UNIVERSITY OF ILLINOIS CHICAGO

PROJECT REPORT

SPOTIFY SONGS & ARTISTS – ANALYSIS, PREDICTON AND RECOMMENDERS

IDS 575 – MACHINE LEARNING & STATISTICS

GROUP 6 – ANIRUDHA BALKRISHNA, PRERNA PRASAD, SHUBHAM KHODE & SHUBHANGI GOYAL

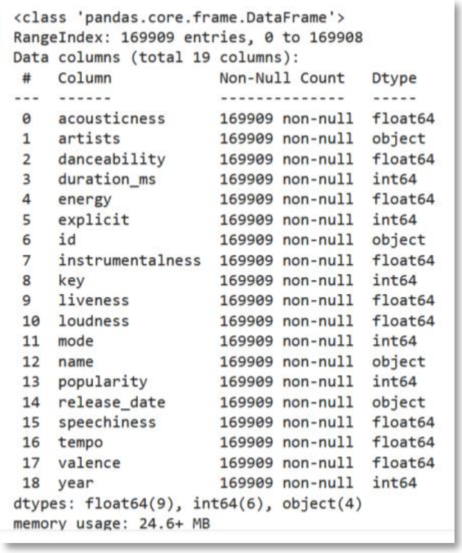
Introduction

Spotify is cloud-based music platform that provides cross-device access to over 50 million songs, and a rapidly rising number of podcasts and videos. Spotify is the biggest player (with 365 million users and 165 million subscribers) in the music streaming market but must maintain its position between Tech giants like Apple (Apple Music), Amazon (Amazon Music), and Google (YouTube Music). To do so, two peer groups must be in the focus of Spotify: artists and users. To deliver the best service to them, there is one thing at the heart of Spotify: Algorithms & Machine Learning. Data and algorithms are Spotify's opportunity to not be crunched between Apple, Amazon, and Google. Identification of popular music records through analysis of features and user preferences helps streaming platforms to create an efficient feedback loop.

Dataset

As mentioned in proposal we shortlisted 3 datasets but choose Spotify 160k dataset as it has a good spread of data, almost the same number of features as other datasets and most importantly it does not strain the computing resources. The data has more than 160,000 rows worth of data with 18 columns. These columns include sound related features like loudness, tempo, acoustics, valence etc. It also includes details on the artists, release dates etc. The dataset has songs spread over multiple years i.e., songs released as early as 1921 & newly released songs as recent as 2021.

The image below shows us details about the dataset -



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 169909 entries, 0 to 169908
Data columns (total 19 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   acousticness      169909 non-null  float64
 1   artists           169909 non-null  object
 2   danceability      169909 non-null  float64
 3   duration_ms       169909 non-null  int64
 4   energy            169909 non-null  float64
 5   explicit          169909 non-null  int64
 6   id                169909 non-null  object
 7   instrumentalness  169909 non-null  float64
 8   key               169909 non-null  int64
 9   liveness          169909 non-null  float64
 10  loudness          169909 non-null  float64
 11  mode              169909 non-null  int64
 12  name              169909 non-null  object
 13  popularity        169909 non-null  int64
 14  release_date      169909 non-null  object
 15  speechiness       169909 non-null  float64
 16  tempo             169909 non-null  float64
 17  valence           169909 non-null  float64
 18  year              169909 non-null  int64
dtypes: float64(9), int64(6), object(4)
memory usage: 24.6+ MB
```

As evident from the findings, the data does not have any null values. This limits the effort involved in cleaning (and imputing) the dataset.

Exploratory Data Analysis

To understand how the top songs and artists are spread across various geographical regions through the years, we first need to get the required data. For this purpose, we used Spotify Charts (https://spotifycharts.com/regional/) which provides the Top 200 songs on a daily or weekly basis for

72 different countries. To make the analysis manageable and understandable, our strategy is to focus only on the top (rank 1) song from each week for all the regions across the last 6 years (01/2017 - 03/2022).

