Do Popular Songs Endure?

DSF Project Final Report

12 December 2018

1 Problem & Approach

1.1 Problem

The main aim of the project is to find the endurance of popular songs, i.e. to formulate if songs which were popular at the time of release actually retain their popularity over time or whether some other songs which were relatively less popular at that time eclipse the former over time. The popularity of a song can be measured through various methods such as ranking (Billboard, Rolling Stones, etc.), or quantifiable parameters like number of plays (on radios, from Last.fm, etc.) and commercial success. We want to understand what are the ingredients that go into determining whether a song will make it through the barrier of time and remain as relevant as it was during the time of its release.

1.2 Approach

We solved this problem by decomposing the problem statement into the following parts.

- **Predict Current Spotify Popularity**
  Predict the current spotify popularity of songs as a function of popularity indicators like billboard features (best rank, weeks in billboard, maximum span, etc) and δy i.e the number of years from date of release of the song till now. The motivation behind using just the billboard features is that we want to predict the current popularity of song given its initial popularity. This would in turn help us determine the songs that beat expectations and endured longer than it should have or songs that under performed in comparison to what we would expect. We did not use any current popularity indicators like radio playcounts, youtube view counts, etc as that will be highly correlated to the actual popularity of the song and will not capture outliers (aka songs which were popular in their heyday but aren't anymore and vice-versa). We used a Regression Model for this purpose.

- **Outlier Detection**
  Once the model for predicting current popularity is trained we predict the expected popularity of all songs and find out the outliers i.e. the songs with maximum variance from the expected popularity. Among them the songs with positive errors are over performing songs while those with negative errors are under performing songs. Based on the popularity trend of songs which featured at least once in the top of billboard rankings , we observed although it has a decreasing trend with δy its not exactly a linear function, but rather a curve. So the errors will be high for all songs where the regression line is far from the curve. For this reason we find out the average error and the standard deviation of errors for each year. We determine the outliers as those whose error are more than 2.5 times standard deviation away from the mean error for that year.

- **Determining Pattern**
  Once we have determined the outliers we classify them as over performing and under performing songs. Now we want to determine the common pattern in these songs. We model this as a classification problem, with the features being song metadata features like acoustic features,lyric features and other features like movie feature counts, award wins, etc. After we train the model on the outliers we found out the most important features determined by the model.

2 Data Collection

Datasets used:

- **Billboard.com**- We used the billboard paginated APIs to query their dataset and fetch the data. We were able to get around 27000 unique songs that featured on the weekly charts within the top 100 rank.
- **last.fm** - Using last.fm API, we got the playcount and no. of listeners for each unique song in our dataset. For each unique song, we had to make sure we were fetching aggregated results as last.fm scrobbles can be linked to a plethora of streaming devices, all with disparate naming conventions.
- **Spotify** - We got track information from Spotify, including audio features of the song like duration, dance-ability, energy, instrumentalness, liveness, etc., as well as information about the artist, album, release date.
- **Lyrics** - We used an amalgamation of lyric websites such as Musixmatch, AZlyrics, Genius to gather lyrics data for the unique songs in our dat sets. We were able to get lyrics data for approximately 20,000 songs.
- **Grammy** - We have scraped data from Grammy awards website for all the winners since the start of the awards.
- **Movie/TV Feature** - We have used data from Whatsong to get which movies or tv shows have featured the song.

3 Features

In this section, we will detail the attributes we have computed from our sources of data above and used in our modelling endeavors so far.

We have used each source of data and computed more features based on the raw data we scraped from the websites. We used a subset of the features for our Regression Model and the rest for our Classification Model.

3.1 Regression Model Features

3.1.1 Billboard Features

artist	weeks_in_billboard_first_year
title	weeks_in_billboard_twenty_years
album_name	weeks_in_top_fifty
average_rank_first_year	weeks_in_top_fifty_first_year
average_rank_in_billboard	weeks_in_top_fifty_twenty_years
average_rank_twenty_years	weeks_in_top_ten
best_rank	weeks_in_top_ten_first_year
best_rank_in_first_year	weeks_in_top_ten_twenty_years
best_rank_in_twenty_years	weeks_in_top_thirty
first_date	weeks_in_top_thirty_first_year
first_year_rank_change	weeks_in_top_thirty_twenty_years
last_date_in_billboard	worst_rank
max_run	worst_rank_first_year
max_run_first_year	worst_rank_twenty_year
max_run_twenty_year	year_range_billboard
release_year	weeks_in_billboard
twenty_year_rank_change	

Figure 1:

Rank Features:These features are related to the rank of the song as it appeared on the Billboard Top 100 chart. We have taken different time spans to aggregate these features, so we can see the decay of the popularity. ***Span Features:*** The span features capture the lifespan of the song on the charts. The spans are uninterrupted runs that the song has on the chart, i.e. how many weeks the song appeared on the chart without dropping off from the rankings. Again, we have calculated span for different time periods so we can capture the change in the span, and the decay in how long the song can secure a rank in the charts without dropping off in relevancy. δy: The number of years from the year of release till now

3.2 Classification Model Features

3.2.1 Song Features

Metadata Features: These features capture the metadata about the song, including the album it featured on, the artist features related to the song, as well as which label it was released under, and its genre.

