(Re)Discovering My Music Through Analytics

Jean Parenty

December 2020

Introduction

As an avid music listener, I'm always on the look for new artists, songs, and just new content to listen to in general. For that purpose I use Spotify and I'm always really impressed by the quality of its recommendations. Spotify provides numerous features for its users to have an endless flow of suggestions at their fingertips. For example, each time I'm impressed by a track, I use the radio features to listen to playlists full of similar songs, that I may already know (already liked) or not. When I don't have the inspiration to listen to particular songs, I simply listen to the custom daily mixes playlists that mix together similar gender artists that I may again already know or not. With this method, listening to new music is an endless process and I'm grateful to Spotify for that. It's clear that I'm not in need of any song recommendation model and I would not have the pretension to do it better than Spotify's algorithms. However, there's one thing Spotify doesn't do for me, it doesn't tell me much about the characteristics of the music I like. For example, despite all its superb features, I'm always surprised by how difficult it is to know the music gender of any artist. This is information that Spotify clearly has and uses for its algorithms, but who knows why, a simple 'gender' tag on the artist page would have work but no, it's not here.

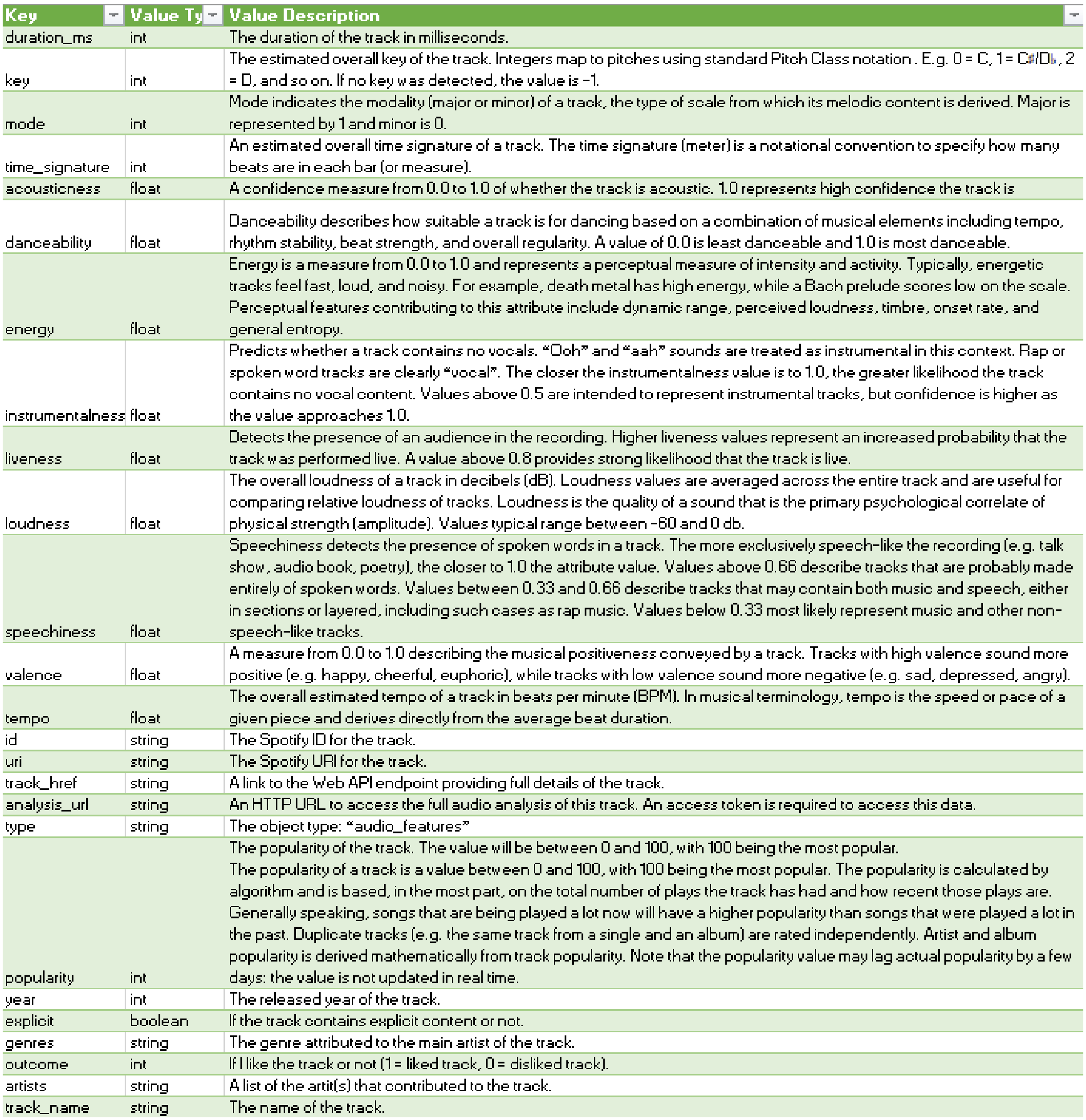
In this project, not only I will be able to analyze the distribution of genres of my musical taste, I want to focus on the characteristics of the music I love and listen to every day, and why is it that I listen to certain artists and not others. I will try to build a recommendation model that can predict whether I will like or dislike certain random tracks. For that purpose, I will be using two datasets built from the data available from my Spotify account using the Spotify API. One of the datasets, 'likedSongs' includes 1112 songs, while the other dataset, 'dislikedSongs' includes 800 songs. With those 2 datasets, we will explore the influence of the multiple variables describing a track. More importantly than the prediction itself, with this model, my goal is to learn more about the music I like and how it differs from the one I don't.

Let's now dive together into my musical world and make it reveal as much as possible about me, about my tastes, and what makes them the way they are.

Data Preparation: How I built my dataset

For this project, I trained and tested models using a quite unique and personal dataset, a collection of 1110 liked songs and 800 disliked songs from my Spotify account. As such a dataset doesn't exist yet, I had to make it myself. Thankfully, Spotify makes accessible all its data by the use of its API. I currently have around 3000 tracks as liked songs on my Spotify account, with around 500 of them scattered in multiple playlists. I consider the songs in my playlist as a 'best of' of my liked songs, so I included all those songs in my 'liked songs 401' dataset for sure. I also want the model to focus on songs I like presently (more or less from the last 2 years) and not consider all the music I was listening to at the beginning of my Spotify experience (there's obviously some music I love from this period, those are in my playlists). With that in mind, I'm going to add my 500 most recent liked songs (removing duplicate) to the dataset, in addition to the playlist's songs, in order to reach a substantial amount of tracks for my model. I also need the model to recognize songs I dislike. For that purpose, I created a playlist 'disliked songs 401'. Looking for songs you don't like is actually not the easiest task especially when you want them to be diversified, gathering 1000 trap or dubstep songs would have been easy but not efficient. To inspire me I started by looking at the chart top 50 Spotify's playlist. Considering my process to gather dislike songs, there could be some bias in the variable 'popularity'. It's always easier to find the most popular songs first. I was also pleased to see I actually had friends willing to give me a hand, recommending me artists I would probably not like (this end up being a huge help). Once the playlists were finalized, I used python and 'spotipy' library (a library that uses Spotify API) to extract all my songs with all the features variable associated with each track. To this features variable, I also added the track name, the artists' names, if it has explicit content, as well as the genre of the main artist **(refer to spotifydataV3.py file for full code)**. I added an outcome column and assigned 1 for liked songs and 0 for disliked songs before merging them into my final dataset.

Data Description



Key	Value Ty	Value Description
duration_ms	int	The duration of the track in milliseconds.
key	int	The estimated overall key of the track. Integers map to pitches using standard Pitch Class notation . E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on. If no key was detected, the value is -1.
mode	int	Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
time_signature	int	An estimated overall time signature of a track. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).
acousticness	float	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is
danceability	float	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
energy	float	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
instrumentalness	float	Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.
liveness	float	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
loudness	float	The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.
speechiness	float	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.
valence	float	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).
tempo	float	The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.
id	string	The Spotify ID for the track.
uri	string	The Spotify URI for the track.
track_href	string	A link to the Web API endpoint providing full details of the track.
analysis_url	string	An HTTP URL to access the full audio analysis of this track. An access token is required to access this data.
type	string	The object type: "audio_features"
popularity	int	The popularity of the track. The value will be between 0 and 100, with 100 being the most popular. The popularity of a track is a value between 0 and 100, with 100 being the most popular. The popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are. Generally speaking, songs that are being played a lot now will have a higher popularity than songs that were played a lot in the past. Duplicate tracks (e.g. the same track from a single and an album) are rated independently. Artist and album popularity is derived mathematically from track popularity. Note that the popularity value may lag actual popularity by a few days: the value is not updated in real time.
year	int	The released year of the track.
explicit	boolean	If the track contains explicit content or not.
genres	string	The genre attributed to the main artist of the track.
outcome	int	If I like the track or not (1 = liked track, 0 = disliked track).
artists	string	A list of the artit(s) that contributed to the track.
track_name	string	The name of the track.