Data Preparation

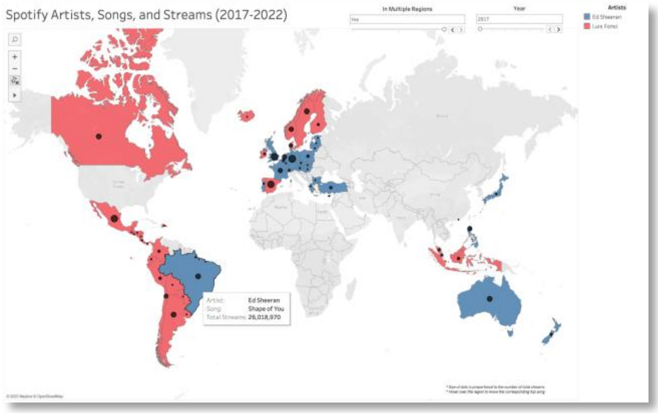
Spotify Charts recently added ‘Cloudflare’ protection to their website, hence, to bypass it for scrapping data, we use an external package called ‘*cloudscrapper*’.

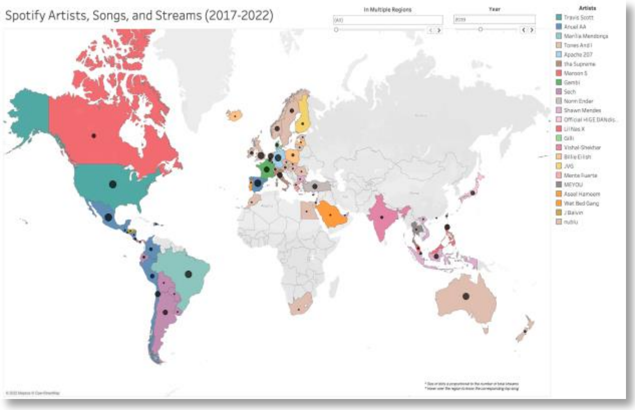
After fetching all the necessary data, we perform pre-processing steps to clean and prepare data for analysis. This involves calculating the total number of streams for each song and then grouping them to find the top song for each year and region based on the maximum number of streams. This data is then saved as a ‘.csv’ file and imported into Tableau for further analysis.

Visualizations & Insights

Spotify Artists, Songs, and Streams visualization:

Our first interactive visualization gives a good representation of the top artists (along with their songs and number of streams) in a particular year across various regions. To answer the question *“Are people listening to the very same top-ranking songs in countries far away from each other?”,* we create an additional filter (called ‘In Multiple Regions’) that marks the songs that have topped the weekly chart in more than one country.

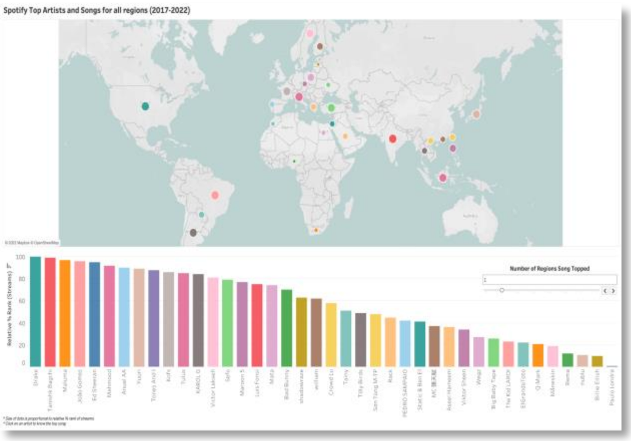


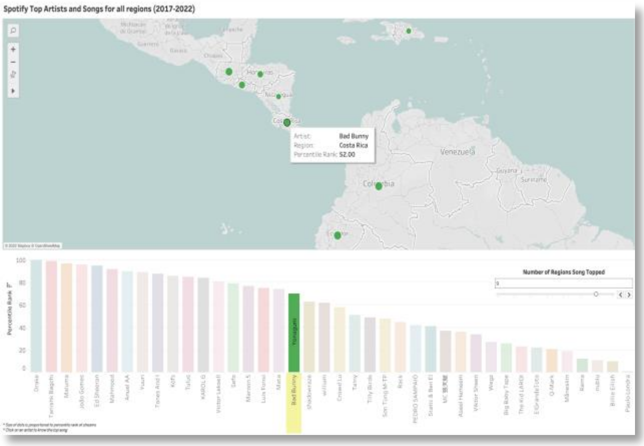


In general, countries that are far away from each other listen to different top-ranking songs. On the other side, we find groups or clusters of neighboring regions (e.g., South American countries, Central European countries, UK & Ireland, and Ukraine & Russia) that listen to same top-ranking artists. One possible explanation is similarity of language among these clusters. The exception to this assumption is when a song is released by a globally popular artist (like Ed Sheeran).