Acoustic Features: These features capture the intrinsic features of the song, such as the audio features, as well as the energy and mood of the song.
Lyric Features:

- **Sentiment** - This gives a sentiment score to the lyrics, with positive sentiment score, negative sentiment score as well as compounded sentiment which is the overall sentiment of the song lyrics.
- **f_k_grade** - The Flesch Kincaid readability indicates how difficult a passage in English is to understand, capturing mass reach. This is a grade formula in that a score of 9.3 means that a ninth grader would be able to read the document.
- **flesch index** - Captures the overall ease of reading a text. The maximum score is 121.22, and the lowest score is the most difficult to read. Negative scores are also possible.
- **fog index** - This is another metric to calculate the complexity of the lyrics.
- **difficult words** - The no. of difficult words in the lyrics.
- **num syllables** - The no. of syllables used in the lyrics.
- **num words** - The no. of words in the lyrics.
- **num dupes** - The no. of duplicate words in the lyrics.

Other Features:

- **movies_TV_feature_count**: The number of times the song has featured across across movies and tv series.
- **oscars_won** : The number of oscars won by the song.
- **artist_lifetime_grammy_achievement**: The number of lifetime achievement awards won by the artist.
- **artist_grammy_wins** : The number of grammies won by the artist
- **artist_grammy_nominations**: The number if grammy nominations received by the artist
- **days_before_charting**: The number of days from date of release to its first charting in billboard
- **age_Percentage_15_30**: The percentage of the demography representing people aged between 15-30 against the total population in US.
- **artist_popularity**: Popularity score of the artist

4 Models, Experiments and Results

In this section, we will discuss the regression models used, their results, classification models tried and their results.

4.1 Data Processing

Before training the data on the models defined below we did the following data preprocessings.

- Derived new features like max_run,first_year_rank_change, etc.
- Imputed missing data by the mean value of the column for that year. We applied this imputation only on columns with integer or float values.
- Simplified genres into their parent genre. For example replacing the following genres 'emo','permanent wave','british invasion','mellow gold','jam band' with rock. We replaced nan values of main_genre with 'other'

4.1.1 Brief overview of models used

- **Linear Regression Model** : It aims to fit a regression line through the true value of the dependant variable by minimizing the sum of square of residual errors.
- **Random Forest** : Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random Forest overcomes the problem of over-fitting of Decision Trees. In particular, trees that are grown very deep tend to learn highly irregular patterns: they overfit their training sets, i.e. have low bias, but very high variance. Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance. This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance in the final model. Random Forest Regressor and Classifier work on the above principle.
- **Gradient Boosting** : Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function. Like other boosting methods, gradient boosting combines weak "learners" into a single strong learner in an iterative fashion. XGBoost Classifier, LightGBM Classifier and LightGBM Regressor work on the above principle.

4.2 Regression Models

Main aim of first step of the project is to predict current spotify popularity of a song given it's billboard features in it's year of release. Hence, we used regression models to fit a curve to predict the current spotify popularity.

For our baseline implementation, we used Linear Regression to predict the current spotify popularity based on the features described in previous section. We found the over-performing and under-performing songs based on the error in prediction of particular songs. We did sniff test to actually check if those songs did over-perform or under-perform.

We then tried out other regression models to see if we can improve our predictions of over-performing and under-performing. We predicted using Random Forest Regressor and LightGBM Regressor. We found out that these models we doing a good job in fitting the original data and made good predictions when compared to Linear Regression model.

Model	RMSE
Linear Regression	13.096
Random Forest Regressor	12.879
Light GBM	12.879

Figure 2:

We found outliers by predicting the expected spotify popularity using LightGBM Regressor and finding the songs which have most error. The songs with highest positive error are classified to be over-performing and the songs with highest negative error are classified to be under-performing. Following picture shows the parameters to our LightGBM regressor.

```
lgbm_regressor = lgbm.train(params, train_set = train_set,
          early_stopping_rounds=500, verbose_eval=500, valid_sets=valid_set)
```

Figure 3:

4.3 Classification Models

Main aim of the second step is to find out what features of a song makes it over-perform and under-perform. Since, we have outliers (over-performing and under-performing songs) from the above regression model, we classified these outliers using various classifiers and found the feature importances for such classification.

We used Random Forest Classifier, XGBoost Classifier and LGBM Classifier to classify the data. Following are the results provided by various classifiers.

Model	Accuracy
Random Forest Classifier	90.79%
XGBoost Classifier	90.79%
LGBM Classifier	94.74%

Figure 4:

We chose LGBM Classifier based on above results and found out the feature importances. Following picture shows the parameters for our LGBM classifier.

```
LGBMClassifier(boosting_type='gbdt', class_weight=None, colsample_bytree=1.0,
        importance_type='split', learning_rate=0.1, max_depth=-1,
        min_child_samples=20, min_child_weight=0.001, min_split_gain=0.0,
        n_estimators=100, n_jobs=-1, num_leaves=31, objective='binary',
        random_state=None, reg_alpha=0.0, reg_lambda=0.0, silent=True,
        subsample=1.0, subsample_for_bin=200000, subsample_freq=0)
```

Figure 5:

Following plot shows the feature importance.

