CSE 519 MID PROGRESS PROJECT REPORT
Do Popular Songs Endure?

Objective:
To build a model that predicts the current popularity of a recording that appeared at a position *p* on a chart which varies with time. Building over this model, we will find songs that endure and the rationale behind it.

Introduction:
The Billboard Hot 100 chart captures each week's most popular songs across all genres. Rankings are determined by radio airplay audience impressions, which are measured by Nielsen Music's radio SoundScan tracking program, Nielsen Music's sales data, and streaming activity data from various online music sources. We identified the following datasets for our use: List of Billboard Hot 100 chart achievements and milestones, Wikipedia, Billboard Ranking, Spotify API (musical attributes, speechiness, danceability), Discogs API (Grammy Award), Gracenote API (genres, tempo classification, the mood of a song) and LastFM, music service application. Using the ranks from the billboard chart data and the Spotify API, we are trying to build a model that predicts the current and intermediate popularity of the songs.
Our aim till this point was to derive a model that gave us an idea about underperforming and overperforming songs and how we can relate it with metrics like Rank on Billboard, Current popularity, Playcount over the years, etc. We plan to build up on this model to finally come up with a model that returns information about the endurance of a popular/unpopular song from different time periods.

Dataset:
For now, we generated a dataset using the data from -

- Billboard chart data
- Spotify API data
- LastFM API data
- Gracenote API

The billboard chart data helps us to determine the release date of the song and the initial chart entry of a particular song. This billboard chart data provides the weekly rankings of each song. The Spotify API data gives us the popularity index of every song and the audio features of songs like tempo.The Gracenote API provided us the genre of the songs. Lastly, we scraped the data for the Billboard yearly top 100 using *BeautifulSoup*.

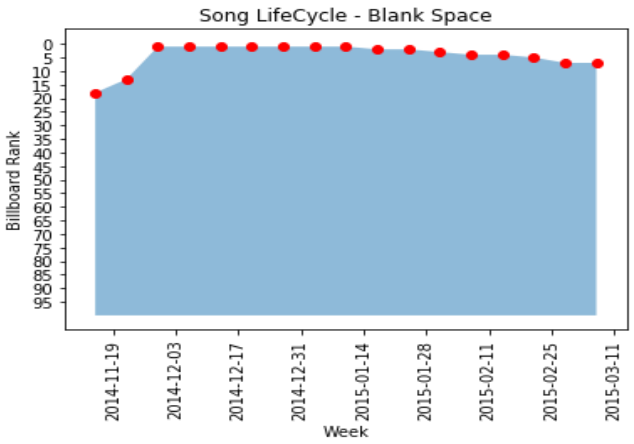
Data Preprocessing:
Using all the data we obtained from the different sites and APIs, we generated an entire dataset. Now, while merging these different sets of data, we encountered several issues and resolved them.

- One main issue was that there were recordings with the same name but different artists. This created a hindrance to distinguish between the recordings. Therefore, for ease of distinguishing, we created a new ID which is the song key. This was the concatenation of the recording name and artist name making it unique. So, even if it reappeared in the charts later, we would know that it is the same recording which became popular again.
- Another issue was abbreviations used in some dataset. For this, we used the harmonization of codes. We processed abbreviations like '*ft.*' and *'feat'* to '*featuring'* and '*&'* to *'and'*.