Figure 1: Data-set's feature description

Now that we have our hand on the final dataset, let's take a look at all the variables available to us (Figure 1). We have 25 variables associated with each track. Some of them are going to be useless in our prediction and we can already drop them. With that in mind, we drop the 'type' variable as it has unique values among all songs, as well as the track's 'id', 'uri', 'track_href' and 'analysis_url', as those are variables we got from the API call to identify each track.

Now we can start taking a look at the distribution of each variable from the liked songs dataset and the disliked songs dataset. We start with the variable 'genres' in order to relieve my frustration from not being able to access this information directly on the Spotify app. My liked songs dataset contains 195 genres while disliked songs dataset contains 151 genres, with the following genre's plot in Figure 2 as my top 25 genres.

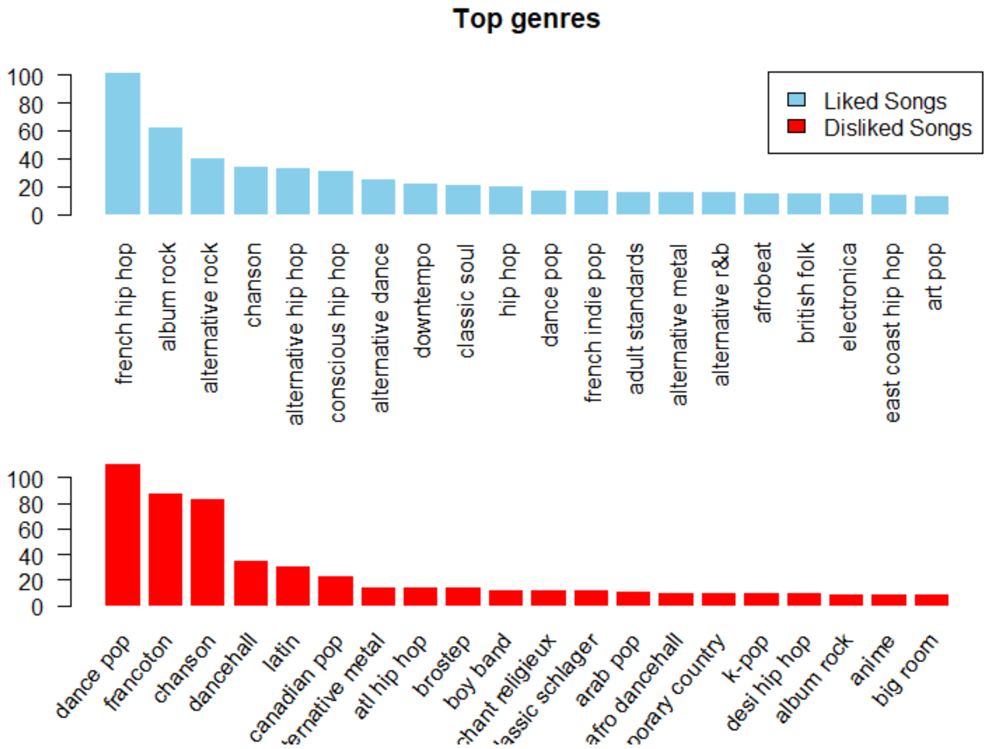


Figure 2: Top 25 genre for like and dislike

Unsurprisingly, rap, rock, and french variety (identified as 'chanson') dominate the ranking, followed by electro and soul. Among the disliked songs, pop, francoton and french variety dominate the ranking. The first "surprising" result appears as french variety is present in the top 5 genres of both liked and disliked songs. When thinking about it, it's not that surprising as french variety is such a huge genre that it is full of good songs as well as bad songs. The distribution of artists follows the same flow as genres.

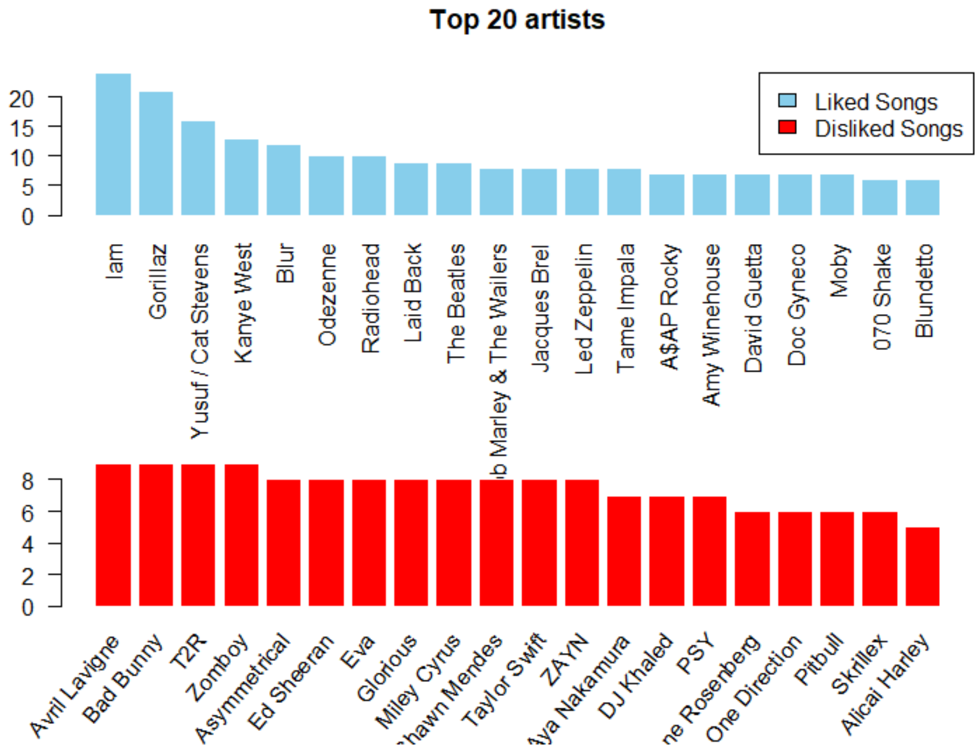


Figure 3: Top Artists

As we can see from Figure 3, liked artists are dominated by french hip hop (IAM is a classic in the genre), alternative rock (Blur, Radiohead), and french variety (Jacques Brel). On the other hand, disliked artists are dominated by pop or Latin artists (Avril Lavigne, Bad Bunny). Just from those 2 plots, we already have a good idea of how my liked music differs from my disliked music. The disliked songs genres are much more 'explosive' and festive contemporary genres. Let's now focus on what the quantitative variable distribution has to say about that.

Let's start with variables that follow a kind of similar distribution. With Figure 4, we analyze the distribution of acousticness, speachiness, tempo, and valence by superposing histogram from disliked and liked songs dataset. Since the liked songs have much more observation (310 to be exact), it's not surprising to have a higher frequency than disliked songs. What we should focus on, is the distribution of the data on the x-axis. For example, the valence distribution shows interesting results as the deviation of the liked songs from the mean is much higher compared to disliked songs deviation. This means I have a lot of songs that have either a

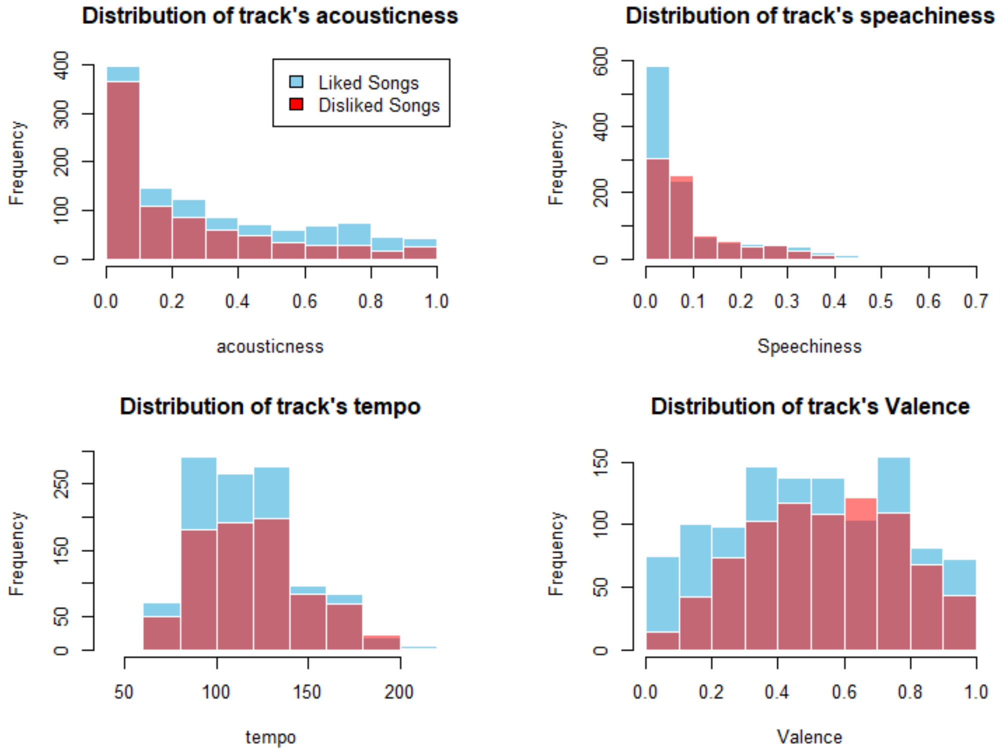


Figure 4

positive mood (high valence) or a negative mood (low valence) while disliked songs' valence is mostly average. The track speechiness tells us that none of the songs in the dataset are speech, as a value under 0.33 means the track is a song with high confidence. Since I'm only analyzing songs, this variable is probably not very significant. Finally, 'acousticness' trend from the liked songs is subtly different from disliked songs, as there are proportionally more liked songs as the confidence of the track being acoustic increases compared to disliked songs.