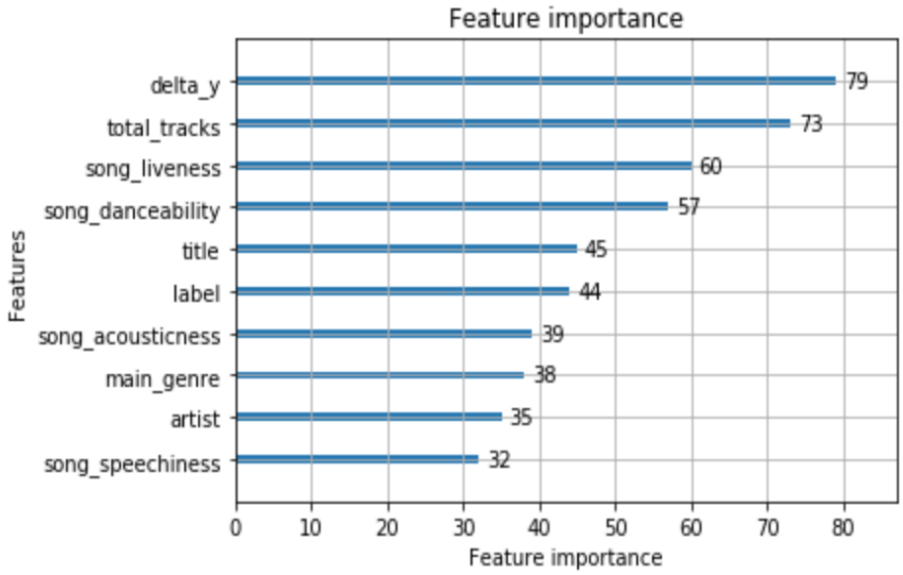
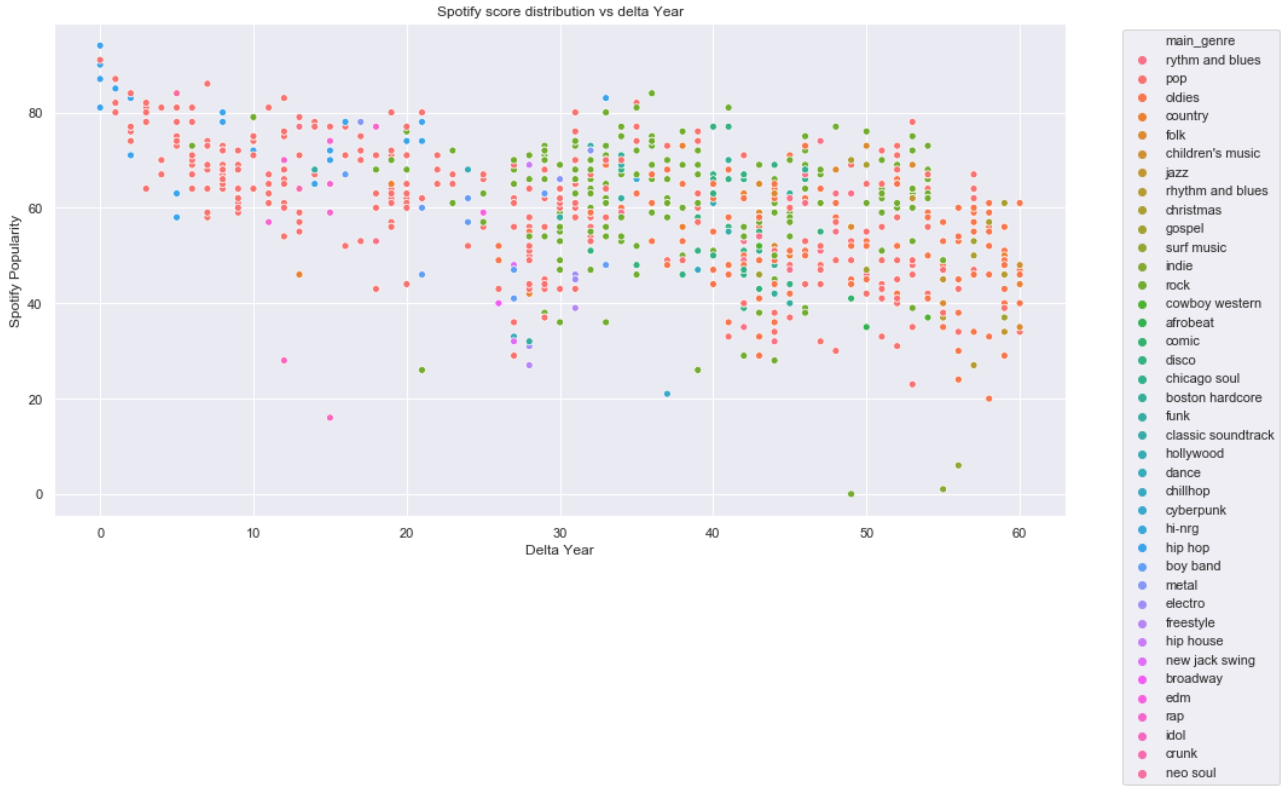


