Spotify Algorithm Analysis Part 2

17 July 2025

Authors:
Marc Schmieder ✉ marc.schmieder01@gmail.com 🌐 www.instagram.com/marcschmiedermusic
Xixue Lou ✉ xixue.lou@uni-muenster.de 🌐 www.linkedin.com/in/xixue-lou-9602b9178
Andrew Southworth ✉ andrew@generastudios.com 🌐www.andrewsouthworth.com

The results of this report, the data, and the code are publicly available. This is a project made possible by the community and will always be freely accessible to the community.

Here you can find the videos that elaborate on this report with additional explanations:
todo: insert video link 1
todo: insert video link 2

There has been a precursor to this analysis. You can find the report on **Spotify Algorithm Analysis Part 1** here:
https://github.com/yellowmarc333/spotifyalgorithmanalysis2/tree/main/03_docs
Here is Andrew's video breaking down the results from Part 1:
https://www.youtube.com/watch?v=oRde100NPEQ

Introduction

Probably all the artists that have released music in the past years have been wanting to grow on Spotify, using the exposure of their algorithmic playlists to gain new streams and listeners, hopefully to increase their reach and raise the income level of streaming royalties.

The metric called Popularity Index is community-wide known for being a gateway to being featured on the algorithmic playlist *Release Radar*, *Discover Weekly Radio,* and many more.

A widely used strategy by artists was as follows:
By promoting the released song heavily with meta ads and organic content, the goal was to bring the Popularity Index to a number above 20. This could trigger an algorithmic Release Radar push, which brought in additional streams. With that push, it was possible to reach a Popularity Index of over 30, which made the song eligible for Discover Weekly.

Although the algorithmic exposure on Release Radar was limited to the first 5 weeks after the release, a song could stay multiple months or even years in Discover Weekly.
This could in practice mean that with an initial promotion of a few thousand streams, with this strategy, the track could obtain multiple hundreds of thousands of streams in a lifetime.

But algorithms get changed by the platform all the time, and we have to ask ourselves if this strategy still works in 2025. Artists have reported that the amount of Release Radar and

Discover Weekly streams has decreased and that Radio has become the dominant algorithmic source.

How does Radio work? Have Release Radar and Discover Weekly streams really decreased? Does the ‘old strategy’ still work?

To answer those questions, one must have access to data from Spotify for Artists for many songs.

Authors

Andrew Southworth made this whole analysis possible in the first place by exposing this issue to the community and then gathering data from different artists via a Google form that you can find here:
https://docs.google.com/forms/d/e/1FAIpQLSdYVh8C4UlbRyW9GqpljcxeR-wx3Ctp6ts3Lalet8l-WqV6_A/viewform

A huge thank you goes out to Andrew as well as to everyone who filled out the form and provided their valuable data - without the participation of this community, these insights would not exist.

The data analysis for this report was conducted by Marc Schmieder and Xixue Lou. As they both worked on separate sections of this report, the style of graphics and tables could vary.

Xixue Lou is a Data Scientist who has voluntarily joined this project and helped out a lot with all parts of the analysis. She is currently open for projects and can be contacted via LinkedIn (linked on top of this report).

Data Description

Data Quality

The data was collected with a Google form by the community. 2 sources were incorporated, each participant's personal *Spotify for Artist* account and the *Spotify for Developers* platform to retrieve the popularity index. The data quality is imperfect due to several reasons. Participants had to manually fill out the Google Form, so easy mistakes could have been made, for example, mistyping, putting the wrong release date, and so on. As described in the following chapter *(preprocessing and data cleaning)*, we tried our best to correct data input mistakes, but probably could not correct them all.

Additionally, there is the issue of data lag, as both the *Spotify for Artists* account and the *Spotify for Developers* platform have a time delay in updating their stats. The *Spotify for Artists* stats are always a snapshot of a specific time of the day, and we have to be conscious about the fact that this is not real-time data, and thus, models created with this data will be imperfect.

Description of Variables/Attributes

Each data point (row) in the dataset represents one song’s attributes at a given time, containing the following 45 variables (columns) provided or engineered through existing data:

Table 1: Summary of dataset variables

Variable Name	Data Source	Data Type	Description
ArtistName	Provided	String	Name of the artist.
SongName	Provided	String	Title of the song.
WhatGenreOfMusicMostAccuratelyDescribesThisArtistProject	Provided	String	Most accurate genre description of the song.
ReleaseDate	Provided	Date	Release date of the song.
StreamsLast28Days	Provided	Integer	Number of streams in the last 28 days.
ListenersLast28Days	Provided	Integer	Number of listeners in the last 28 days.
SavesLast28Days	Provided	Integer	Number of saves in the last 28 days.
PlaylistAddsLast28Days	Provided	Integer	Number of playlists where the song is added in the last 28 days.
StreamsLast7Days	Provided	Integer	Number of streams in the last 7 days.
ListenersLast7Days	Provided	Integer	Number of listeners in the last 7 days.
SavesLast7Days	Provided	Integer	Number of saves in the last 7 days.
PlaylistAddsLast7Days	Provided	Integer	Number of playlists where the song is added in the last 28 days.

Table 1: Summary of dataset variables

StreamsAllTime	Provided	Integer	Number of streams since release.
RadioStreamsLast28Days	Provided	Integer	Streams from Spotify Radio in the last 28 days.
DiscoverWeeklyStreamsLast28Days	Provided	Integer	Streams from Discover Weekly in the last 28 days.
ReleaseRadarStreamsLast28Days	Provided	Integer	Streams from Release Radar in the last 28 days.
RadioStreamsLast7Days	Provided	Integer	Streams from Spotify Radio in the last 7 days.
DiscoverWeeklyStreamsLast7Days	Provided	Integer	Streams from Discover Weekly in the last 7 days.
ReleaseRadarStreamsLast7Days	Provided	Integer	Streams from Release Radar in the last 7 days.
CurrentSpotifyFollowers	Provided	Integer	Number of current Spotify followers.
PopularityIndex	Provided	Integer	Popularity Index of the song.
WhatIsTheDominantWayThisArtistHasPromotedThisSong	Provided	String	Dominant promotion method used for the song.
ReleaseConsistency	Provided	String	Artist’s release frequency.
HowManySongsDoYouHaveInRadioRightNow	Provided	Integer	Number of songs from the artist currently in Spotify Radio.
HowManySongsHasThisArtistEverReleased	Provided	Integer	Total number of songs the artist has released.

Table 1: Summary of dataset variables

AreYouIndependentOrSignedToALabel	Provided	String	Whether the artist is independent or signed to a label.
IsThisSongOptedIntoSpotifyDiscoveryMode	Provided	String	Whether the song is opted into Spotify Discovery Mode. Not included in the first version of the survey, and was added later.
Timestamp	Provided	Date	Timestamp of data entry, automatically created by submitting the data.
DaysSinceRelease	Engineered	Integer	Number of days between the ReleaseDate and *Timestamp* (rounded up).
WeeksSinceRelease	Engineered	Integer	Number of weeks between *ReleaseDate* and *Timestamp* (rounded up).
NewRelease	Engineered	String	‘yes’ if the song was released within the last 35 days; otherwise ‘no’.
ReleasePhaseEarly	Engineered	Bool	TRUE if the song was released within the last 35 days; otherwise FALSE.
PI_Category	Engineered	String	Categorized version of *PopularityIndex* (ranges: 0-5, 6-10, 11-13, 14-16, 17-19, 20-21, 22-25, 26-29, 30-32, 33-35, 36-40, 41-45, 46-50, 51-100)
PI_Category_2	Engineered	String	Another categorized version of *PopularityIndex* (ranges: 0-5, 6-10, 11-15, 16-20, 21-25, 26-30, 31-40, 41-100)
ListenersStreamRatio28Days	Engineered	Float	Ratio of *ListenersLast28Days* to *StreamsLast28Days*