Spotify Top Artists and Songs for all regions (2017-2022) visualization:

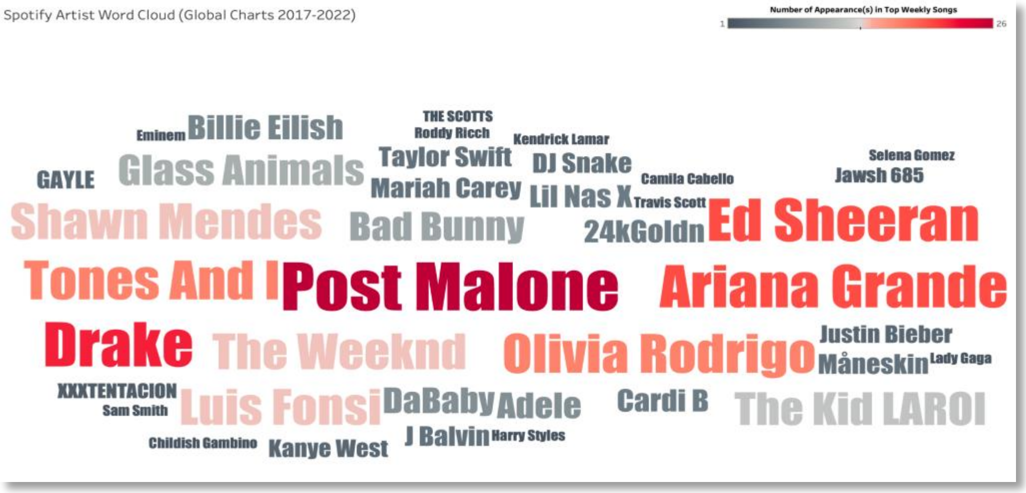
Our second visualization is geared toward answering another question *“Do continents share the same top-ranking artists or songs?”.* For this, we drop the year, and then based on the criteria of the highest number of streams, we find the top-ranking artists and their songs for each region. Due to huge difference in the number of streams for different artists across the globe, we perform feature scaling to get their relative rankings in percentile form. We also group the artists by the number of regions their songs topped to get a better picture of how artists are performing across regions. Between continents, we do observe the same top-ranking artists. However, within a continent or in neighbouring continents, we can see few artists that are clustered together.


Spotify Top Artists and Songs for all regions (2017-2022)
Number of Regions Song Topped
Relative % Rank (Streams)



Spotify Artist Word Cloud (Global Chart) visualization:

To find out the most popular artists across the world, we took a different approach. Here we scrapped the data for the top (rank 1) artists and their songs from Spotify's Global Weekly Charts (https://spotifycharts.com/regional/global/weekly/) for years 2017 to 2022. The artists were ranked by the number the times they topped the global weekly chart. A word cloud of artists based on their ranks gives a quick visual representation of their popularity and consistency through the years.



An interesting observation is that artists with collaborations in songs are more popular. Featuring artists make a song powerful, richer, and get more attention from different audiences.