Figure 6:

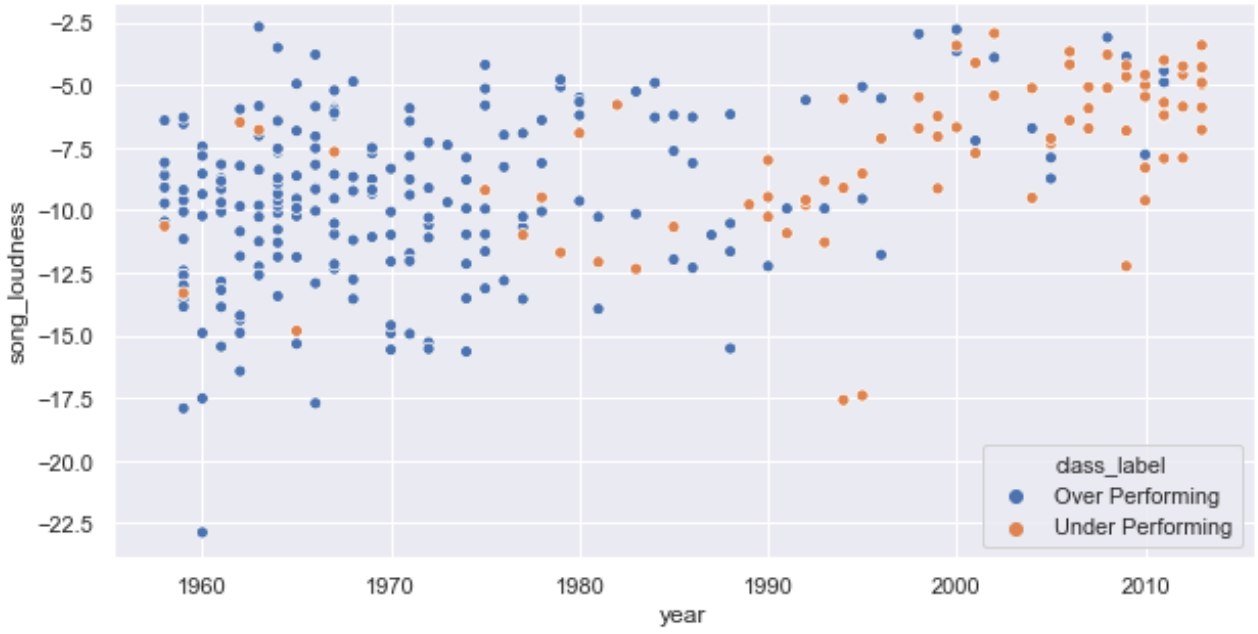
We can see that the most important feature is delta_y which is very obvious as evident from the decreasing trend in popularity with time as illustrated in below graphs. We see that main_genre also plays an important role as evident from the spread of different genres across time.

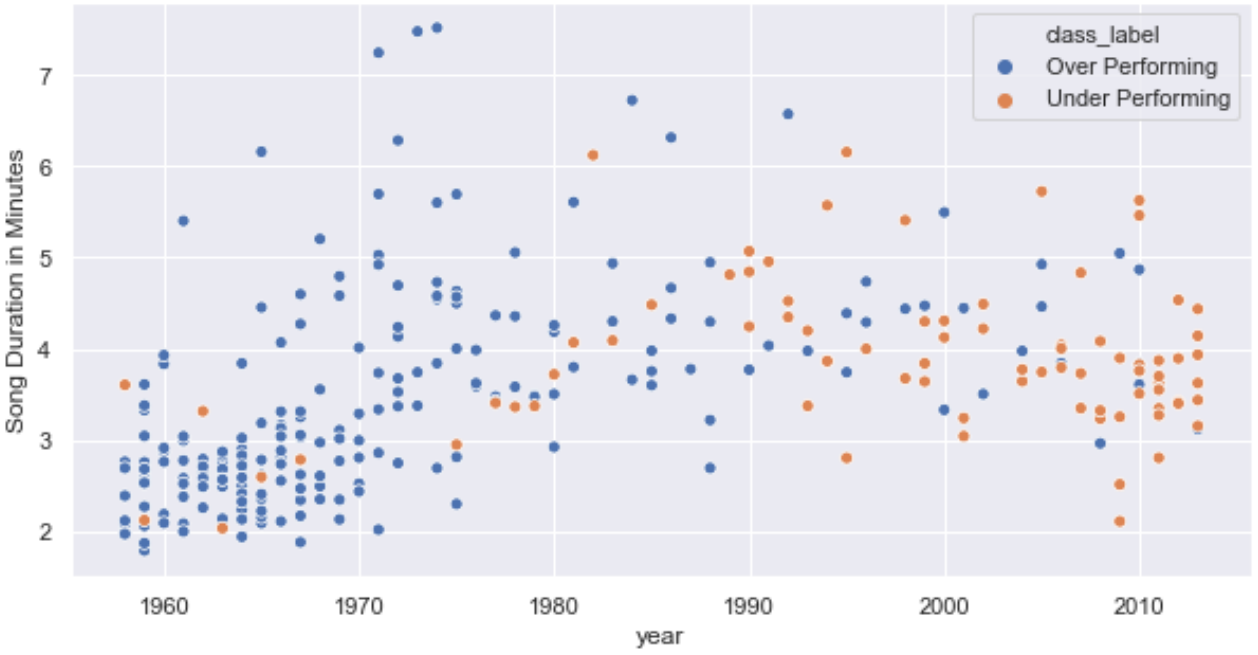
5 Result Analysis - What Makes A Song Overperform?