Table 1: Summary of dataset variables

ListenersStreamRatio7Days	Engineered	Float	Ratio of *ListenersLast7Days* to *StreamsLast7Days*
MeanRadioStreams7Days	Engineered	Float	Mean value of *RadioStreams7Days* for all data points with the same popularity index
RelativeRadioStreams7Days	Engineered	Float	Value of *RadioStreams7Days* divided by the *MeanRadioStreams7Days* for the corresponding Popularity Index
AlgoStreams28Days	Engineered	Integer	Sum of algorithmic streams (*RadioStreamsLast28Days*, *DiscoverWeeklyStreamsLast28Das* and *ReleaseRadarStreamsLast28Days)* in the last 28 days
AlgoStreams7Days	Engineered	Integer	Sum of algorithmic streams (*RadioStreamsLast7Days*, *DiscoverWeeklyStreamsLast7Das* and *ReleaseRadarStreamsLast7Days)* in the last 7 days
NonAlgoStreams28Days	Engineered	Integer	Non-algorithmic streams (*StreamsLast28Days* minus *AlgoStreams28Days)* in the last 28 days
NonAlgoStreams7Days	Engineered	Integer	Non-algorithmic streams (*StreamsLast7Days* minus *AlgoStreams7Days)* in the last 7 days

Preprocessing and data cleaning

To identify inconsistent and irregular inputs, the following cleaning processes are done:

- Searching and correcting inconsistencies in *ReleaseDate* for the same songs with multiple inputs.
- Searching for negative results in the variable *DaysSinceRelease* to identify and correct irregular release dates.

- Detecting inputs where irregular/unreasonable relations between two variables exist. In cases we couldn't make sense of it, the irregular value is replaced by NA (a missing value). This includes:
  - *StreamsLast28Days* < *Listeners28Days*
  - *Saves28Days* > *Listeners28Days*
  - *PlaylistAdds28Days* > *Listeners28Days*
  - *StreamsLast28Days* > *StreamsAlltime**1.02 (due to potential delay in data synchronization, it is possible that *StreamsLast28Days is* slightly larger than *StreamsAlltime*. To take this factor into account, we allow a tolerance by multiplying *StreamsAlltime* by 1.02.)
  - *RadioStreamsLast28Days* > *StreamsLast28Days*
  - *DiscoverWeeklyStreamsLast28Days* > *StreamsLast28Days*
  - *ReleaseRadarStreamsLast28Days* > *StreamsLast28Days*
  - The above checks for the same variables for the last 7 days
- Searching for irregular values in variable *PopularityIndex* (<0 or >100). No entry is found.

Overview of the dataset

After the manual cleaning process of data points containing missing or incorrect values, the remaining 508 data points are used for the analysis. This means that the analysis builds on the data of 508 songs, recorded at a specific time point. This section shows a brief overview of the dataset.

Songs, Artists, Release Consistency

This section summarizes key information about the artists and songs included in the dataset.

In total, 326 songs by 157 artists are included (excluding anonymous entries), with 147 (93.6%) independent and 10 (6.4%) signed to a label.

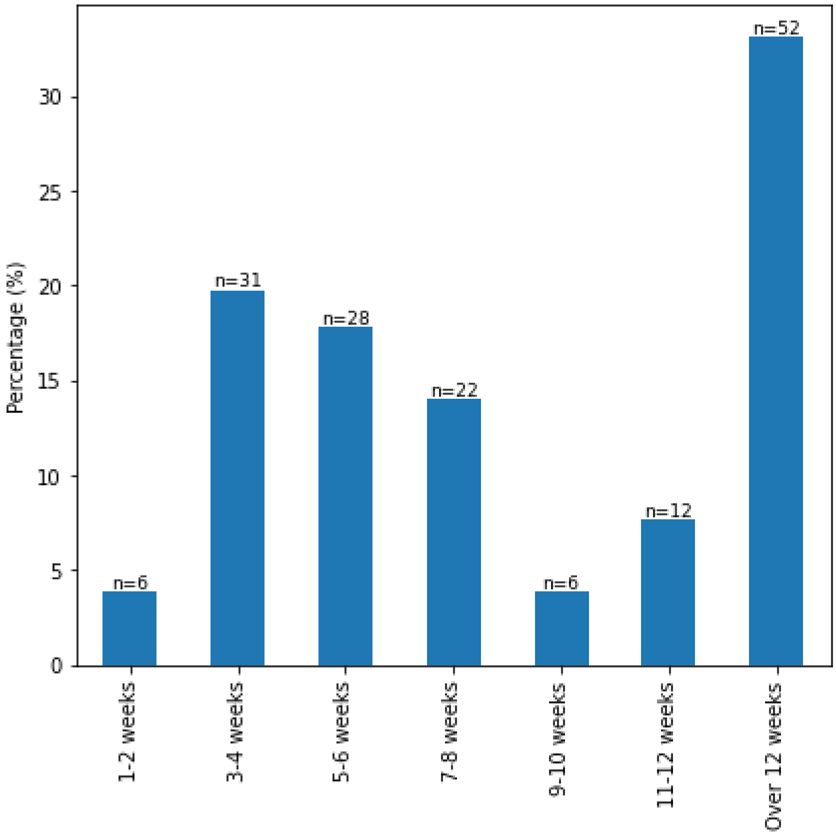
On average, an artist has 3135 followers (median=1000), with values ranging from 3 to 83025.

The average number of songs released per artist is 39 (median=21), ranging from 1 to 963.

The number of songs currently in Spotify Radio ranges from 0 to 55, with an average of 8 and a median of 10.

Figure 1 shows the distribution of release consistency (this is a summary on artist level). More than 35% (52 out of 157) of the artists maintain a release interval of more than 12 weeks.

Figure 1: Distribution of release consistency



This section presents an exploratory analysis of the dataset. **Note that all the following results are derived from all 508 data points, including multiple records for the same song at different times.**

Genre

Figure 2: Distribution of genres

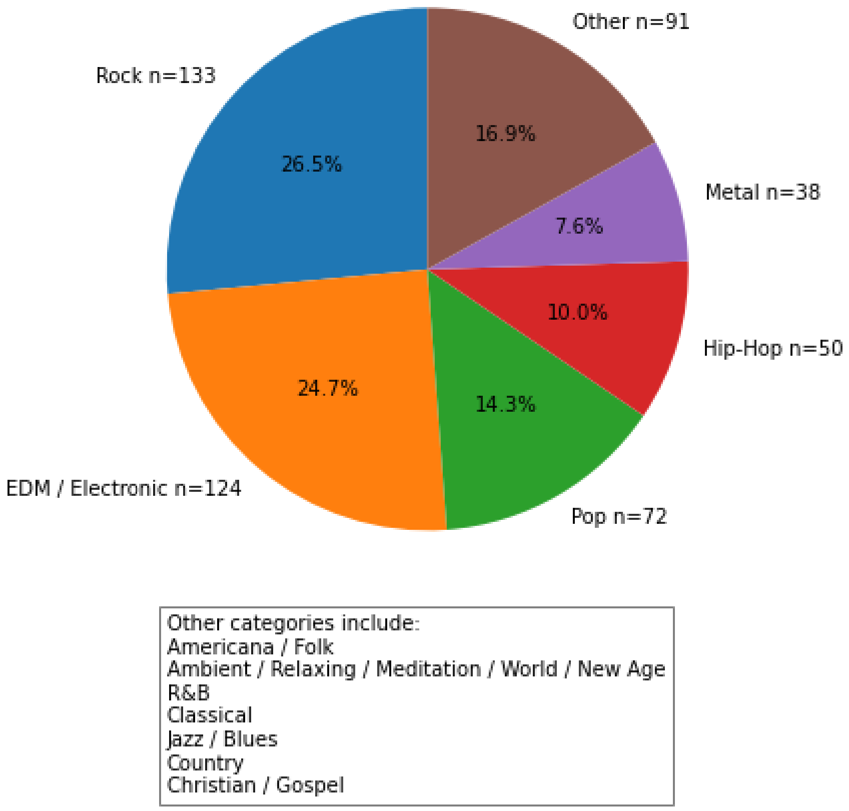


Figure 2 shows the genre distribution. Over 50% are categorized in two main genres: Rock(26.5%, 133 out of 508 entries) and EDM/Electronic (24.7%, 124 out of 508 entries).