Let's now give our attention to variables that have a significantly different distribution for liked and disliked songs (Figure 5). We can see that danceability, energy, and popularity of liked songs are all shifted toward the lower bound of the x-axis compared to the distribution of disliked songs. This tells us quite interesting information, my liked songs tend to be less danceable, less energetic, and less popular than the songs I dislike. On the other side, my liked songs tend to be much more instrumental than the disliked songs, as not even one observation appears to be an instrumental track. The loudness of the liked songs is also significantly higher compared to disliked songs. Finally, the songs I like are also older in general.

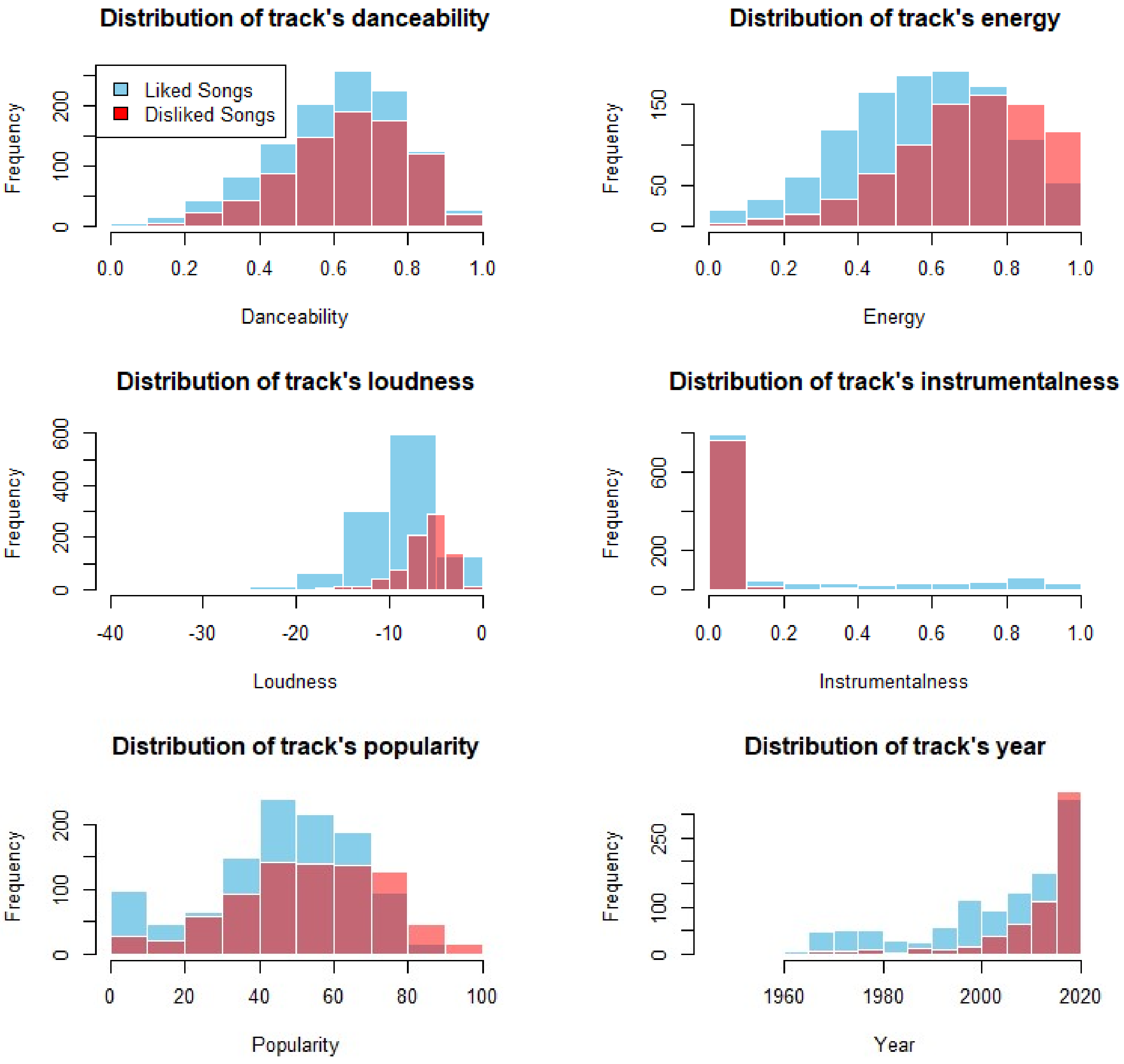


Figure 5

We also observe less relevant distributions as shown in Figure 6. The proportion of explicit content in my disliked songs is higher than in my liked songs (my mother would probably be proud), but I don't think the difference and the impact of this variable is big enough to make a difference in terms of prediction. It's also a factor that I rarely consider for my music taste. The time signature has just a few values (4) and they seem to be quite similar among liked and disliked songs, so again this variable will probably be useless (like mode distribution).

Still, all of our findings go along with what we observe with the genres and artist distribution, as pop music is more energetic and danceable than hip hop or rock, and is also a more recent genre, while I listen to a lot of old school rap, multiple kinds of rock or soul, all of which belong to another period. To truly get a sense of how the variables that matter differ among liked and disliked songs, we plot the average of each in Figure 7 for liked and disliked songs separately before subtracting the disliked variables' average from the liked variables' average in Figure 8. Here we get meaningful concluding information on my liked music. It tends to be more instrumental, more acoustic, less positive, less energetic, and less danceable compared to music I dislike. This shows a possible negative correlation among those factors, as it wouldn't be surprising that more acoustic and instrumental music is less positive, danceable, and energetic. In Figure 9, we plot separately the same information for tempo, loudness, and popularity as the scale is different. This tells us with no doubt that the music I like is less popular, louder, and has a slower tempo. We now have a good idea of what matters and doesn't as well as what differs and doesn't among the various features that explain each track. With that in mind, we can start building a model to predict if I like or not a given song.