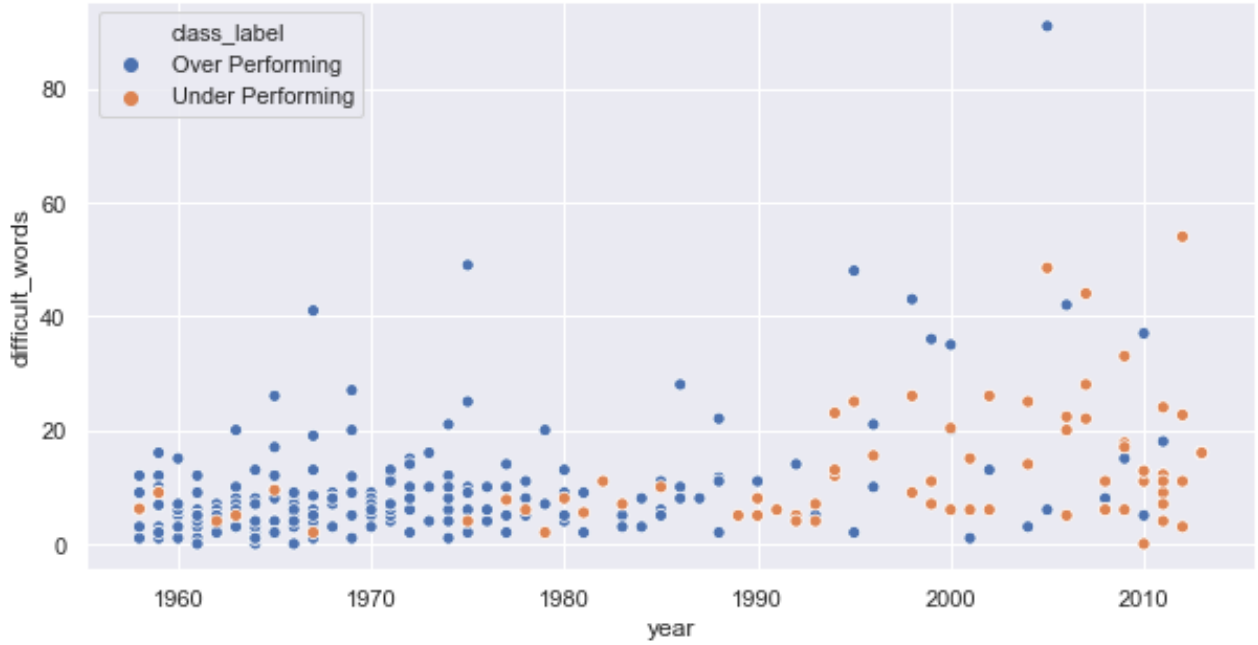
From common knowledge of music, we know that prevalent musical genres tend to change over time, as people's tastes change, and experimental sounds become mainstream and desire for newer sounds or genres take over. This is well-represented by our exploratory graph, which shows that the popular songs had a shift of genre over the range of years we are currently analyzing. We can see pop songs are more popular now, while genres like doo-wop, western, etc. which older generations used to prefer have fallen out of favor.