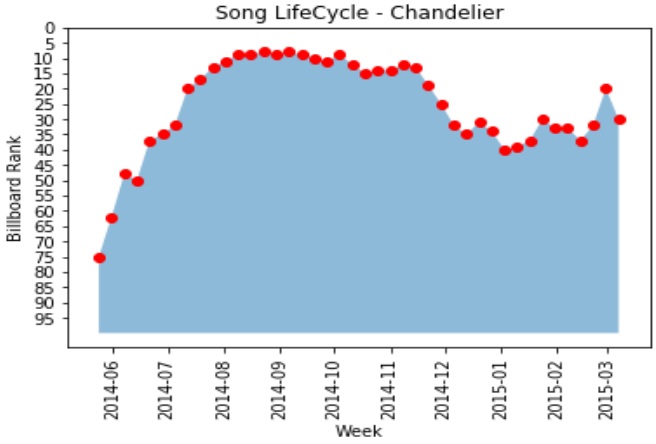
- There were also discrepancies like '*The Association'* and '*Association'* , *'5th Dimension'* and *'Fifth Dimension'* and differences in punctuations which we processed by using just a single value throughout all the datasets i.e. we standardized such values.
- Now, to merge different datasets, we applied a string matching algorithm on the song key. The song keys having a higher similarity value were mapped to each other.
- After all this, when we checked the data we were still left with a few inputs that were not getting merged correctly. We tried to remove these errors by manually mapping them.

One innovative metric that we came up with was Area Under Curve(AUC). AUC is basically the value that we get from calculating the area of the curve between ranks & weeks for a particular song. What we observed from calculating these AUC values was that a song that was high on charts but for a small duration of weeks actually had a low AUC compared to a mid ranking song that was on chart for a long time. This is what we think defines endurance, it's not about topping the charts, but being there for a long time which our calculated values conclude with utmost definition.

AUC - 2201



AUC - 3345



Now, in all from the billboard yearly top 100 we had data from 1960 to 2015. Our final dataset had columns like: SongID, Song, Artist, Rank, Year, Relative PlayCount, AUC, Current Popularity. Some of the data from the Spotify API had popularity index(Current Popularity) as null or 0, some values of PlayCount from LastFM as null or 0 and AUC as null or 0. So, we dropped these rows as they didn't contribute to the baseline model.

Features:

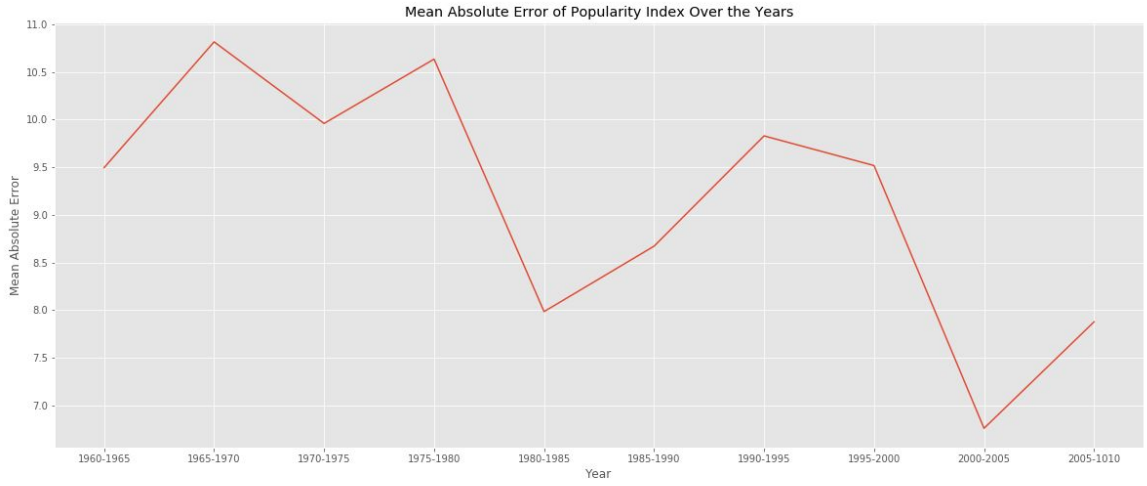
SongID - Concatenation of recording name and artist name
Song - Name of the recording
Artist - Name of the artist who recorded the song
Rank - The yearly rank of the song from the billboard chart
Year - The year in which the song was released
Relative PlayCount - Relatively encoding the PlayCount obtained from LastFM in each year.
AUC - Area under the curve for the number of weeks a recording is in the billboards weekly charts.
Current Popularity - The popularity index obtained from Spotify API.