Understanding Features

In our dataset, we need to understand the features for exploration of pattern analysis for prediction and recommendation system.

Target Variable Analysis:

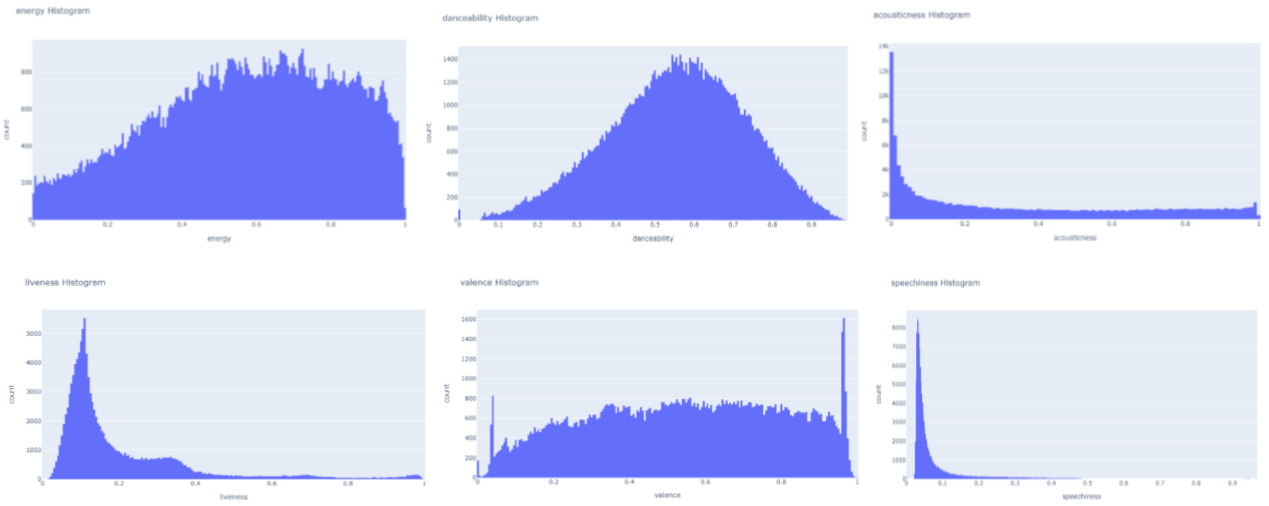
Our target variable is 'Popularity' where it is the popularity of the song. The value will be between 0 and 100, with 100 being the most popular.

Feature Engineering:

- Added one column as a part of data cleaning 'Song_decade'.
- Year - Dropped 'release date' and kept only year column while filtering 1960 to 2020 data.
- Duration_minutes – Converted duration millisecond to minutes for data cleaning purpose.
- Artists – our original dataset had artists column as a string with all artists listed, that was cleaned to have only the main artist's name in str format.
- Song and Artist Popularity scaled – Here song popularity ranges from 0 to 100. The rank of an artist is scaled so the values represent the rank in percentile of the ranking value. This was then balanced using MinMaxScaler on range from 0 to 1.
- To add this custom feature (Artist Popularity), we harnessed artists worldwide ranking from third party site and after mapping them with our unique set of artists we got top 200 artists across 13k rows of song data.

Visualizations to understand features:

Grouped Years and then plotting features (danceability; Speechiness and Song_popularity) with respect to years distribution and size is based on song_popularity variation –



Insights:

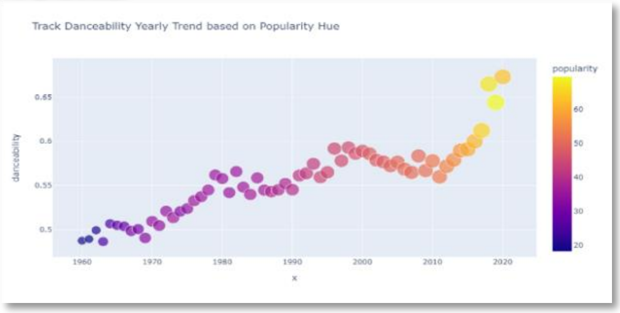
- Loudness and energy are highly correlated. This makes some sense as energy is influenced by the volume the music is being played at.