Promotional Method

Figure 3: Distribution of promotional method

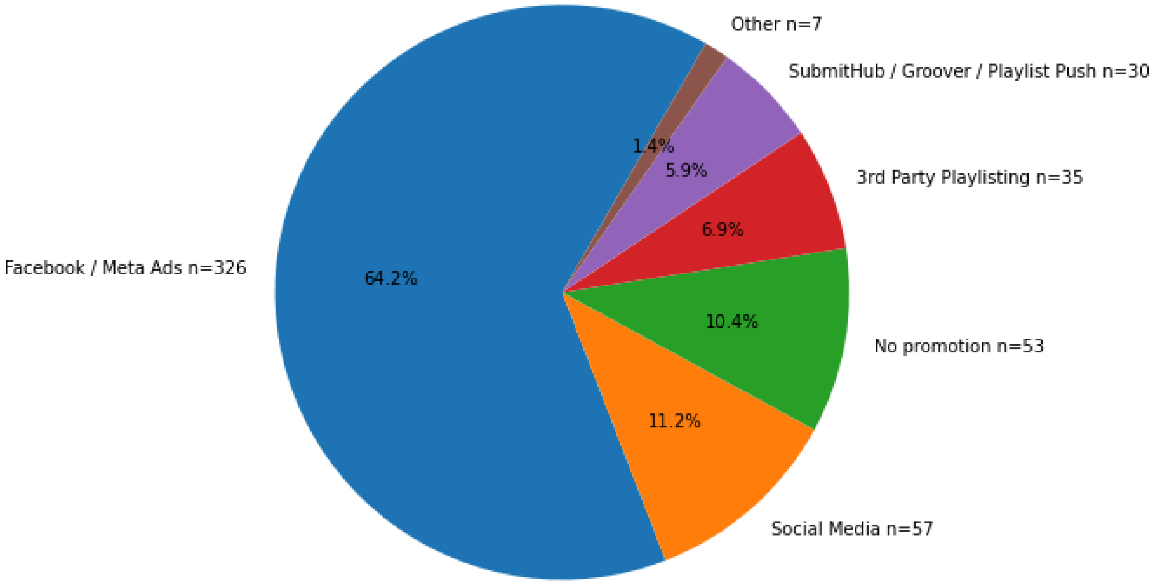
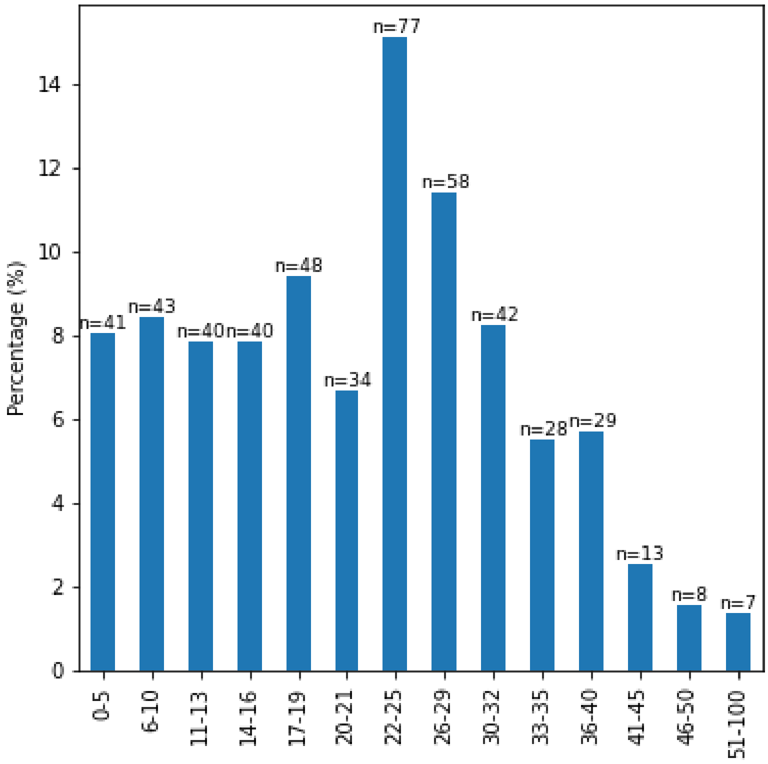


Figure 3 shows the distribution of promotion methods. Notably, only 10.4% of the entries (53 out of 508) are not promoted, meaning that nearly 90% are promoted through at least one channel. Among all promotion methods, advertising on Facebook/Meta is the most common way, accounting for 64.2% (326 out of 508 entries).

Popularity index

Figure 4: Distribution of popularity index categories



The popularity index of all the data points ranges from 0 to 60, with an average of 21 and a median of 22. The graph shows the category Popularity Category 1 on the x-axis. It can be seen that the dataset features songs with low, middle, and high Popularity Scores, having at least 20 observations in each category, below 40. In the higher categories (41-25, 46-50, and 51-100) there are fewer data points in each group.

Discovery Mode

As previously mentioned, the variable *IsThisSongOptedIntoSpotifyDiscoveryMode* was not included in the first version of the survey and was added only later. Therefore, this information is missing in 177 (34.8%) of 508 data points. Among the remaining data points, 84 (16.5% of the full dataset) are opted into Spotify Discovery Mode, while 247 (48.6% of the full dataset) are not.

Initially, we considered using this variable for modeling radio streams. However, due to the high proportion of missing values, including it would have required excluding more than one-third of the data points. Given this limitation, we decided not to use it in the modelling.

Exploration of Algorithmic Playlists

Release Radar

Table 2: Release Radar 28 days summary for different Popularity Index categories

PI_Category_2	*N*	*RR28Mean*	*RR28Min*	*RR28Max*	*N old*	*RR28Mean old*
0-5	7	4,4	0	15	5	4
6-10	10	59	0	226	10	N/A
11-15	17	144,8	0	649	7	21
16-20	36	217,5	0	1611	10	691,8
21-25	38	322,9	14	1227	7	1536,3
26-30	32	725,5	0	3846	18	2457,3
31-40	16	1952,9	10	14057	7	1101,5
41-100	3	2326,3	12	6041	2	N/A

First, we will have a look at the streams from Release Radar, listed in Table 2. The data here was reduced to data points that are no older than 35 days after the date of release, since this algorithmic playlist is only active for the first 5 weeks of a release. *N* indicates the number of data points in each category.

When we examine *RR28Mean*, we can see that even for the classes 0-5 and 6-10 there are positive entries for Release Radar streams (note that some artists are getting Release Radara streams from their followers from the first Friday after release). For higher Popularity Score classes, the average streams from Release Radar for the category 26-30 in our dataset is for example, 725,5. Looking at the *RR28Min* and *RR28Max* values, we can see that the numbers vary a lot.

At the right side of table 2, *RR28Mean old* shows the mean Release Radar Streams in the past 28 days, calculated from the data used for the first report. For the calculation, the exact same filtering has been applied so that the numbers are comparable.

N old are the sizes of the different Popularity Index categories. When comparing N and N old, it becomes clear that the new dataset has more data points. Even though the sample size in the PI categories is low, we can see that the groups 16-20, 21-25, and 26-30 show more streams in the old dataset compared to the new dataset. There is a tendency that Release Radar exposure has decreased, but for a thorough comparison, the data quality of the old dataset is not sufficient (low sample size, lack of data quality measures undertaken).