Implementation:

Approach 1:

For the first baseline, we implemented a model to predict the current popularity index of the song using area under the curve, the number of years before which the song was released, and the relative play count of the songs. Linear Regression gave a mean absolute error of 12 on the test set and 11 on train set and Random Forest Regressor on this model gave a mean absolute error of 11 on the test set and 5 on the train set. Mean absolute error gives us an idea of how far the predicted popularity is from the actual popularity

After this, we implemented another model using the area under the curve, the number of years before the song was released, relative play count of the songs, and *c* as the features. c value for each song was calculated using the decay model $M_1 = M_2 * c^y$ where M_1 denotes the current popularity, M_2 denotes the relevant corresponding metric for the song and *'y'* denotes the number of years from the year of release of the song. Linear Regression gave a mean absolute error of 11 on the test set and 10 on train set and Random Forest Regressor on this model gave a mean absolute error of 9 on the test set and 3 on the train set.



This graph shows the plot of mean absolute error for five years as is obtained by our baseline model. We see here that the mean absolute error decreases for the songs that are released in recent years as compared to the songs that were released earlier.

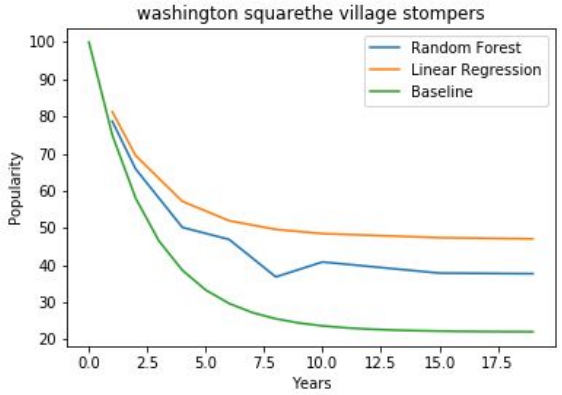
Approach 2:

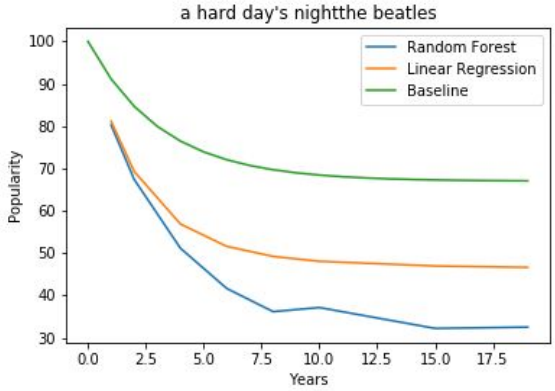
We implemented the decay law for each song to create an innovative model to predict songs that over-perform and underperform. The rationale behind implementing the decay law for each song was to derive a dataset which gave us popularity as a target variable over the years. Then, we fitted our model on this derived dataset and predicted values for intermediate years as target variables. We further plotted these predicted values with the decay curve for each song to make a comparison between our predicted values and the original derived values. This gives us a parameter to measure and compare the performance of a song over the years and analyze whether a song is underperforming or over performing with respect to our model.