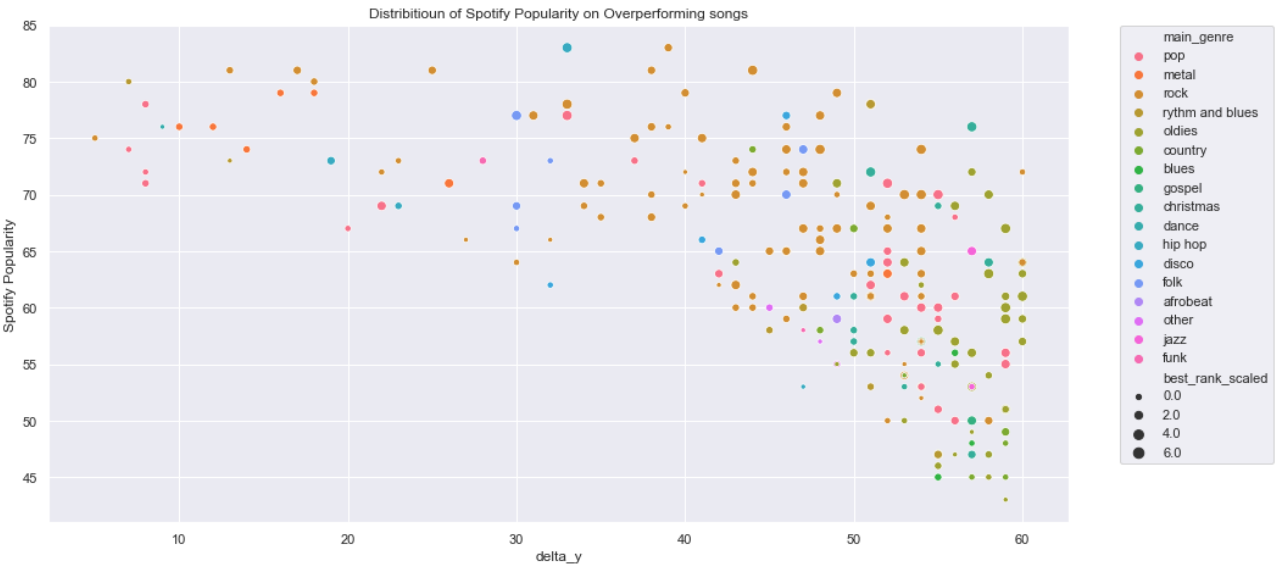
We also present an analysis of intrinsic features of a song versus whether they over-performed or under-performed, and what year they were charting on Billboard. This analysis helped us gain an initial idea of which features would impact our classification of whether a song was going to endure or not, and we used our models to validate these hypotheses.

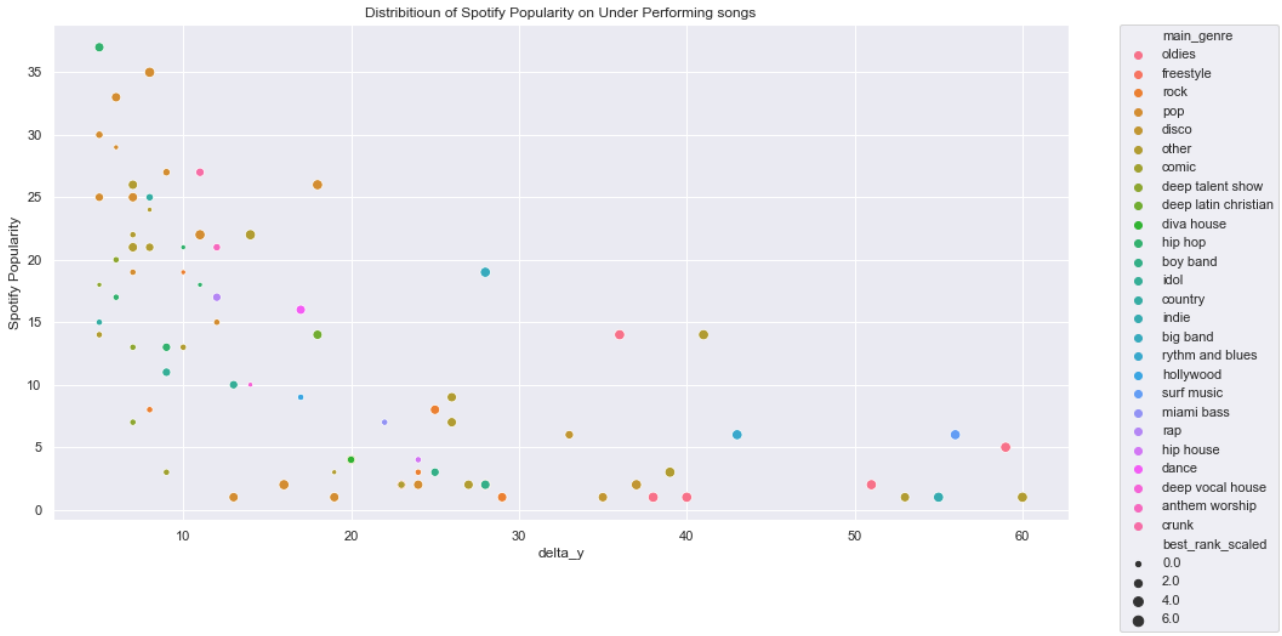

song_loudness
−2.5
−5.0
−7.5
−10.0
−12.5
−15.0
−17.5
−20.0
−22.5
1960
1970
1980
1990
2000
2010
year
dass_label
Over Performing
Under Performing


Song Duration in Minutes
7
6
5
4
3
2
1960
1970
1980
1990
2000
2010
year
dass_label
Over Performing
Under Performing