Discover Weekly

For the stats portrayed below, the data was reduced to only those songs that have been out for more than 35 days. As is publicly known, streams from Discover Weekly mostly appear after that time.

Table 3: Discover Weekly 28 days summary for different Popularity Index categories

PI_Category_2	N	DW28Mean	DW28Min	DW28Max
0-5	34	0	0	0
6-10	33	0	0	0
11-15	50	0	0	0
16-20	40	0	0	0
21-25	58	21,9	0	851
26-30	44	597,2	0	5203
31-40	65	1746,5	0	16973
41-100	25	1524,6	0	114

As displayed in Table 3, the Discover Weekly streams are mainly starting at the category 26-30 (in the category 21-25 was only one datapoint with 851 streams).
Having a Popularity Index of 31-40, the average amount of streams is 1746,5, ranging between 0 and 16973 for the past 28 days. Similar to the Release Radar Streams, having a high popularity index does not guarantee Discover Weekly streams, as we have songs in our data set that got 0 streams for each Popularity Index Category.

A comparison to the old dataset was desirable, but in that instance, the discover weekly stats were only collected in the late phase of data collection, and sothere is not enough data on Discover Weekly Streams in the old dataset to allow a decent comparison.

Radio

Table 4: Radio 28 and 7 days summary for different Popularity Index categories

PI_Category_2	N	Radio28Mean	Radio28Min	Radio28Max	Radio7Mean	Radio7Min	Radio7Max
0-5	41	16	0	361	4,4	0	136
6-10	43	78,3	0	498	15	0	87
11-15	67	256,7	0	1381	92,5	0	1151
16-20	76	502,3	8	3733	183	0	1391
21-25	96	1177,8	17	30803	304,6	0	1974
26-30	76	1987,1	196	8936	567,9	26	2868
31-40	81	7519,3	272	29997	2024,1	73	8114
41-100	28	78768,2	4277	457010	25368,4	718	148664

Table 4 shows both metrics for the 28-day period and the 7-day Period for radio. Let's focus on the 28-day period.

If we take a look at the 28-day average stream count for the 31-40 category, it is notable that Radio generates significantly more streams (7519,3) than Discover Weekly (1746,5) or Release Radar (1952,9). This ratio is similar in the other categories. This indicates that Radio has become the dominant algorithmic playlist on Spotify.

Algorithmic vs Non-Algorithmic Streams

Let's have a look at the distribution of algorithmic streams and non-algorithmic streams in our dataset. *AlgoStreams28* for example, was calculated by adding *RadioStreamsLast28Days*, *ReleaseRadarStreamsLast28Days* and *DiscoverWeeklyStreamsLast28Days* together.
Table 5 lists the respective means and percentages.

Table 5: Algorithmic and non-algorithmic streams and percentages by Popularity Index category

PI_Category_2	N	AlgoStreams28 DaysMean	NonAlgo Streams28	AlgoPc	NonAlgoPc

Table 5: Algorithmic and non-algorithmic streams and percentages by Popularity Index category

			DaysMean		
0-5	41	8	267	0,08	0,92
6-10	42	88	546	0,16	0,84
11-15	67	311	850	0,25	0,75
16-20	76	606	1617	0,26	0,74
21-25	95	1035	2946	0,28	0,72
26-30	77	2692	6302	0,31	0,69
31-40	81	9733	14885	0,4	0,6
41-100	29	78741	103867	0,43	0,57

We can see that for every category of the Popularity Index, the average number of non-algorithmic streams is higher than the average of algorithmic streams. The percentages show, that the higher the popularity score category, the higher the percentage of algorithmic streams. In the category *21-25*, 28% of the streams come from algorithms, in the category *31-40*, 40% of the streams are of an algorithmic nature.

As is commonly known, the popularity index is calculated using a time window of 6-8 weeks. When no new streams are coming in, the index will inevitably drop over time. Table 5 also shows us roughly how many streams from non-algorithmic sources have to come in so that the popularity index number will be maintained over time.

Exploration of Popularity Index

A main point of interest is to investigate what metrics influence the popularity score and how high these metrics need to be to achieve a certain score.
In the first popularity analysis (conducted by Andrew Southworth and Marc Schmieder in 2022), the results showed that the streams and listeners for the past 28 days were the main factor influencing the score. The first report is linked on the first page of this report.

Table 6 lists the new results along with the results from 2022.

Table 6: Average Streams and Listeners for Popularity Index Categories. A comparison of old and new data.

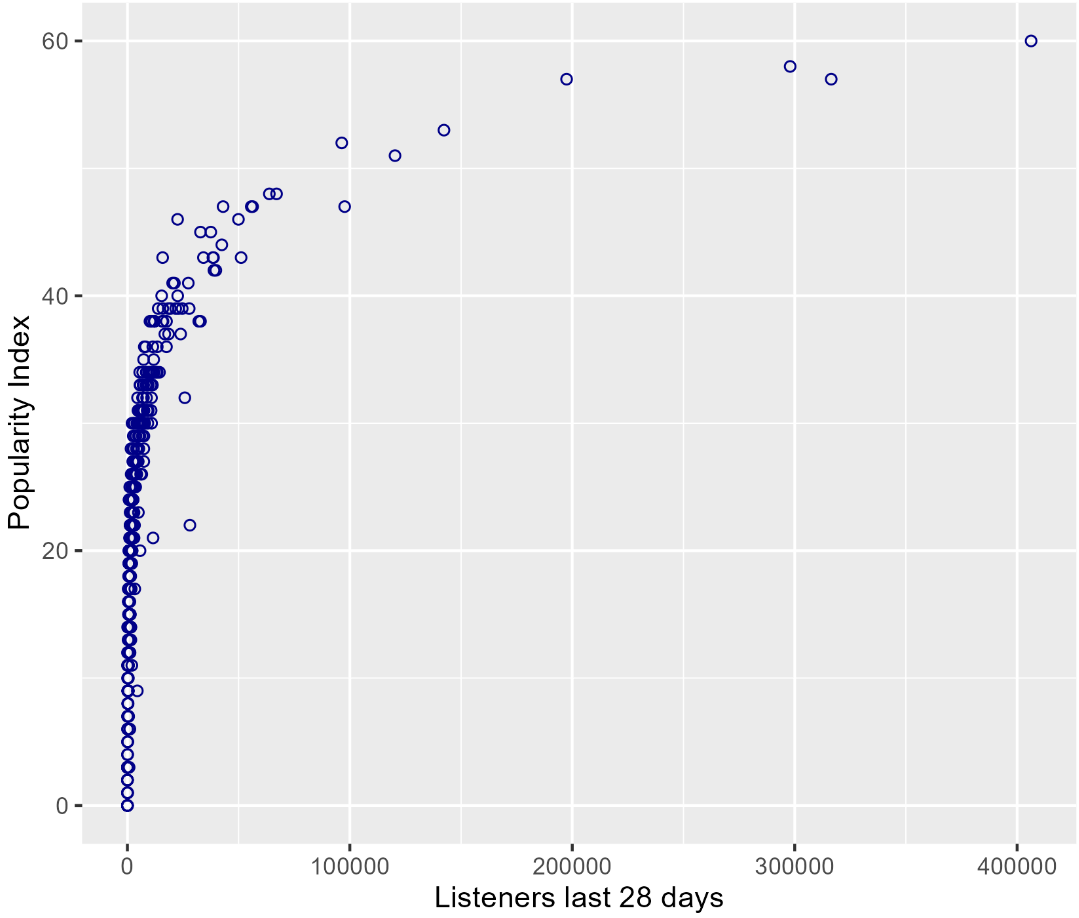
PI_Category	N	Streams28 Mean	Listeners 28Mean	PI Category Old	Streams28 Mean	Listeners28Mean
0-5	41	299,9	73,5			
6-10	43	819,9	367,8			
11-13	40	1464,5	529,1			
14-16	40	1414,2	782,4			
17-19	48	2079,6	1105,7			
20-21	34	3801,6	2094,6	20-22	2503,0	993,8
22-25	77	5074,9	2470,4			
26-29	58	8264,5	3933,1			
30-32	42	13295,3	7123,5	30-32	9217,8	4097,1
33-35	28	18807,7	9416,9			
36-40	29	38018,4	17325,9			
41-45	13	74132,3	33737,7	40-42	32185,8	13334,5
46-50	8	140218,4	57004,2			
51-100	7	508310,6	225300,6			