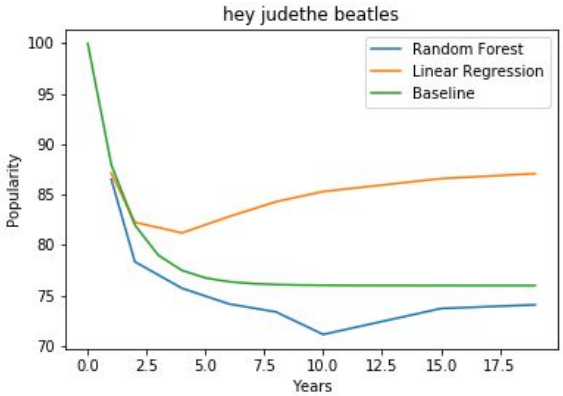
Step by Step

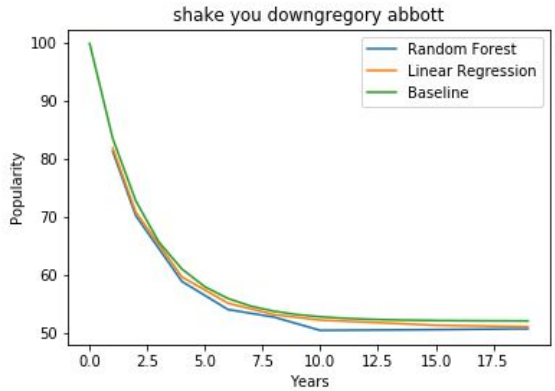
1. Calculate c value for each song for the decay model $M_1 = M_2 * c^y$ where M_1 denotes the current popularity, M_2 denotes the relevant corresponding metric for the song's release year and *'y'* denotes the number of years before which the song was released.
2. Next, we derive a vector for M_2 using this *c* value over the range of years, which results in a decay graph that we normalized.
   Assumptions
   a. The popularity of a song was at its peak in the year it was released, and we have considered 100 to be that peak value for every song irrespective to each other.
   b. This derived dataset depicts the true distribution of popularity for that song over the years.
3. Next, we used different models to predict popularity for various years for some predetermined intervals, which in turn again gives us a vector of predicted popularity over the years.
4. We plot these 2 vectors together on a graph for the test songs and make a comparison to determine if a particular song is over-performing or underperforming relative to our predictions.
5. So, if a song's derived popularity curve is above the predicted popularity curve, we say that the song is over performing and we say it is underperforming for the alternate scenario according to our predictions.

We implemented this model on a small subset of 1500 songs between 1960-1998. And below are some graphs depicting the comparison between our derived predictions (from the decay law, which we assume to be the true distribution of popularity over the years) and our prediction model.