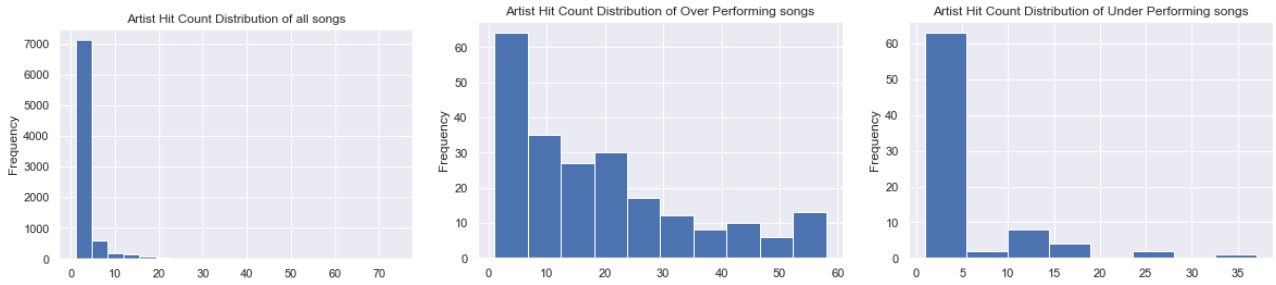


Apart from the trends analyzed over all the Billboard Top 100 ranking songs, and over the range of under and overperforming songs, we also isolated the overperforming and underperforming songs individually. We visualized some of the features to see whether the distribution of these features differed from overperforming to underperforming songs, so we could gain an understanding of which features could be important in deciding the endurance of a popular song. In these graphs, we also wanted to see the distribution of highly-ranked (i.e. extremely popular relative to other songs at time of release) for both of these categories, and how they were represented within the features. We found a wide distribution of genres in both these graphs, which implies that there is no linear relationship between main genre or initial popularity - it is rather a complex interplay of these features, combined with others, that can predict with confidence whether a song will endure.





A key insight was derived from the following graphs. An intuition for certain songs charting high on Billboard Top 100, but fading out of relevance over the years could be due to the fact that they were one-hit wonders. One-hit wonders are those songs where an artist is known only for one song, while the rest of their discography is considered mediocre, or simply without any notion of popularity or public favor. We tried to see whether this was the case for the underperforming and overperforming songs we had isolated. The first graph shows the distribution of hit counts, i.e. the no. of songs that have charted over the period that our data captures per artist. For all artists whose count for this factor is one, we call them one-hit wonders. Hence, we can see the spread of one-hit wonders over the entire data, and for our outliers.



In the first graph, we can see that the distribution is left-skewed to quite a large extent. This implies that most artists tend to chart only once on Billboard Top 100 (that is, with one song, which could appear for multiple weeks, or drop from the charts and appear again). However, the interesting insight appears once we isolate the overperforming and underperforming songs. We see that in overperforming songs, the data is still left-skewed, but now we have relatively more frequency distribution in the other bins. In contrast, underperforming songs have majority of the data in the left, implying that songs that tend to fade out of obscurity tend to be by one-hit wonder artists more often than not.

6 Sniff Test

6.1 Over Performing Songs

title	artist	Y_pred	Y_true	year	best_rank	weeks_in_billboard	average_rank_first_year	Movies_TV_feature_count	error
at last	Etta James	17.877457	72.0	1961.0	47.0	8.0	59.500000	10.0	54.122543
johnny b. goode	Chuck Berry	19.375593	72.0	1958.0	80.0	1.0	80.000000	0.0	52.624407
surfin	The Beach Boys	15.742658	68.0	1962.0	75.0	6.0	83.500000	0.0	52.257342
rock and roll all nite	KISS	23.175256	73.0	1975.0	68.0	6.0	77.666667	0.0	49.824744
tiny dancer	Elton John	26.785483	76.0	1972.0	41.0	7.0	54.428571	5.0	49.214517
pain in my heart	Otis Redding	19.871201	69.0	1963.0	61.0	11.0	79.454545	1.0	49.128799
highway to hell	AC/DC	34.116462	83.0	1979.0	47.0	10.0	62.100000	0.0	48.883538
starman	David Bowie	23.146782	72.0	1972.0	65.0	9.0	81.555556	2.0	48.853218
good times bad times	Led Zeppelin	21.579800	70.0	1969.0	80.0	4.0	84.750000	2.0	48.420200
my generation	The Who	19.913431	68.0	1966.0	74.0	5.0	80.200000	0.0	48.086569
jolene	Dolly Parton	26.071219	74.0	1974.0	60.0	8.0	71.875000	3.0	47.928781
run rudolph run	Chuck Berry	16.078602	64.0	1958.0	69.0	3.0	75.000000	0.0	47.921398
ain't no mountain high enough	Marvin Gaye & Tammi Terrell	30.830434	78.0	1967.0	19.0	12.0	35.250000	2.0	47.169566
back in black	AC/DC	33.854650	81.0	1980.0	37.0	15.0	57.666667	0.0	47.145350
big iron	Marty Robbins	17.048394	64.0	1960.0	26.0	10.0	48.100000	0.0	46.951606
don't stop me now	Queen	29.162053	76.0	1979.0	86.0	4.0	89.250000	0.0	46.837947
rebel rebel	David Bowie	24.902584	71.0	1974.0	64.0	8.0	75.125000	8.0	46.097416

The list shows the most over performing songs defined by the prediction error. As we can see the list contains some of the most iconic songs of the 20th century like *Johny B. Goode*,*Surfin*, *Tiny Dancer*, *My Generation*, *Jolene* and ofcourse *Dont stop me now* among others.
One observation is that their best rank in billboard was very average (i.e above 50) which indicates that they were only moderately popular at time of release. Probably the thing that helped them cement their position in pop culture was that they featured in some iconic movies that released at that time. We can see that from the number of times they have been featured in movies or TV series.