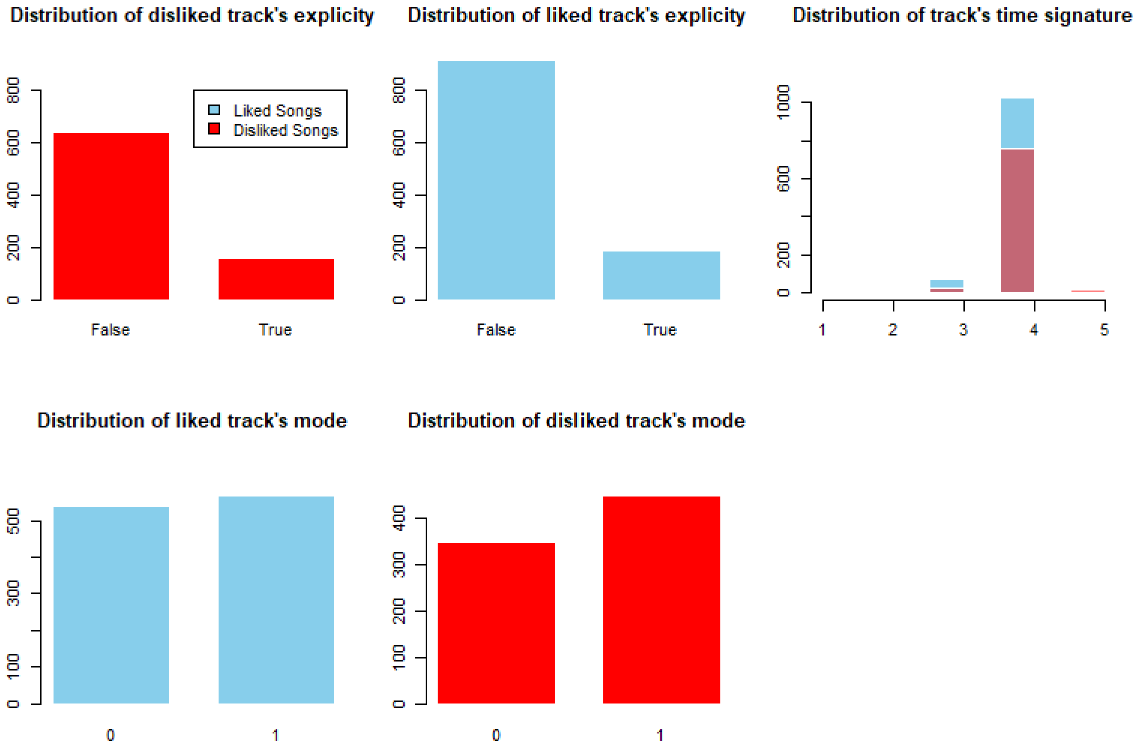


Figure 6

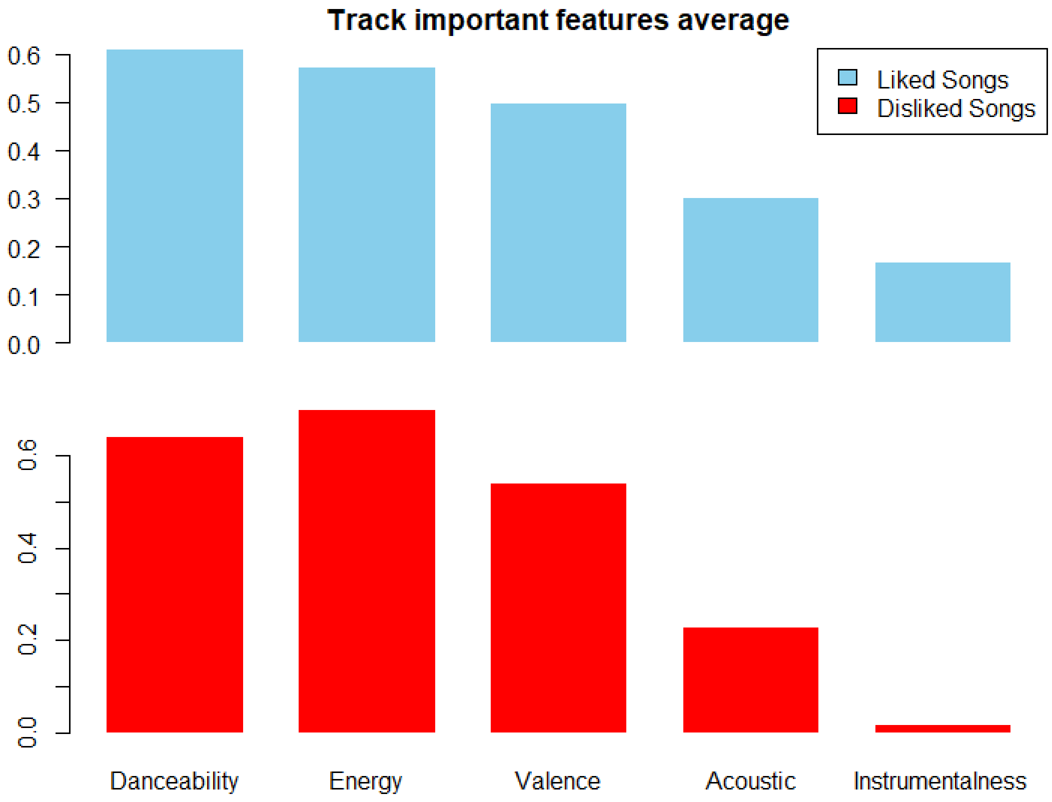


Figure 7

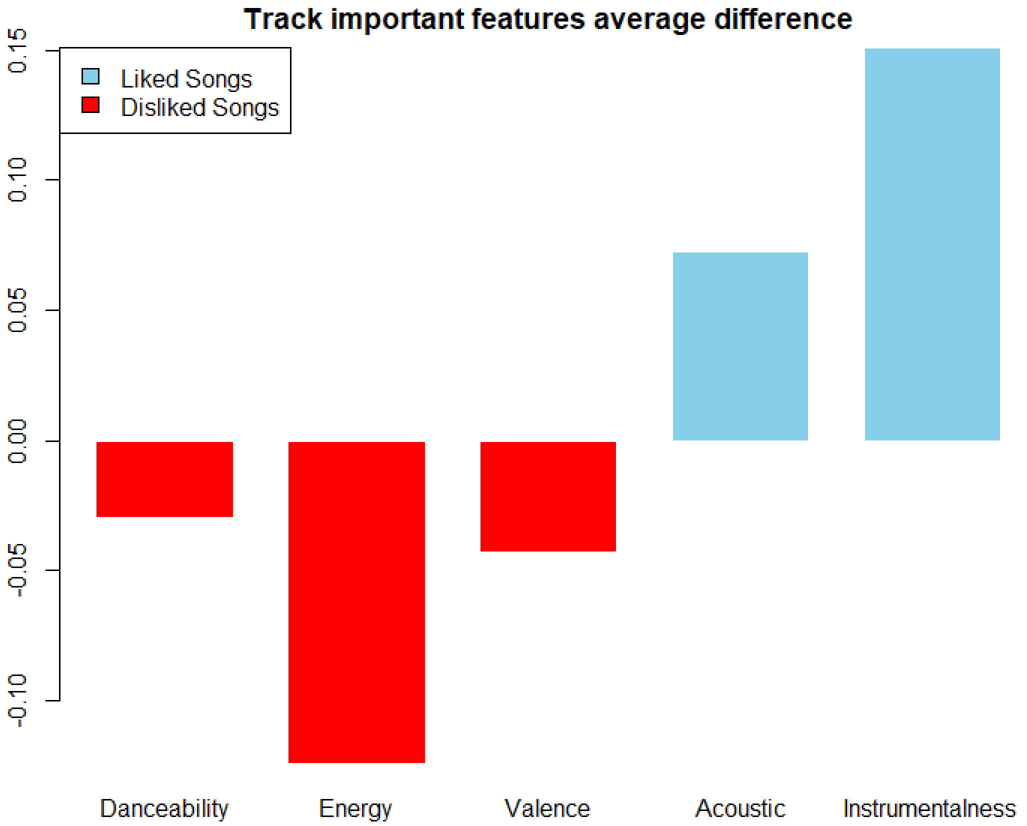


Figure 8

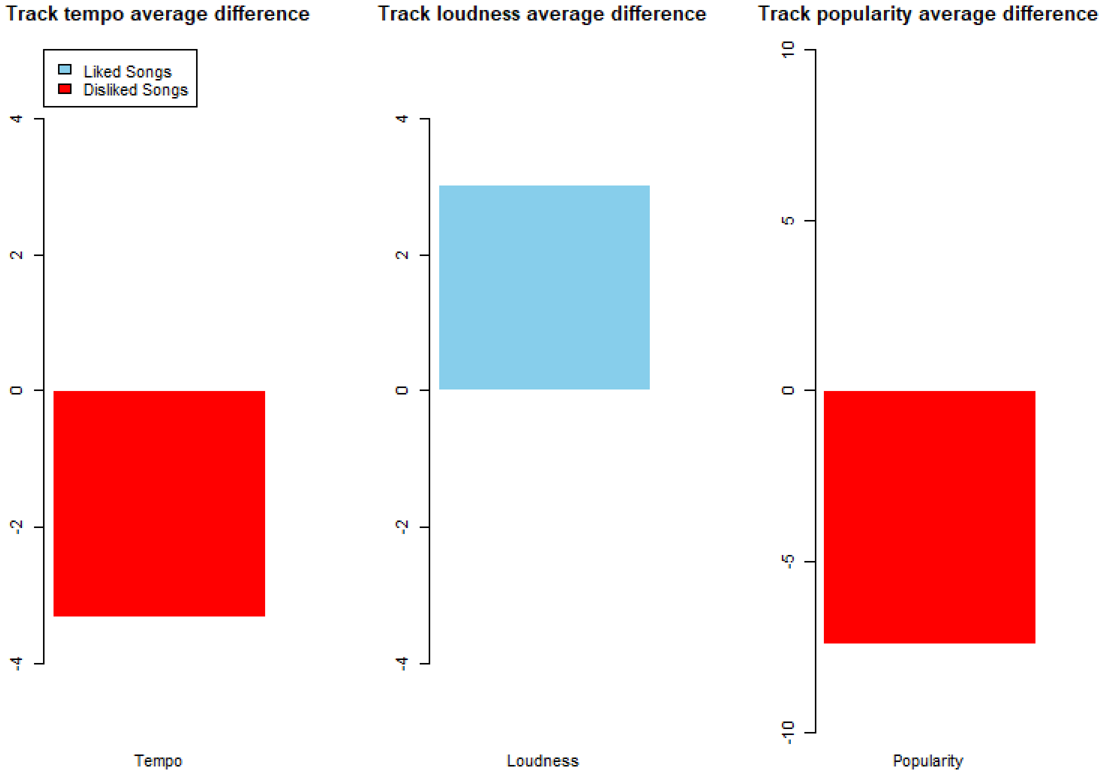
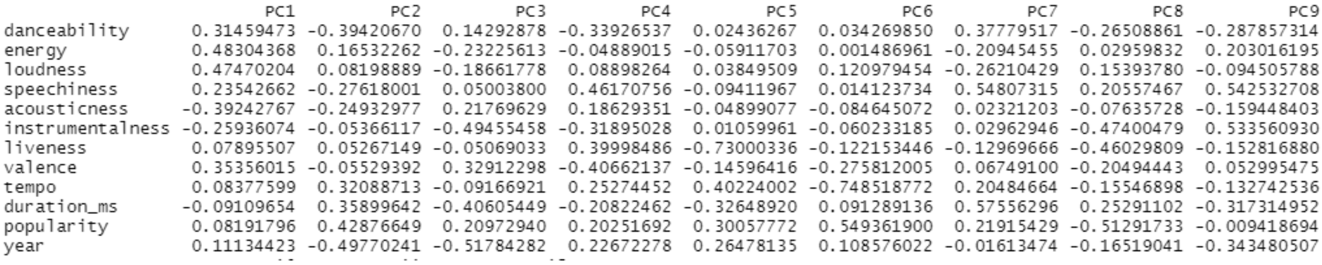