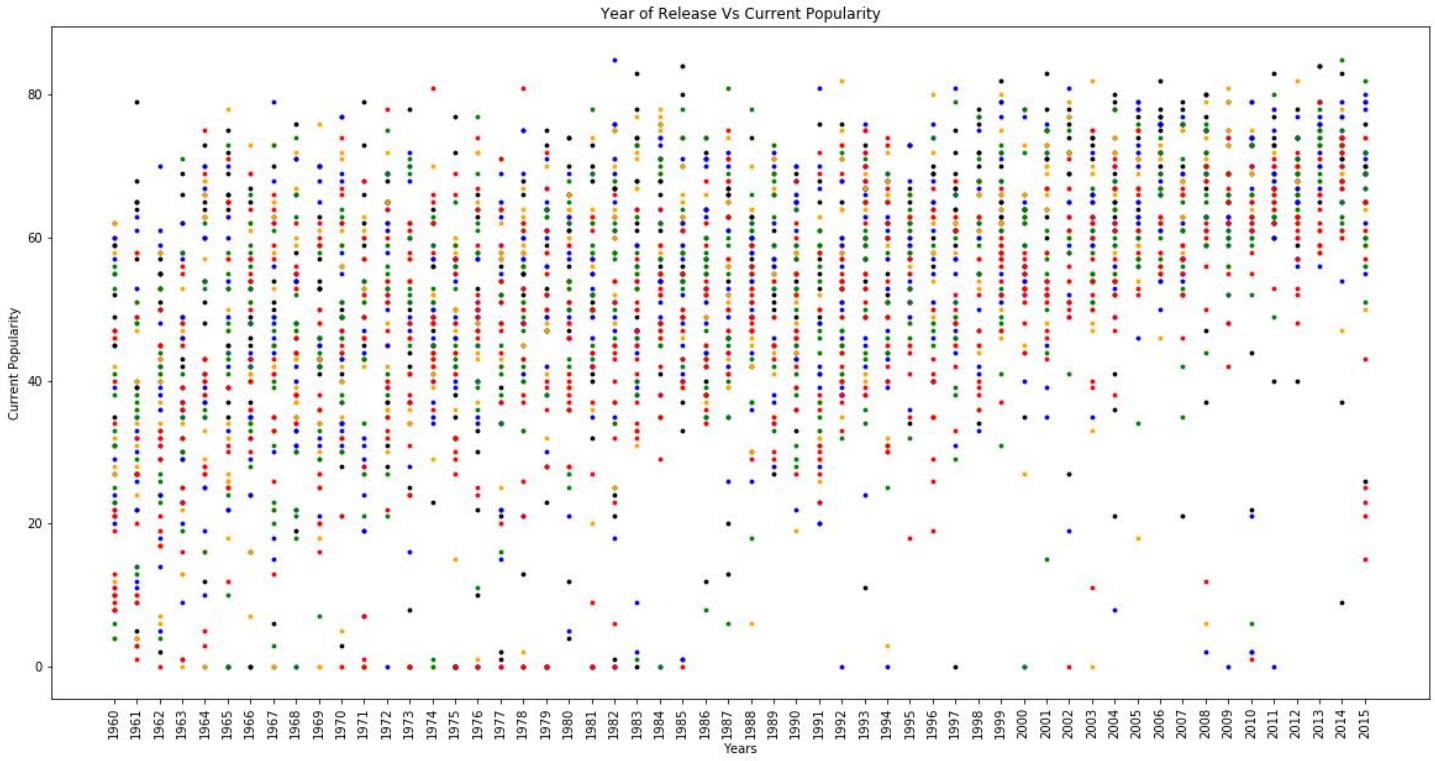






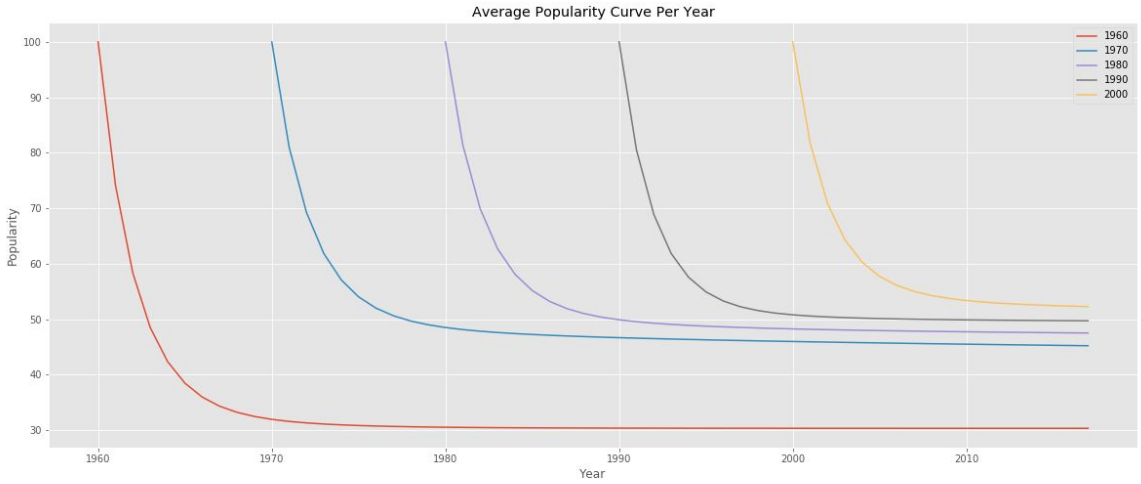
Above are some plotted graphs which depict some interesting observations.
Y-Axis - Popularity Index.
X-Axis - Years from the initial release year.
Washington Square is a billboard rank 9 song from 1963 which has a good AUC value, and thus, in our sniff test we predicted that this song should have a good current popularity but our decay law didn't think so. But, our prediction models perfectly predict that the song is under performing which is what we thought initially. Similarly, the famous song from Beatles should definitely have a high current popularity but for a song from 1964 having a popularity index of 80+ does seem odd, and our prediction models do predict that it is slightly over performing. Further 2 plots go on to depict some other interesting observations that 2 different prediction models can predict two completely different popularity distributions and the same models can predict exactly the same distribution of popularity. Also, as hypothesized, it is not necessary for all songs to strictly follow the decay law which we can clearly see in the 3rd graph.