Another factor for popularity would be that these songs are from iconic artists that have held on to their popularity over the years (such as KISS, Elton John, Chuck Berry, Led Zeppelin). These artists have a cult-like fan following which grows with time and attract lots of followers who like niche music and non-mainstream artists.

6.1.1 Under Performing Songs

title	artist	Y_pred	Y_true	year	best_rank	weeks_in_billboard	average_rank_first_year	Movies_TV_feature_count	error
give me the night	George Benson	50.742481	1.0	1980.0	4.0	23.0	36.869565	0.0	-49.742481
dream about you/funky melody	Stevie B	49.722461	2.0	1995.0	29.0	23.0	44.130435	0.0	-47.722461
hands clean	Alanis Morissette	49.389196	2.0	2002.0	23.0	20.0	48.300000	0.0	-47.389196
across the universe	Various Artists	48.021807	1.0	2005.0	22.0	1.0	22.000000	0.0	-47.021807
what about us?	Brandy	48.624727	2.0	2002.0	7.0	18.0	35.277778	0.0	-46.624727
turn the beat around (from "the specialist")	Gloria Estefan	48.429403	2.0	1994.0	13.0	25.0	29.560000	0.0	-46.429403
almost doesn't count	Brandy	47.037164	1.0	1999.0	16.0	20.0	38.250000	0.0	-46.037164
a woman needs love (just like you do)	Ray Parker Jr. & Raydio	46.491666	2.0	1981.0	4.0	27.0	37.740741	0.0	-44.491666
ooh baby baby	Linda Ronstadt	44.893018	1.0	1978.0	7.0	16.0	28.562500	1.0	-43.893018
king of wishful thinking (from "pretty woman")	Go West	45.354159	2.0	1990.0	8.0	24.0	42.708333	0.0	-43.354159
a mover la colita	Artie The 1 Man Party	44.413629	2.0	1995.0	65.0	19.0	78.578947	0.0	-42.413629
ghetto cowboy	Mo Thugs Family Featuring Bone Thugs-N-Harmony	46.365866	4.0	1998.0	15.0	20.0	38.600000	0.0	-42.365866
i still can't get over loving you	Ray Parker Jr.	43.245327	1.0	1983.0	12.0	19.0	40.263158	0.0	-42.245327
you can't change that	Raydio	45.101440	3.0	1979.0	9.0	22.0	38.909091	0.0	-42.101440

The under-performing songs were quite popular at their time of release, as shown by the Billboard features, and hence given predictive scores that were as high. However, these songs failed to endure for various reasons. For example, *A Woman Needs Love (Just Like You Do)* was the last hit by an R&B band called Raydio, in their fourth album after which they broke up to differing opinions and career desires. Their lead singer Ray Parker Jr. eclipsed the success of the band, spawning six Top 40 hits in the 1980s, which implies that the group fell out of favor, probably due to the fact that the fans had migrated to his solo fanbase and hence, leading to the song not enduring.

Another aspect which we have discussed is captured by the son *Give Me The Night*, which was one of the last *disco* songs to gain a high rank on the Billboard Top 100 chart, before other genres grew in popularity (according to the Wikipedia page for the same song). This shows that while the song may have been popular during the *disco craze*, it probably was a function of that, and did not manage to stick once the public taste shifted genres.

6.2 Conclusion and Future Work

We have analyzed the trends in the music industry and tried to predict the endurance of a song based on its intrinsic and extrinsic feature, as a function of its initial popularity, and the time that has elapsed since it was released. The question *"What makes a song unforgettable?"* is a difficult one to answer, because there are a myriad of features that interact in complex ways that cannot fully be examined to determine whether a song sticks in the public consciousness. We have determined some of those features in the work above, but much more extensive analysis will benefit this exploration. If we could capture topics that songs mention, we could find correlations between a song's decline or rise in popularity with the shifting public mood, whether it is due to recession, globalization, the rise of the internet, etc.

Another feature that can help is details about the artist's life: sometimes artists get popular due to factors external to their musical careers - they may branch off into acting, they may change bands, they may feature on songs of other artists, or lead extravagant or shocking lives that fix them in the public's minds. These may also lead to some of their less popular songs suddenly coming to the forefront.

Finally, we can capture information unrelated to both the song itself and the artist. In this age of Youtube and Twitter, sometimes relevant artists share their favorite songs, or cover those songs, which may lead to a sudden spike in popularity of an old song. Especially as recent artists come from talent shows, and various online forums, old songs have a high probability of being covered and made famous due to these factors. Hence, we can add covers as a feature to indicate current popularity of a song.