- Acousticness is negatively correlated with energy, loudness, and year. This is thought provoking as it seems that acoustic songs have reduced over the years per trends identified on this dataset.
- Valence & danceability are highly corelated. Dance songs are usually happier and in a major key.

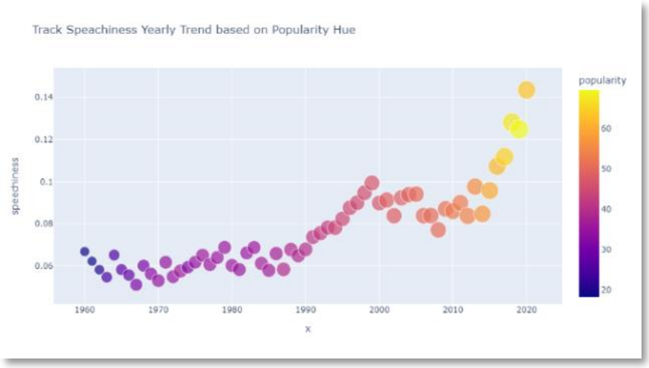
Thus, from this data, it would be better for an artist to create a high energy song with either electric instruments or electronic songs to have the best chance at generating the most popularity.

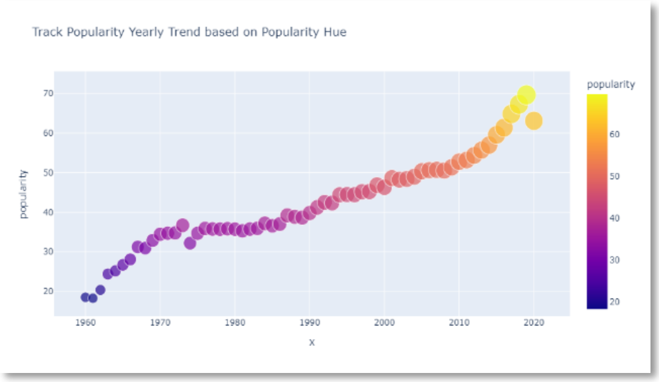
Yearly feature trend analysis

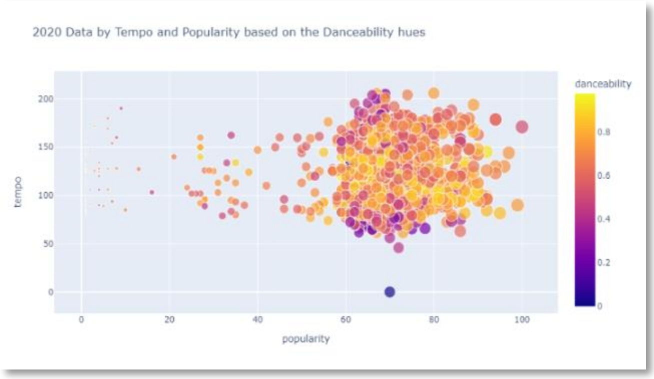
Popularity by Tempo for the year 2020, varying by danceability.