Figure 9

Model Selection & Methodology

In the effort of trying to build the best model possible, we will analyze the variable impact on the outcome (like or dislike) and among them (collinearity) with the use of PCA analysis. As we get a better idea of which predictors matter the most, we will test multiple models before performing a validation test.

First, we make the following variable categorical: 'key', 'mode', 'explicit', and 'time signature'. Then we only keep the quantitative variable to run PCA analysis. We first focus on how the variable for liked songs reacts with a PCA analysis of those (Figure 10). The first PCA focuses on 'danceability', 'energy', 'acousticness', 'loudness', and 'valence'. All of these variables translate the emotions/feeling conveyed by the track. We saw previously that my liked songs tend to be more acoustic and less danceable and energetic, so it's not surprising that the main PCA (describing 25% of the track characteristic) focuses on those aspects (the vibe of the track is also unquestionably the most important aspect). The second PCA still includes danceability but also focuses on more factual aspects about the track, such as its 'duration', 'tempo', 'popularity', and 'year' of release. While the third PCA focuses on 'instrumentalness', 'valence', 'duration', and 'year'. It makes a lot of sense, as it focuses on longer songs, which likely are more instrumental (classical music, alternative rock, soul/jazz, electronic) and belong to a specific period (year). Finally, the 4th PCA focuses on 'danceability', 'speechiness', 'instrumentalness', 'valence', and 'liveness'. The role of this PCA is probably to distinguish speechiness from instrumentalness from a track, as valence, liveness, and danceability are directly affected. Let's now take a look at how scattered are the observations on the plot of PCA1 and PCA2 (Figure 11). As expected, my liked songs are mostly scattered around, 'instrumentalness', 'duration', 'acousticness' and 'year' compared to the other variables. This plot also confirms that 'acousticness' and 'instrumentalness' are positively correlated with each other as well both being negatively correlated with 'energy' and 'loudness'. Duration is also negatively correlated with year, confirming our assumption about longer songs being older.



```
                         PC1         PC2         PC3         PC4         PC5          PC6         PC7         PC8          PC9
danceability      0.31459473 -0.39420670  0.14292878 -0.33926537  0.02436267  0.034269850  0.37779517 -0.26508861 -0.287857314
energy            0.48304368  0.16532262 -0.23225613 -0.04889015 -0.05911703  0.001486961 -0.20945455  0.02959832  0.203016195
loudness          0.47470204  0.08198889 -0.18661778  0.08898264  0.03849509  0.120979454 -0.26210429  0.15393780 -0.094505788
speechiness       0.23542662 -0.27618001  0.05003800  0.46170756 -0.09411967  0.014123734  0.54807315  0.20557467  0.542532708
acousticness     -0.39242767 -0.24932977  0.21769629  0.18629351 -0.04899077 -0.084645072  0.02321203 -0.07635728 -0.159448403
instrumentalness -0.25936074 -0.05366117 -0.49455458 -0.31895028  0.01059961 -0.060233185  0.02962946 -0.47400479  0.533560930
liveness          0.07895507  0.05267149 -0.05069033  0.39998486 -0.73000336 -0.122153446 -0.12969666 -0.46029809 -0.152816880
valence           0.35356015 -0.05529392  0.32912298 -0.40662137 -0.14596416 -0.275812005  0.06749100 -0.20494443  0.052995475
tempo             0.08377599  0.32088713 -0.09166921  0.25274452  0.40224002 -0.748518772  0.20484664 -0.15546898 -0.132742536
duration_ms      -0.09109654  0.35899642 -0.40605449 -0.20822462 -0.32648920  0.091289136  0.57556296  0.25291102 -0.317314952
popularity        0.08191796  0.42876649  0.20972940  0.20251692  0.30057772  0.549361900  0.21915429 -0.51291733 -0.009418694
year              0.11134423 -0.49770241 -0.51784282  0.22672278  0.26478135  0.108576022 -0.01613474 -0.16519041 -0.343480507
```

Figure 10

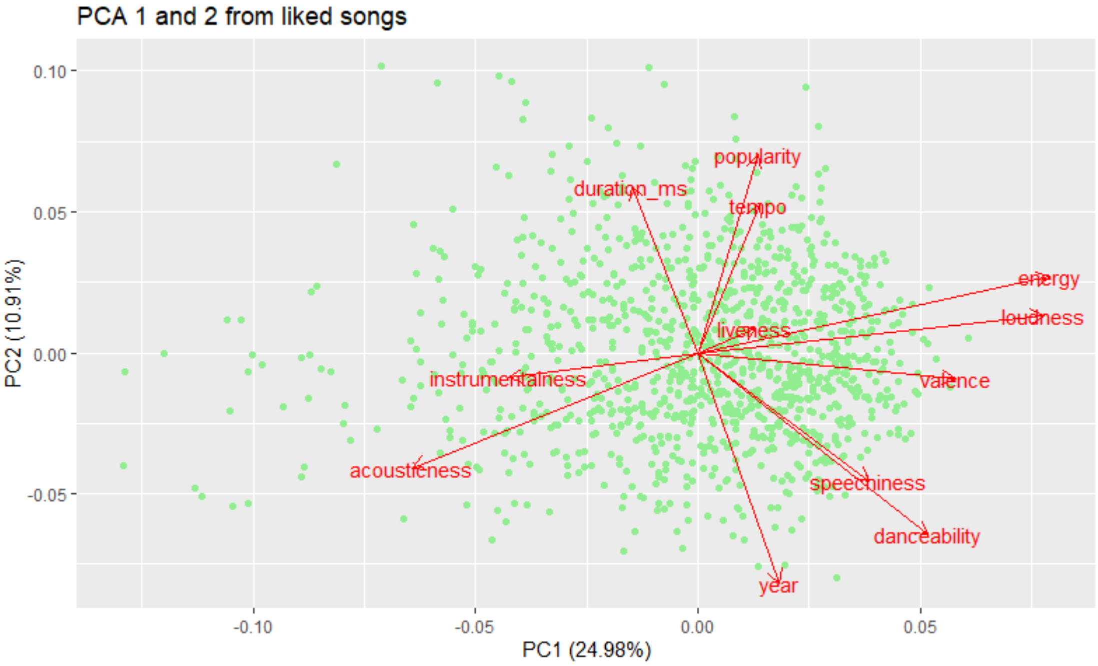
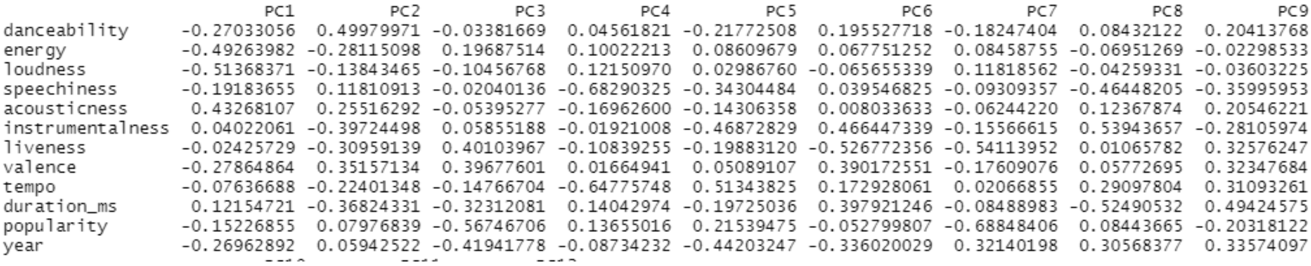