In this overview, the *PI_Category* (which is more granular) was used to match the categories of the first report. The corresponding numbers were marked in colour to allow a quick comparison. For the *PI_Category* 20-21 (20-22 in the first report), the average amount of *Listeners28Mean* is 2094,6, compared to 993,8 in the first report. *Streams28Mean* are 3801,6 and 2503,0 respectively. Assuming that those variables are the most influential, this means that in 2025, songs have to reach roughly double the number of listeners to get a popularity score of 20-22. Looking at category 30-32, we can see that the numbers increased as well. In the first report, average listeners and streams were 9217,8 and 7123,5, and for this dataset, they are 13295,3 and 7123,5.

Modelling of Popularity Index

In this chapter, we construct a statistical model that explains the Popularity Index as good as possible. The graphic below shows a plot of *ListenersLast28Days* (x-axis) vs *Popularity Index* (y-axis).

Figure 5: Popularity Index vs ListenersLast28days



We observe a clear dependence (possibly logarithmic) on the 2 variables and can suspect that the *Popularity Index* is related to *ListenersLast28Days* in the form. The same graphical relation has been seen with *StreamsLast28Days*, *SavesLast28Days*, *PlaylistAddsLast28Days*, as well as those metrics regarding the last 7 days.

After trying all the variables in the dataset in various configurations (listing them would be beyond the scope of this report), the following linear regression model was selected, as it is among the best-performing models and also the simplest one.

$$y = \beta_0 + \beta_1 \cdot log(x_1 + 1) \qquad (1)$$

Here y is the Popularity Index, x_1 *ListenersLast28Days,* and β_0, β_1 the regression coefficients.
The model in (1) has been applied to the dataset of 508 data points and yielded a $R^2 = 0.91$. The coefficients are $\beta_0 = -21.71$, $\beta_1 = 5.90$.

This metric scales between 0 and 1. In simple terms, that means that the model explains 91% of the target variable.

We apply the *MeanAD* as a preferred metric of how good this model performs. This is the mean absolute deviation of our predicted values to the true values in the dataset, which means in simple terms, how many *Popularity Index* points our model is wrong on average.

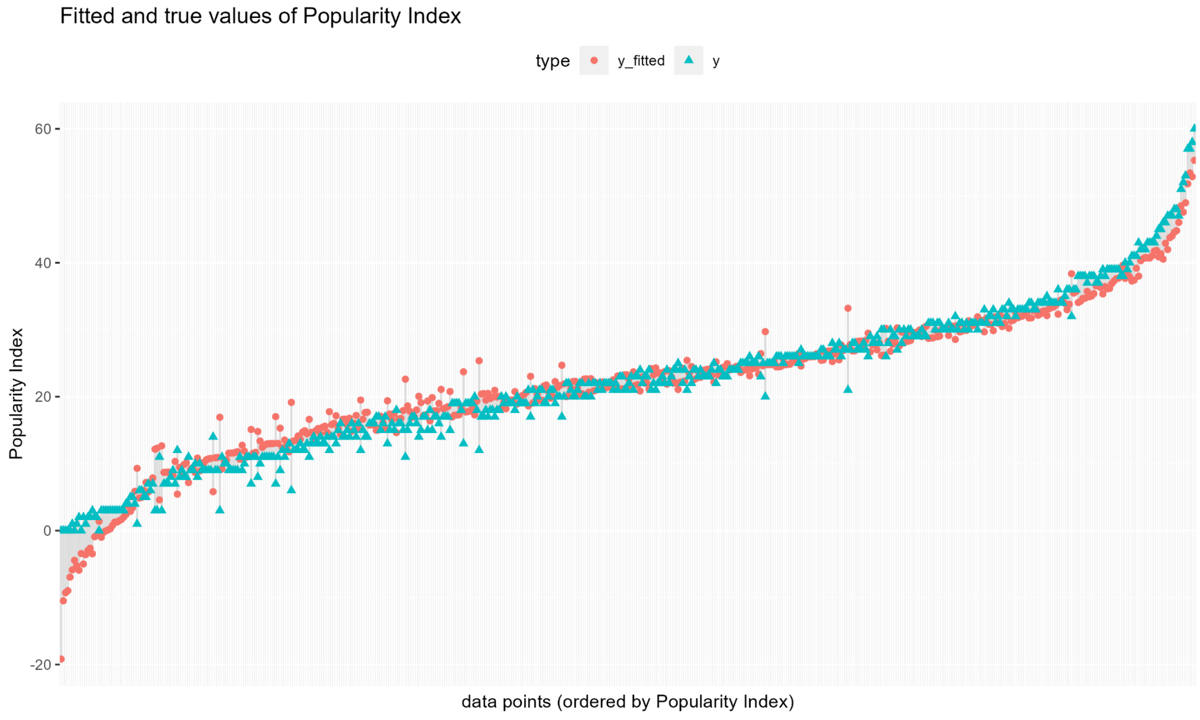
$$MeanAD = 2.48$$

This means that with our best model/formula, we can predict the Popularity Score knowing only *ListenersLast28Days* and only be about 2.48 points wrong on average.

Now, this model can surely be improved when using a more complex model like a non-linear model or a tree algorithm like *xgboost* (this was used in the first Spotify Algorithm Analysis and was able to predict the *Popularity Index* more accurately). For this report, we wanted to use a simple model so that the results are transparent and explainable.

Figure 6 shows a plot of the predicted *Popularity Index* values compared to the true *Popularity Index* values.

Figure 6: predicted and true values of Popularity Index



We can observe that the general function/curve has been well approached by our model (1).

As there were no restrictions implemented, some of the predictions on the left are negative. If we manually set those predictions to 0, we get an improved $MeanAD = 2.28$. For a better-suited model, a non-linear model with constraints could be implemented.

As our model performs well on this dataset, we can assume that Spotify's exact formula for the *Popularity Index* will most likely be in a form similar to formula *(1)*.

Can we achieve an $MeanAD$ of 0 using a community-collected dataset? The answer here is no, since the data is always a snapshot in time, both on *Spotify for Artists* and *Spotify for Developers*. When collecting this kind of data manually, there is always a time lag between the two sources. Looking at Figure 6, one can observe that some variability is just part of the dataset.

Knowing that only with the knowledge of *ListenersLast28Days*, we can predict the *Popularity Index* pretty accurately, we can use the numbers in Table 6 to know how many listeners we have roughly to acquire to achieve a certain Popularity Score.

A final note on this chapter: It is common to evaluate model performance on a hold-out data set. We wanted every data point to contribute to the model, and therefore, this was not applied here. A possible consequence could be that the models in this chapter and the chapter *Modelling of Radio* could be slightly overfit.

Exploration of Radio 7 Days

Tables 2, 3 and 4 have shown that Radio is the most dominant playlist nowadays. In the next chapter, we will construct a model to predict RadioStreams7Days.