Analysis:

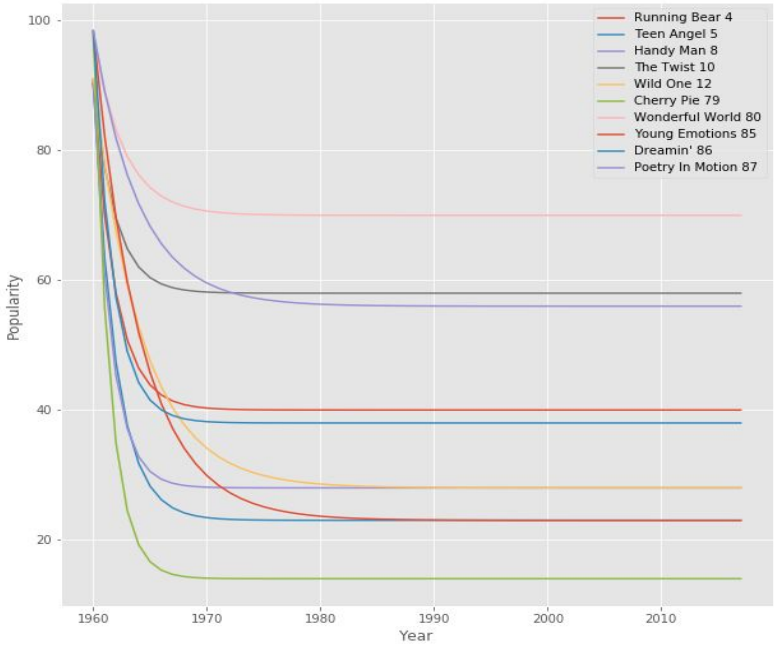


The graph above shows the current popularity of the song vs the year of release of the song. The different colours used represents the rank of the song in the year of release. The color code used - black(rank 0 - 15), orange(rank 15-30), blue(rank 30-45), green (rank 45-60) and red(rank 60-75). We observe that the songs of recent years have an upper trend as compared to the songs from the 60s and 70s. The popularity for recent songs is more clustered in the higher range of popularity while the 60s and 70s are scattered throughout the popularity range. There are some outliers in this as well. Few songs released even in recent years do not have a high popularity index while few songs from 60s and 70s have a comparatively higher popularity than other songs from their years. There are few old songs that have endured even today, and few that have lost

popularity over the years. From this plot we also analyze that if the black or orange colored points are in the lower spectrum of the plot then they are underperforming compared to their ranks and if green and red are in the higher spectrum then they are over performing.



The above graph is the average popularity curve of the songs in that year. This plot also supports the hypothesis that the current songs are more popular compared to the old songs. This in addition to the plot above it implies that only a few songs from the older era like 1960s and 1970s endure.



In the above plot, we have plotted the popularity curve of a few of the songs from the year 1960. We observe that only the initial rank of the song doesn't correlate much with the popularity curve. Th initial rank in that year's billboard chart doesn't really imply whether the song will endure. For eg: Wonderful World has rank 80 but has the slowest decay in the curve whereas Wild One with rank 12 has a steeper decay.

Next Steps :

- In the next part of our project, we want to focus on developing a method to define some robust features which helps us define the initial popularity index and then test our best model to predict its popularity index over the years and analyse if a popular song really does endure over the years or not.
- To do so, we plan to incorporate some more features and implement some weighted importance according to some model fitting to get an idea about how each feature affects our model.
    - Genre is a feature we want to incorporate as it defines a shift in the music taste of general public over the years, which should be a good metric for the popularity index for a song from a particular era.
    - Spotify provides some features for songs like danceability, energy, loudness that we would like to explore and check if they help in calculating this popularity metric.
    - It will be interesting to see if the performance of a song at prestigious award ceremony like the Grammys would affect the notion of a song in the eye of the public i.e. would it really lead to the song being more popular and ultimately will it really endure.
- We will also try to correlate the artist popularity to the recording popularity and check whether the endurance of artist affects the endurance of songs.