Figure 11

Now let's focus on the PCA analysis of my disliked songs (Figure 12) with the first PCA explaining 22.65% of disliked track features. It focuses on 'energy', 'loudness', and 'acousticness'. 'Energy' and 'loudness' are surely affected by each other and positively correlated, while acousticness, as we saw, is fore sure negatively correlated with those 2 variables. This first PCA tells us that those variables, among my disliked songs, are the most significant to make a fist distinction between the tracks. While the 2nd PCA focuses on 'danceability', 'instrumentalness', and 'duration' as 'danceability' is surely negatively correlated with the 2 other factors. As we plot the PCA1 and 2 (Figure 13) we observe in this case that 'duration' and 'instrumentalness' is negatively correlated with 'danceability' and 'valance' instead of 'year', which has less importance among my disliked songs (as they are mostly all from the same period compared to my liked songs).



```
                        PC1         PC2         PC3         PC4         PC5          PC6         PC7         PC8         PC9
danceability    -0.27033056  0.49979971 -0.03381669  0.04561821 -0.21772508  0.195527718 -0.18247404  0.08432122  0.20413768
energy          -0.49263982 -0.28115098  0.19687514  0.10022213  0.08609679  0.067751252  0.08458755 -0.06951269 -0.02298533
loudness        -0.51368371 -0.13843465 -0.10456768  0.12150970  0.02986760 -0.065655339  0.11818562 -0.04259331 -0.03603225
speechiness     -0.19183655  0.11810913 -0.02040136 -0.68290325 -0.34304484  0.039546825 -0.09309357 -0.46448205 -0.35995953
acousticness     0.43268107  0.25516292 -0.05395277 -0.16962600 -0.14306358  0.008033633 -0.06244220  0.12367874  0.20546221
instrumentalness 0.04022061 -0.39724498  0.05855188 -0.01921008 -0.46872829  0.466447339 -0.15566615  0.53943657 -0.28105974
liveness        -0.02425729 -0.30959139  0.40103967 -0.10839255 -0.19883120 -0.526772356 -0.54113952  0.01065782  0.32576247
valence         -0.27864864  0.35157134  0.39677601  0.01664941  0.05089107  0.390172551 -0.17609076  0.05772695  0.32347684
tempo           -0.07636688 -0.22401348 -0.14766704 -0.64775748  0.51343825  0.172928061  0.02066855  0.29097804  0.31093261
duration_ms      0.12154721 -0.36824331 -0.32312081  0.14042974 -0.19725036  0.397921246 -0.08488983 -0.52490532  0.49424575
popularity      -0.15226855  0.07976839 -0.56746706  0.13655016  0.21539475 -0.052799807 -0.68848406  0.08443665 -0.20318122
year            -0.26962892  0.05942522 -0.41941778 -0.08734232 -0.44203247 -0.336020029  0.32140198  0.30568377  0.33574097
```

Figure 12

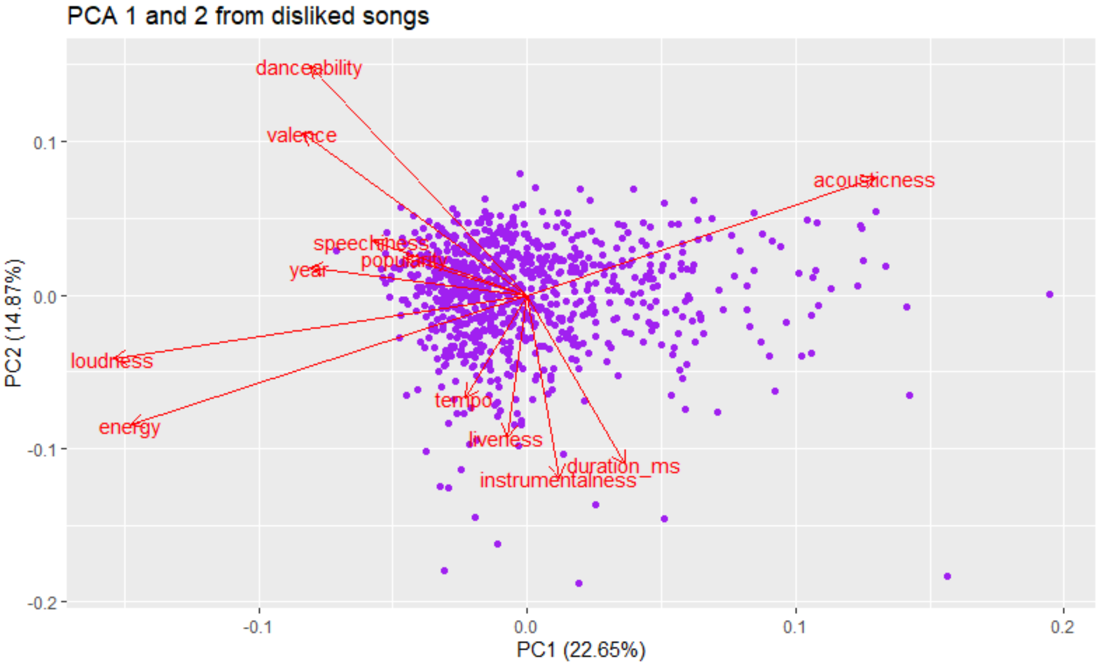
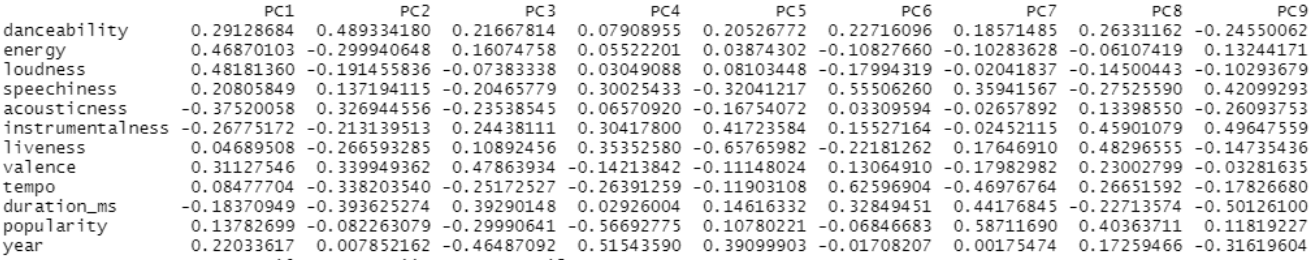


Figure 13

Finally let's take a look at the PCA analysis of the 'songs' dataset, which includes both liked and disliked songs, as we now know how variables interfere with liked and disliked songs individually (Figure 14). The first PCA describing both liked and disliked songs focus on 'energy', 'loudness', and 'acousticness'. As they are variables that significantly differ among liked and disliked songs it's not surprising to see that the main PCA relies on those, also 'acousticness' is negatively correlated with 'energy' and 'loudness', therefore including it to distinguish liked and disliked songs make sense. The second PCA focus especially on 'danceability' and 'duration'. Again, those are negatively correlated, as longer tracks are usually calmer and short tracks more energetic (pop music), it helps distinguish liked songs from disliked songs as they both differ on those variables. From the plot of this PCA analysis (Figure 15), we can see how the liked songs are scattered differently compared to disliked songs. As expected, we see much more observation of liked songs around ‘acousticness’, ‘instrumentalness’ and ‘duration’ while the disliked songs occupy the other end with variable negatively correlated with the one just stated (‘energy’, ‘danceability’).



```
                         PC1          PC2         PC3         PC4         PC5         PC6         PC7         PC8         PC9
danceability      0.29128684  0.489334180  0.21667814  0.07908955  0.20526772  0.22716096  0.18571485  0.26331162 -0.24550062
energy            0.46870103 -0.299940648  0.16074758  0.05522201  0.03874302 -0.10827660 -0.10283628 -0.06107419  0.13244171
loudness          0.48181360 -0.191455836 -0.07383338  0.03049088  0.08103448 -0.17994319 -0.02041837 -0.14500443 -0.10293679
speechiness       0.20805849  0.137194115 -0.20465779  0.30025433 -0.32041217  0.55506260  0.35941567 -0.27525590  0.42099293
acousticness     -0.37520058  0.326944556 -0.23538545  0.06570920 -0.16754072  0.03309594 -0.02657892  0.13398550 -0.26093753
instrumentalness -0.26775172 -0.213139513  0.24438111  0.30417800  0.41723584  0.15527164 -0.02452115  0.45901079  0.49647559
liveness          0.04689508 -0.266593285  0.10892456  0.35352580 -0.65765982 -0.22181262  0.17646910  0.48296555 -0.14735436
valence           0.31127546  0.339949362  0.47863934 -0.14213842 -0.11148024  0.13064910 -0.17982982  0.23002799 -0.03281635
tempo             0.08477704 -0.338203540 -0.25172527 -0.26391259 -0.11903108  0.62596904 -0.46976764  0.26651592 -0.17826680
duration_ms      -0.18370949 -0.393625274  0.39290148  0.02926004  0.14616332  0.32849451  0.44176845 -0.22713574 -0.50126100
popularity        0.13782699 -0.082263079 -0.29990641 -0.56692775  0.10780221 -0.06846683  0.58711690  0.40363711  0.11819227
year              0.22033617  0.007852162 -0.46487092  0.51543590  0.39099903 -0.01708207  0.00175474  0.17259466 -0.31619604
```