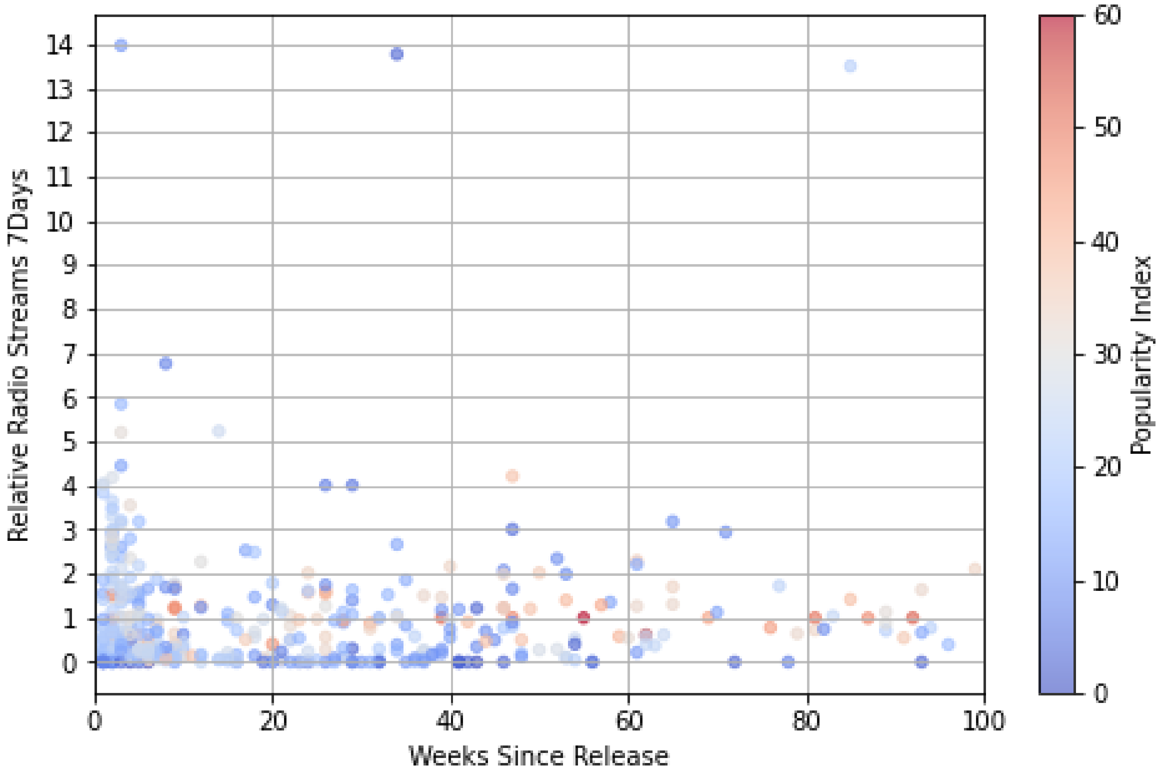
Some artists in the community have reported that they get a Radio boost in the first 4 weeks, and then the streams from that source decrease. It is of interest to examine whether we can see patterns in the data over time.

Table 4 shows that the amount of *RadioStreams28Days* varies a lot. The same applies to *RadioStreams7Days*, so showing them in one graphic would not be visually optimal.

A new variable *RelativeRadioStreams7Days* is calculated by dividing *RadioStreams7Days* by the average *RadioStreams7Days* for all the data points that have the same Popularity Index. We will learn later that the streams from Radio are influenced by the Popularity Index, and with increasing Popularity Index, there is generally an increased amount of streams from Radio. Since *RelativeRadioStreams7Days* are relative to the *Popularity Index,* they can be compared and visualized in one graphic.

Figure 7 shows *RelativeRadioStreams7Days* for the first 100 weeks after release.

Figure 7: Relative Radio Streams 7 days vs weeks since release

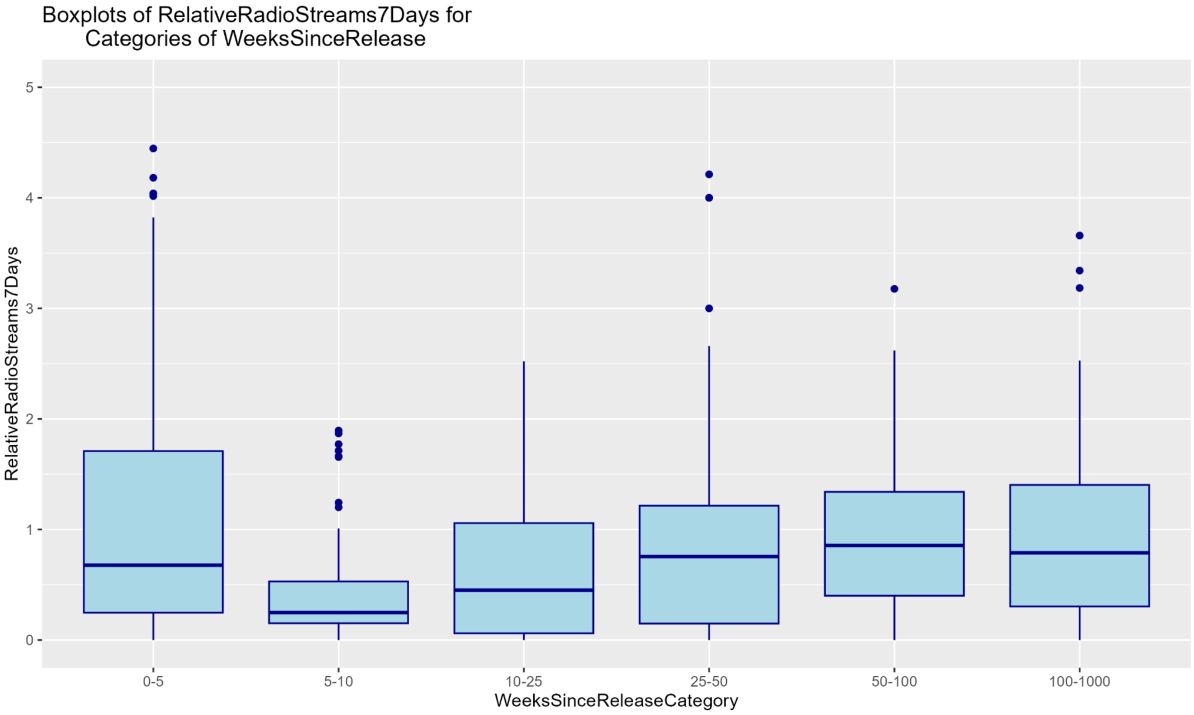


RelativeRadioStreams7Days can be interpreted as follows. A value of 1 means that the song/data points has scored the average amount of streams inside the group of songs with the same *Popularity Index*. A value of 0 indicates 0 streams. A value lower than 1 means a below-average performance on Radio, and a value over 1 means an above-average performance within its group.

We can also see, that as time progresses, there are always songs that score 0 streams on the radio and at the same time there are songs that get average or above average Radio exposure, even 1 year or longer after release.

It can be observed that there is a large number of data points in the first few weeks, and many of them have a *RelativeRadioStreams7Days* higher than 2. This visual impression suggests that the number of streams is comparatively higher in the first weeks. Figure 8 shows boxplots for groups of weeks after release.

Figure 8: Boxplots for Relative Radio Streams 7 days by weeks since release category



The lines in the middle of the boxes are the median *RelativeRadioStreams7Days* for the groups. The biggest difference regarding the median of 2 categories is between the 0-5 and 5-10 weeks categories. We can see a decrease in Radio streams after the first 5 weeks of release. Afterwards, the medians for the groups do increase again. This shows that even a long time after release, radio can yield good exposure.

Modelling of Radio

In this section, we are creating a model with the aim of explaining the target variable *RadioStreams7Days* as accurately as possible. In this matter, we tried including almost all the variables in the dataset. The following variables were at one point part of the configuration but did not show any significant influence. For example,

WhatGenreOfMusicMostAccuratelyDescribesThisArtistProject, WhatIsTheDominantWayThisArtistHasPromotedThisSong, DaysSinceRelease, ReleaseConsistency, HowManySongsHasThisArtistEverReleased, AreYouIndependentOrSignedToALabel did not show a significant influence.