The higher the Tempo, the higher is the Danceability displayed by the yellow hues towards the right side of the plot


Track Danceability Yearly Trend based on Popularity Hue

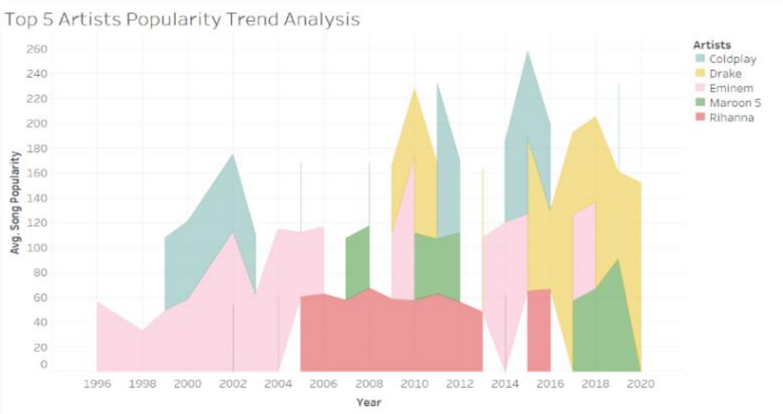

Track Speachiness Yearly Trend based on Popularity Hue


Track Popularity Yearly Trend based on Popularity Hue

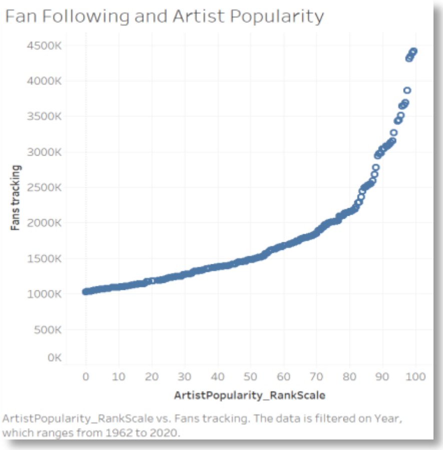

2020 Data by Tempo and Popularity based on the Danceability hues

Understanding Popularity:

We try to understand the influence of an Artist's popularity on their Song's Popularity and Fans Tracking that is directly related to an Artist's Rank from below visualizations:


Top 5 Artists Popularity Trend Analysis

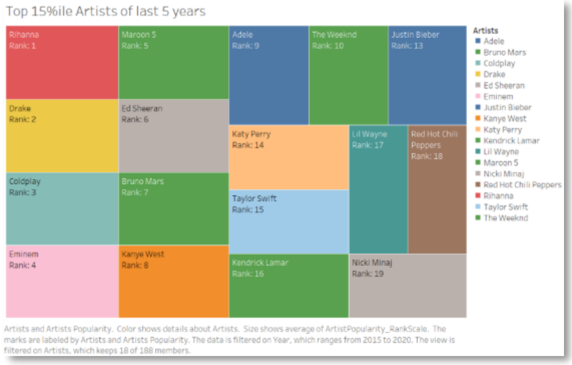
The plot of average of Song Popularity for Year. Color shows details about Artists. The view is filtered on Artists, which keeps Coldplay, Drake, Eminem, Maroon 5 and Rihanna.


Fan Following and Artist Popularity

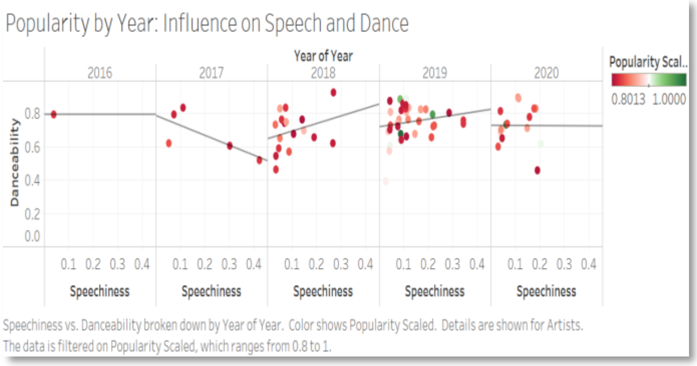
ArtistPopularity_RankScale vs. Fans tracking. The data is filtered on Year, which ranges from 1962 to 2020.

Insights:

Spotify's Algorithm to rank a Song's popularity considers the number of times it is streamed in 'recent' times. The recent factor might negatively influence songs that are over many decades ago no matter how well the song fared during its release time. The same applies to the artist's popularity as well, time sensitivity and an upward trend as seen in the graph above.

