 19). *MIDiA Research 2018–2026 Streaming Music Forecasts*. Retrieved from Music Industry Blog: https://musicindustryblog.wordpress.com/2019/03/19/midia-research-2018-2026-streaming-music-forecasts/

Montecchio, N., Roy, P., & Pachet, F. (2019, March). *The Skipping Behavior of Users of Music Streaming Services and its Relation to Musical Structure.* Retrieved from arXiv: https://arxiv.org/abs/1903.06008

Ni, Y., Santo-Rodríguez, R., Mcvicar, M., & De Bie, T. (2011). Hit song science once again a science? *NIPS 2011.* Sierra Nevada.

Nielsen. (2017, February 11). *Time With Tunes: How Technology Is Driving Music Consumption*. Retrieved from Nielsen: https://www.nielsen.com/us/en/insights/article/2017/time-with-tunes-how-technology-is-driving-music-consumption/

Pachet, F., & Roy, P. (2008). Hit song Science Is Not Yet a Science. *ISMIR.* Philadelphia.

Perrie, J. (2019, August). *Modelling Chart Trajectories using Song Features.* Retrieved from University of Waterloo Library: http://hdl.handle.net/10012/14937

Pham, J., Kyauk, E., & Park, E. (2016). Predicting Song Popularity. *Stanford University*.

Reiman, M., & Örnell, P. (2018, May). *Predicting Hit Songs with Machine Learning.* Retrieved from Semantic Scholar: https://pdfs.semanticscholar.org/e6cc/edb50d2c2b01bca108cb090943e86fb58135.pdf

Russell, J. A. (1980, December). A Circumplex Model of Affect. *Journal of Personality and Social Psychology, 39*(6), 1161-1178.

Salganik, M. J., Dodds, P. S., & Watts, D. J. (2006, February 10). Experimental study of inequality and unpredictability in an artificial cultural market. *Science, 311*, 854-856.

Samuel, A. L. (1959, July). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development, 3*(3), 210-229.

Silber, J. (2019, April). *Music Recommendation Alwgorithms: Discovering Weekly or Discovering Weakly?* Retrieved from MediArXiv: https://mediarxiv.org/6nqyf/download

Soundcharts Blog. (2019, June 14). *How Music Streaming Works and The Popular Music Streaming Trends of Today*. Retrieved from https://soundcharts.com/blog/how-music-streaming-works-trends#5-music-streaming-trends-shaping-the-industry

Statista. (2019). *Digital Media Report 2019 - Digital Music.* Statista.

Tzanetakis, G., & Cook, P. R. (2002, July). Musical Genre Classification of Audio Signals. *IEEE Transactions on Speech and Audio Processing, 10*(5), 293-302.

Yang, L.-C., Chou, S.-Y., Liu, J.-Y., Yang, Y.-H., & Chen, Y.-A. (2017, April). Revisiting the Problem of Audio-Based Hit Song Prediction Using Convolutional Neural Networks. *IEEE International Conference Acoustics, Speech and Signal Processing* (pp. 621-625). Taiwan: IEEE.

Appendices

Project is available on GitHub here.