IsThisSongOptedIntoSpotifyDiscoveryMode was significant, but we decided to not include it in the model (reasoning in chapter: Overview of the dataset). Other variables like *StreamsLast7Days* cannot be included in the model because they are directly related and therefore highly correlated with our target variable.

After trying out many model configurations (to list them would be beyond the scope of this report), the following model equation yielded the best result:

$$log(y + 1) = \beta_0 + \beta_1 \cdot log(x_1 + 1) + \beta_2 \cdot x_1 \qquad (2)$$

Here y is *RadioStreams7Days* , x_1 *Popularity Index* and $\beta_0, \beta_1, \beta_2,$ the regression coefficients. In this equation, *Popularity Index* is modelled as a logarithmic and linear dependent variable on a logarithmically transformed target variable.

The model in (2) has been applied to the dataset of 508 data points and yielded a $R^2 = 0.74$. This metric scales between 0 and 1. In simple terms, that means that the model explains 74% of the logarithmically transformed target variable. The regression coefficients are $\beta_0 = -0.41, \beta_1 = 0.57, \beta_2 = 0.16.$

This result is hard to interpret because in practice, we need to know how accurately we can predict the number of Radio streams (and not the logarithmically transformed).

To access this information, we transform the predictions of the model (y_fitted) back to the original scale. Now, we can see how good the model predictions are when we compare them to the true values. The mean absolute deviation is

$$MeanAD = 989.$$

Which means that our model is on average 989 streams wrong. However, a problem with using a metric like the *MeanAD* (used in chapter Modelling of Popularity Index) for evaluating this models performance is the following: If the prediction is 1200 streams and the true streams are 200 (bad prediction), the *MeanAD* punishes this in the same way as if the prediction is 120000 streams and the true streams are 119000 (good prediction). What we desire here is a metric that weights data points with high Radio Streams differently than data points with low Radio Streams. An easy, robust metric is the *MAD*, the median absolute deviation.
The used model achieves:

$$MAD = 117.$$

This means that with only knowing the Popularity Score, our model can predict the *RadioStreams7Days* $\pm$ 117 - with a median deviation.

This is still difficult to interpret, as knowing what the median deviation is for all observations does not give us a good estimate of how far a prediction will be off for a concrete example. To be able to tell how much the prediction will deviate from the true value on average, we created the following metric:

$$mean\ relative\ deviation\ (MRD) = mean(\frac{|\widehat{y} - y|}{y + z}|)$$

z is a constant that was added to counter the weight of data points with Radio streams close to zero. We picked $z = 50$.

MRD is similar to a mean percentage deviation, with an adjustment to reduce the weight on errors regarding data points with few streams. If the prediction matches the true value perfectly, *MRD* is 0, and when the prediction is far off, the metric is 1 or higher. But let us make a few examples to understand this metric.

If $y = 20, \widehat{y} = 40$, $MRD = \frac{|40 - 20|}{20 + 50} \sim 0.28$

If $y = 200, \widehat{y} = 400$, $MRD = \frac{|400 - 200|}{200 + 50} = 0.8$

If $y = 2000, \widehat{y} = 1800$, $MRD = \frac{|1800 - 2000|}{2000 + 50} \sim 0.1$

If $y = 1000, \widehat{y} = 1500$, $MRD = \frac{|1500 - 1000|}{1000 + 50} \sim 0.47$

Our model achieves $MRD = 0.73$, which can be interpreted that the models predictions deviate on average by around 73% from the true value.

How does that look visually? Figures 9, 10, and 11 show the true *RadioStreams7Days* and the models' predictions for the 507 data points, which were divided into 3 groups. *Popularity Index* smaller than 25, between 25 and 40, and greater than 40. The reason for the creation of the groups is a better visualization. If all data points were displayed in one figure, one would not see what is happening for the data points with a low number of radio streams.

Figure 9: Fitted and true values of RadioStreamsLast7Days for data points with Popularity Index <= 25

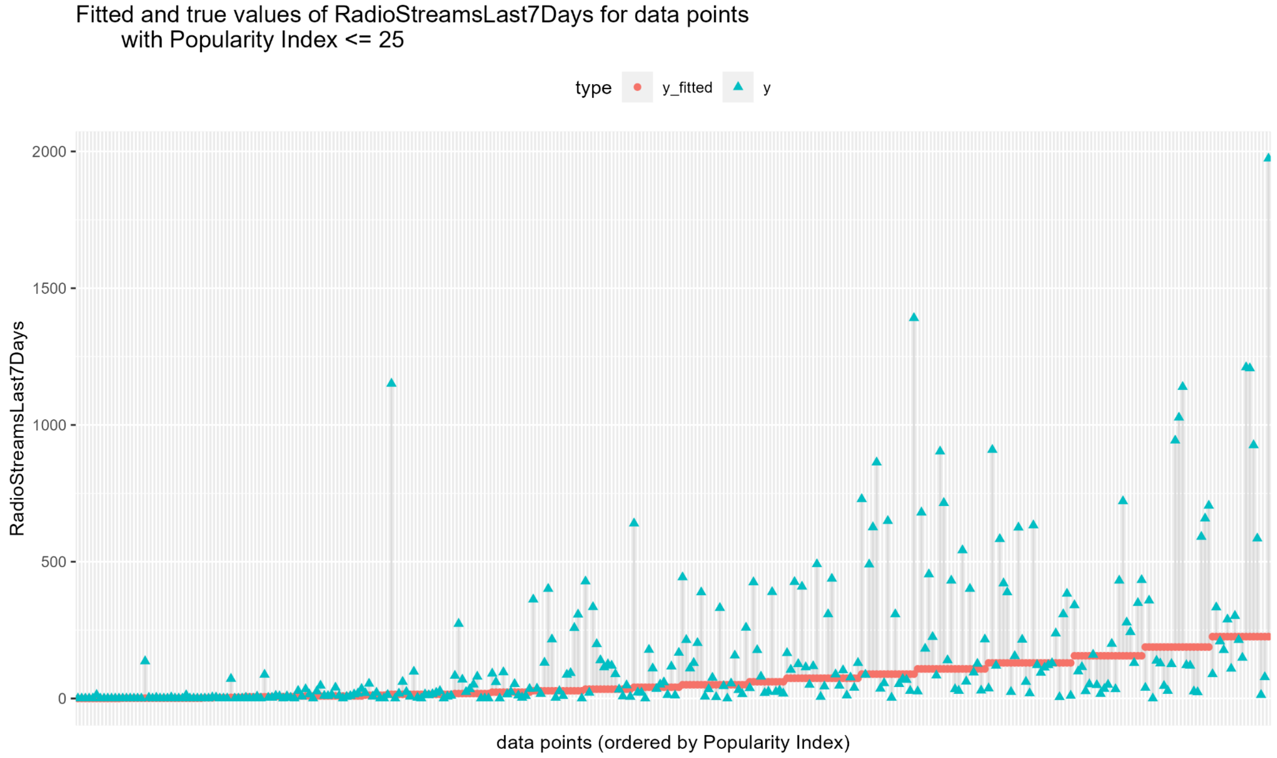


Figure 10: Fitted and true values of RadioStreamsLast7Days for data points with Popularity Index between 25 and 40