Figure 14

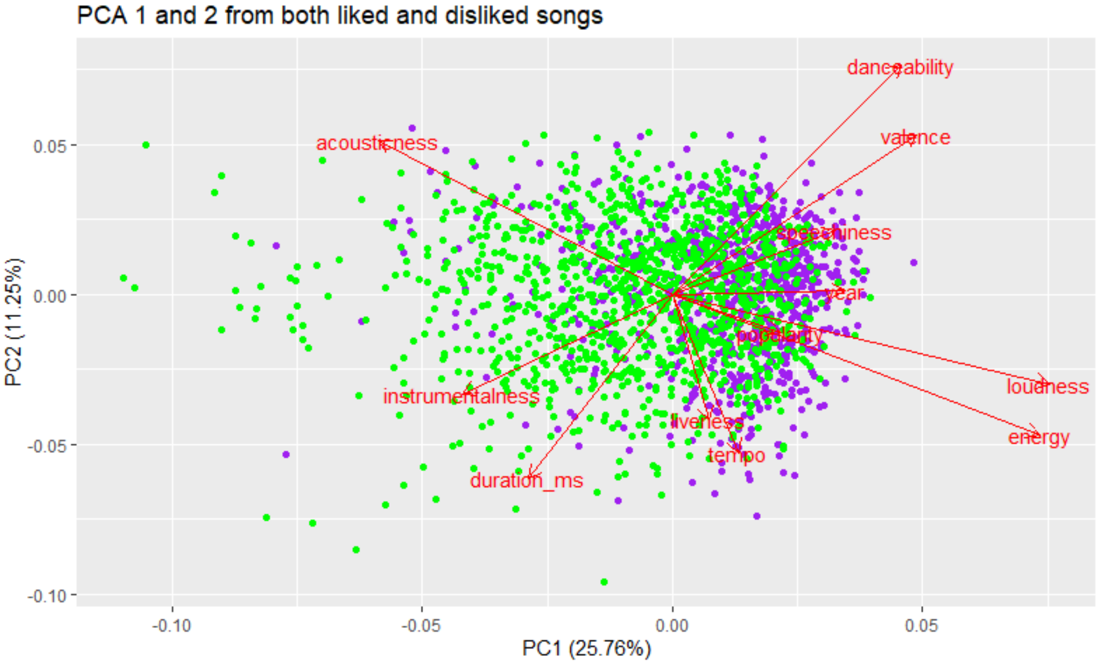
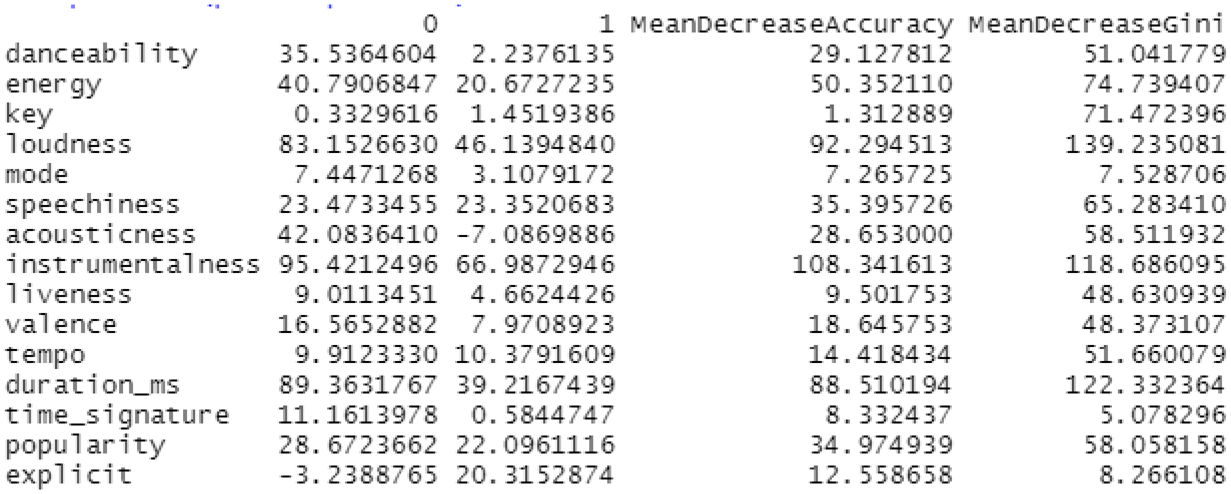


Figure 15

We now have enough insight into our variables to start building our first models. We first create a random forest with all the predictors, to evaluate the importance of each with the ‘importance()’ attribute from the algorithm. Our random forest includes: ‘danceability’, ‘energy’, ‘key’, ‘loudness’, ‘mode’, ‘speechiness’, ‘acousticness’, ‘instrumentalness’, ‘valence’, ‘tempo’, ‘duration’, ‘popularity’ and ‘explicit’. From the importance() function (Figure 16), we get the following variable as the top predictors: ‘instrumentalness’, ‘loudness’, ‘duration’, ‘energy’, and ‘popularity’. The 5 least important predictors are as follows: ‘key’, ‘time signature’, ‘mode’, ‘liveness’, and ‘explicit’. Those were the results we were expecting based on the data description analysis we did in the first part.



```
                         0          1 MeanDecreaseAccuracy MeanDecreaseGini
danceability     35.5364604  2.2376135            29.127812        51.041779
energy           40.7906847 20.6727235            50.352110        74.739407
key               0.3329616  1.4519386             1.312889        71.472396
loudness         83.1526630 46.1394840            92.294513       139.235081
mode              7.4471268  3.1079172             7.265725         7.528706
speechiness      23.4733455 23.3520683            35.395726        65.283410
acousticness     42.0836410 -7.0869886            28.653000        58.511932
instrumentalness 95.4212496 66.9872946           108.341613       118.686095
liveness          9.0113451  4.6624426             9.501753        48.630939
valence          16.5652882  7.9708923            18.645753        48.373107
tempo             9.9123330 10.3791609            14.418434        51.660079
duration_ms      89.3631767 39.2167439            88.510194       122.332364
time_signature   11.1613978  0.5844747             8.332437         5.078296
popularity       28.6723662 22.0961116            34.974939        58.058158
explicit         -3.2388765 20.3152874            12.558658         8.266108
```

Figure 16

We can now, with confidence, start testing different models with the following variable that we keep as predictors: ‘danceability’, ‘energy’, ‘loudness’, ‘speechiness’, ‘acousticness’, ‘instrumentalness’, ‘valence’, ‘tempo’ and ‘popularity’. We first build a logistic regression, then we build another random forest with hypervariable ‘ntree=1000’, followed by a forest classifier with ‘cp=0.01’, then finally we build a gradient boosting model with hyper variables ‘n.trees=1000’ and ‘interaction.depth=3’. To test the predictive power of those model on our dataset, we split our ‘songs’ dataset into a train set (70% of the data) and a test set (the remaining 30%) for all the model except random forest who can give us directly the out-of-bad performance accuracy. We plot in Figure 17 the test performance result of each model. As we can see, the performance of each model is more or less the same, gradient boosting algorithm performs the best with an accuracy of 80%, while random forest algorithm is very close with an accuracy of 78.7%, on the other hand, logistic regression and forest classifier perform slightly worse with an accuracy of 75.5% and 74.8% respectively. With those results in mind, we stick to the gradient boosting algorithm as it's the best performing model among all of them.