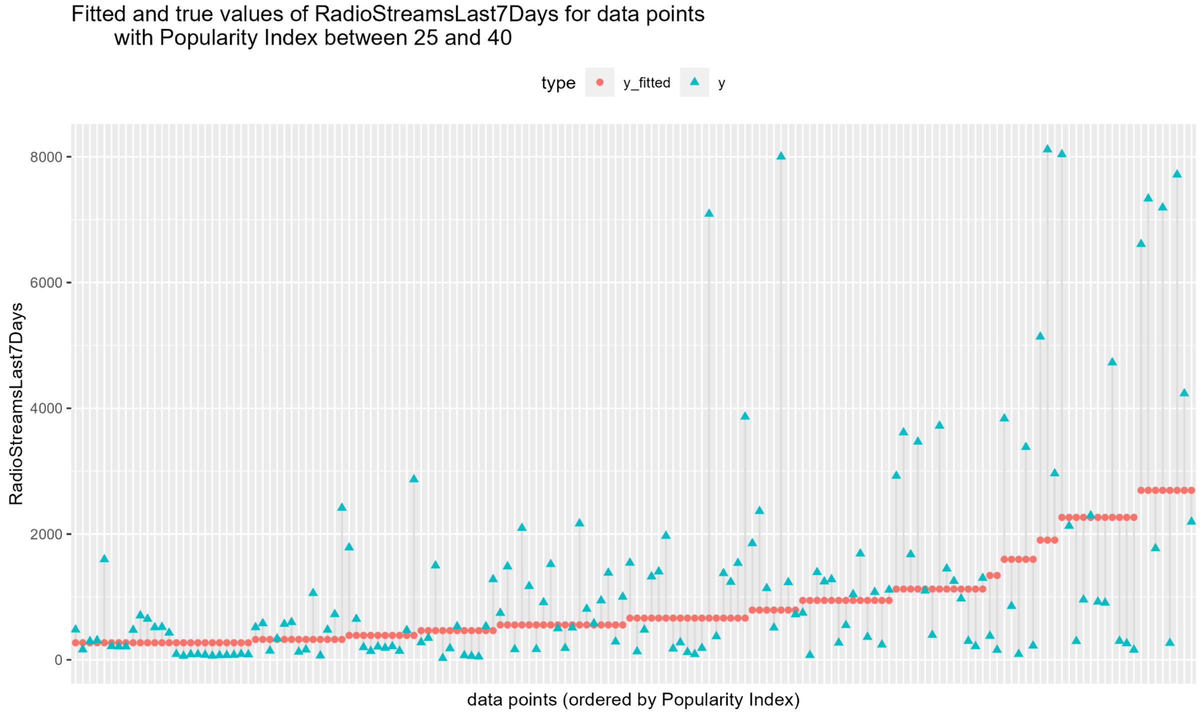
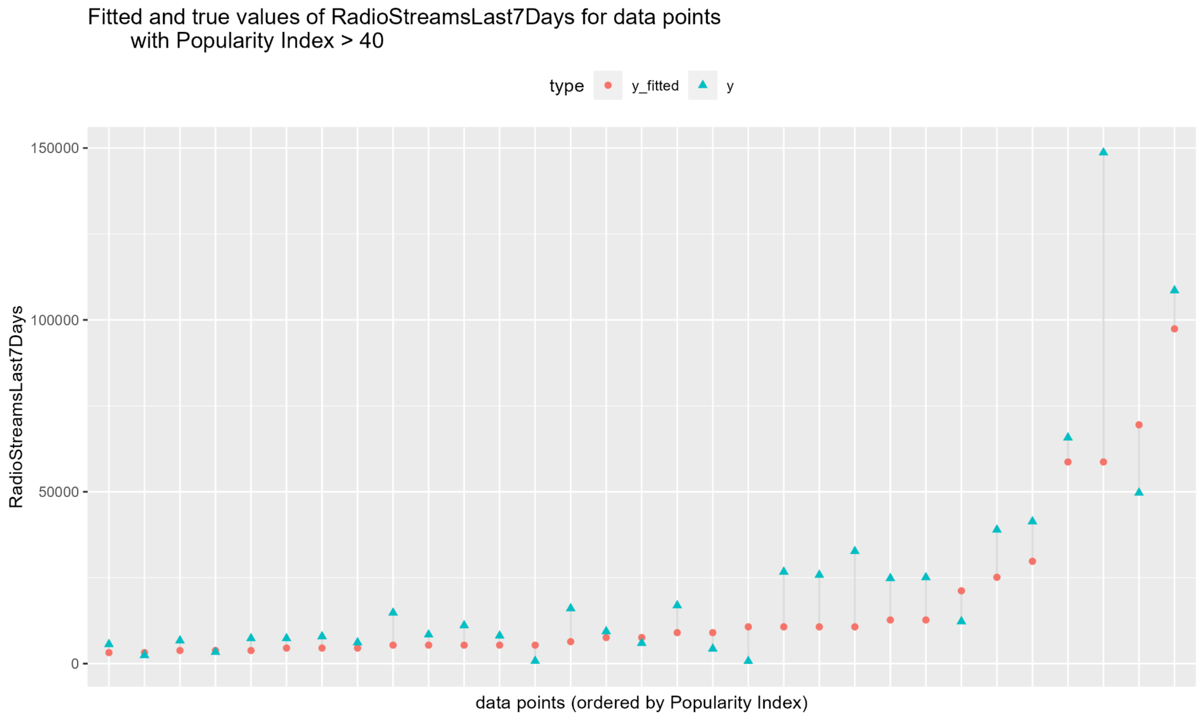


Figure 11: Fitted and true values of RadioStreamsLast7Days for data points with Popularity Index > 40



The model (2) covers the main tendency of the radio streams, just taking in the *Popularity Index* as a dependent variable. The rule seems to be, the higher the popularity, the higher the amount of radio streams an artist gets for their songs. We can also see, that for each group, there are always songs that get 0 or close to 0 streams on Radio, despite their popularity. Looking at table 4 shows that the minimum value for the shown *Popularity Index* classes is always zero or close to zero.

Using the average, minimum, and maximum values from Table 4 for *RadioStreams7Days* and *RadioStreams28days*, we can have a good estimate of what amount of radio streams to expect and if the song is performing over-average or under-average.

The model is far from perfect, and there is a lot of variability in the *RadioStreams7Days* that the model (2) can not explain.

Having tried out all the variables in the dataset in various configurations, we can confidently say that the information that could account for the unexplained variability in the data is not in the dataset and not accessible with *Sport for Artists.* In the next chapter, we discuss what those unknown data sources could be and why a model can never be perfect.

Discussion of Radio Modeling

This section revolves around my (Marc Schmieder) personal options and theories. Let us talk about an example Artist Z. Z's Radio streams can come from one of 2 sources. Autoplay and Song/Album/Artist Radio. How many streams Z gets on their on a specific time period depends on a few factors:

- The *Popularity Index* of their songs
- How many people generally listen on Spotify during that time period?
- On which and how many Song/Album/Artist Radios is Z generally playing or ranked?
- How many people listen to the songs/albums/artists on which Z is ranked during that time period
- How many new songs were released from artists on which Z is ranked or associated (a befriended artist told me that when another associated artist dropped a new EP, the befriended artist's radio streams saw a huge increase).

Further, I think that it matters that the artists that Z is associated with are creating similar music so that their audience will like the musical catalog of both artists. If the algorithmic association makes sense, Z will probably be ranked higher on the other artists radio and will be associated with more similar artists. When the algorithmic association does not make sense, the opposite could happen.

Part 3 of the Spotify Algorithm Analysis

With your help, we can continue this investigation and conduct Part 3 of the analysis.

In the process of conducting the data analysis for this report, one big remark was that in most cases, we only had one or a few data snapshots per song/artist (meaning data from one song on one certain day). If we had a whole sequence (data for each day of a release) of data points, there would be much more knowledge to gain.

The following data could be integrated additionally on Part 3 of the Spotify Algorithm Analysis, as we suspect that it has an influence on the amount of Radio Streams:

- The number of collaborations that an artist has done
- 1-5 Scale on how well the artist's radio matches their genre
- 1-5 Scale on how well the artist's Fans also like matches their genre

If you are reading this and want to help and provide your data, we want to encourage you to reach out to Andrew or Marc via the contact info below:

Marc Schmieder ✉ marc.schmieder01@gmail.com 🌐 www.instagram.com/marcschmiedermusic
Andrew Southworth ✉ andrew@generastudios.com 🌐www.andrewsouthworth.com