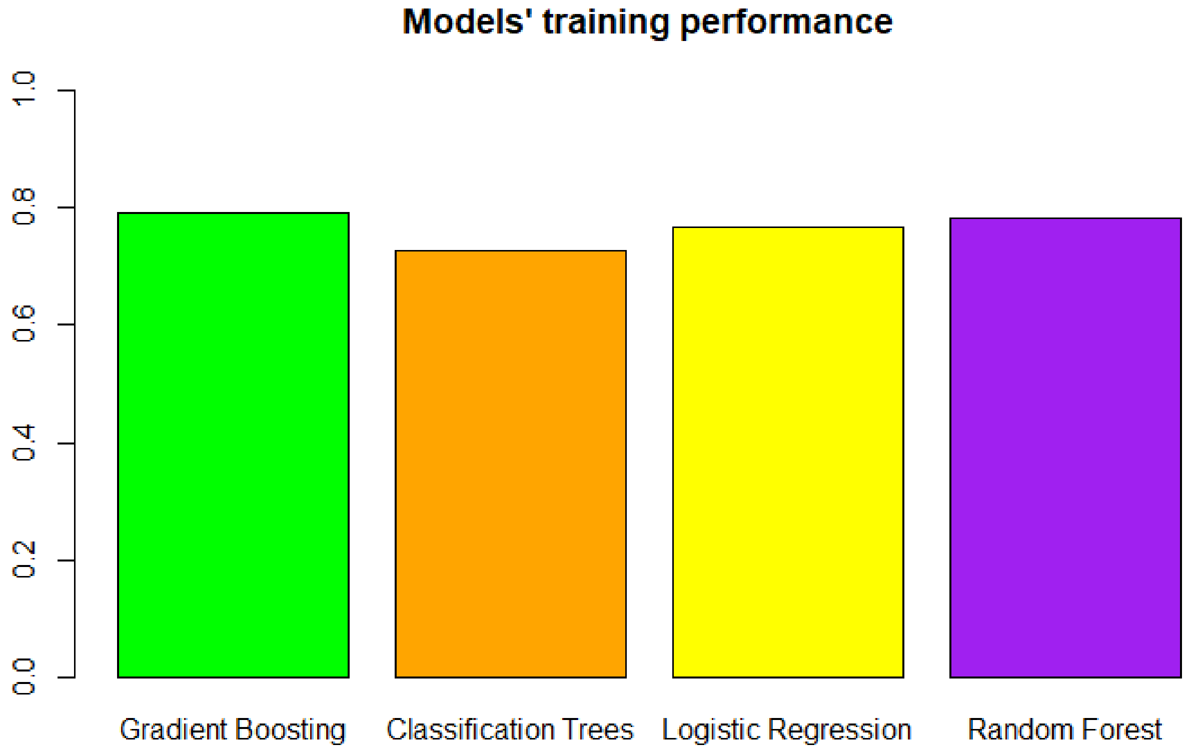


Figure 17

Now that we know which algorithm we want to use, we want to tune it a bit to try to improve our model performance. Since random forest and gradient boosted algorithms performed almost the same, we will apply the tuning on both and make our final decision after. The first step I do is add more predictors to see how it would impact the accuracy since I realized the first random forest that was built to test the predictors' importance had a slightly lower OOB error rate. By adding all predictors and running the models again the accuracy doesn't significantly change even if there's a really small improvement. We can play with the hyperparameters of both models to try to improve our results. Setting the n.trees to '1500, 2000 and 10 000' and interaction.depth to '2, 4, 5' didn't provide any improvement to the models. However, there's one variable that wasn't used as it's a categorical variable with way too many possible values, 'genres'. To come with a way to use it, I decided to make 'genre' a binary variable, set to 1 if the genre of a track is present in my top 25 genres (recall Figure 2) and set to 0 if that's not the case. With such a new variable I'm pleased with the new testing performance accuracy of the models (Figure 18). The accuracy of the gradient boosting algorithm jumped to around 85% while random forest reached above 80%. We now have a clear best model to use for our final model, which is the gradient boosting algorithm. Let's now see how this model performs on a validation test.

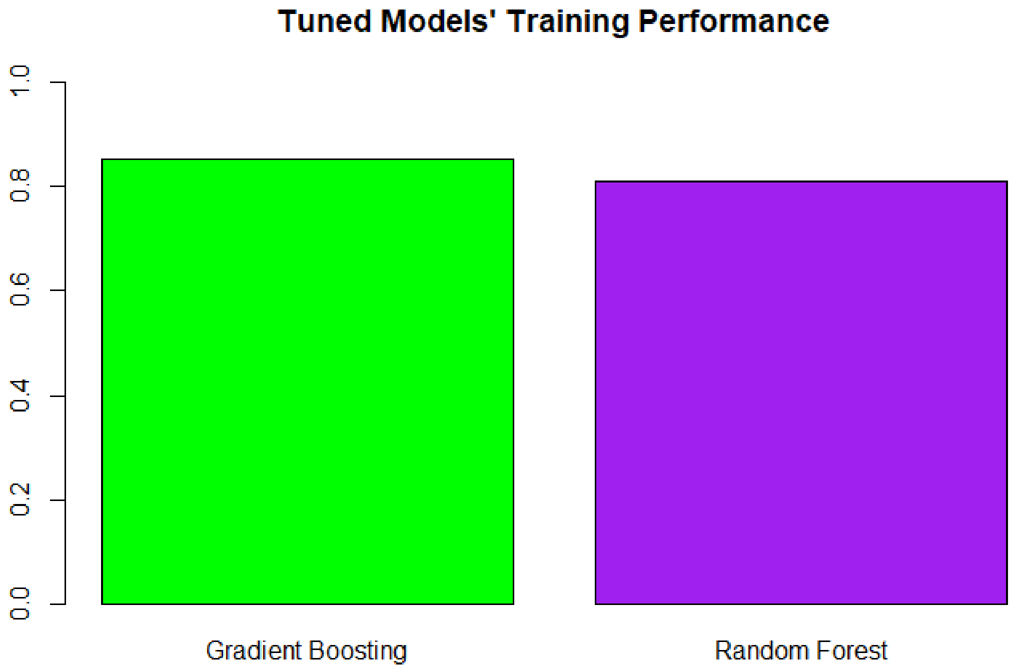
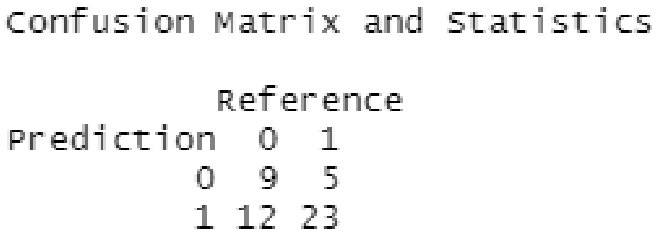


Figure 18

Results

With our model being finalized, we do a validation test to see how it performs with unknown data. For that purpose, I asked 2 of my dear friends (again) to participate a bit. I told them to each make a playlist where they gather songs they knew for sure I've never listened to, from various genres, and to not let my taste influence them in any way. On the contrary, I was looking for them to surprise me, with content I'm not used to listening to (they did a great job). Then, to gather this new data into a test dataset, I proceed the same way as I did with my liked and disliked playlist, by using the Spotify API in python. I end up with 'full_test_playlist.csv', containing 49 tracks from the 2 playlists. Before running the model, I listened to each song and gave them a value of 1 if I liked the song and 0 if I disliked the song. Then came the eagerly-awaited moment, how will the final model prediction perform? I ran the prediction and got an accuracy score of 65.3%, the model being able to correctly predict the outcome 32 tracks while 17 tracks were missed predicted. To get a better idea of where the prediction failed, let's analyze the confusion matrix we got (Figure 19). We see that among the correctly predicted tracks, 9 were disliked songs and 23 were liked songs, however, the model predicted 12 songs as liked when they were songs I disliked and predicted 5 songs as disliked when they were songs I liked listening to.



```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0  9  5
         1 12 23
```

Figure 19

Conclusion

Overall I'm really satisfied with the numerous findings this project revealed about my musical taste, in a way some findings were expected as I know my taste, but it was clearly great analysing all those factors (which for some I had never thought of before) and see how they change for describing liked and disliked songs. I was able to learn about the factors that really matter for my music. As I listen to music mostly alone, and whenever I can, I knew I wouldn't be a huge fan of tracks with high danceability because dancing alone is quickly boring. I was pleased to see that instrumental and long tracks played a bigger role than what I would have expected to distinguish music I don't like, as those variables matter a lot to me. It shows that the model was able to get a good grasp of my taste. However I was quite surprised to see 'loudness' as a predictor that describes my music, as it's positively correlated with energy (even in the liked songs PCA plot). To better understand, I looked up the loudest songs of my liked songs dataset (over 15db) and among those 86 songs, the dominant genres were album rock, classical, electronic and adult content, and their 'energy' rating wasn't particularly high. I guess this correlation must come from songs slightly less loud and more energetic. Still I just made the discovery of a genre I never heard of before: 'adult content', which was in my top genres also. Luckily, It has nothing to do with what I thought of first place. When I looked for the liked songs' artist which had this genre, it returned me artists like Nina Simone, Micheal Franks or Geore Benson. This was just an attempt from Spotify to tell me that I should be 40 years older regarding my taste (Adult standards (also sometimes known as the nostalgia format) is aimed at "mature" adults, meaning mainly those persons over 50 years of age, but it is mostly targeted for senior citizens src.), but at least I would have great music taste. Now I truly feel like I know myself a bit better and everything will be clear in my head the next time someone asks me about the music I listen to (this person will certainly have to take a seat).

R and Python code can be found in the main directory of the submission folder.