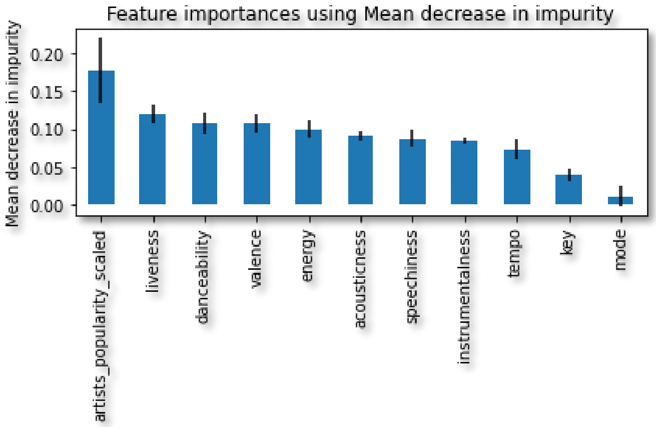

Top 15%ile Artists of last 5 years

Artists and Artists Popularity. Color shows details about Artists. Size shows average of ArtistPopularity_RankScale. The marks are labeled by Artists and Artists Popularity. The data is filtered on Year, which ranges from 2015 to 2020. The view is filtered on Artists, which keeps 18 of 188 members.


Popularity by Year: Influence on Speech and Dance

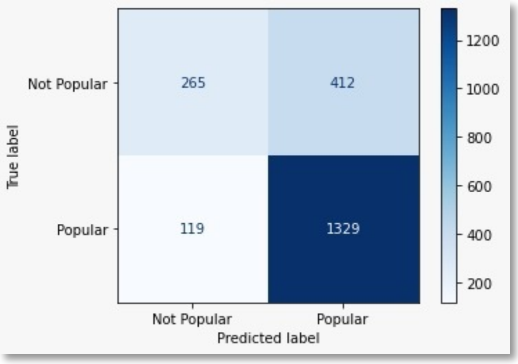
Speechiness vs. Danceability broken down by Year of Year. Color shows Popularity Scaled. Details are shown for Artists. The data is filtered on Popularity Scaled, which ranges from 0.8 to 1.

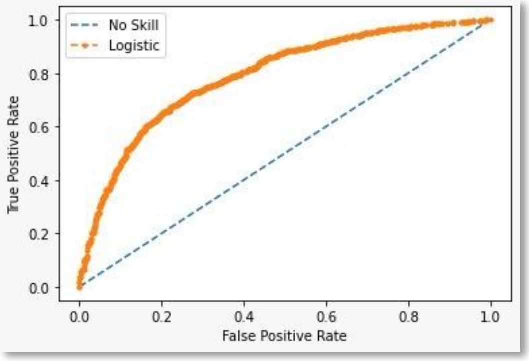
Feature Importance:


Feature importances using Mean decrease in impurity

Predictive Model Building

In the modelling process as already mentioned the target variable is 'song_popularity' which should classify if the song being popular or not. Considering added feature of scaled popularity which varies from 0.0 to 1.0 range has been optimized based on thresholds which is at 42. Once threshold is selected, made original target variable binary that is 0 and 1 for classification modeling. Selected all features required and then split for train and test. Later ran two important algorithms – Logistic Regression and Random Forest for modelling and built confusion matrix along with AUC for evaluation. For our first modelling approach the above-mentioned process was followed where all the features were fed in the model and accuracy of 65% was achieved on Logistic model and 68% for random forest. For our second modeling approach we considered our custom artist popularity scaled feature in addition to our model 1. The model 2 accuracy with logistic regression went to 75% whereas our random forest model gives a promising result of 80%. Here, ~20% feature importance of scaled artists feature holds with next highest being liveliness at 10% and so on.





Artist ranking data over the years is not widely available and thus we went with assumption that the third-party data we selected is worldwide true ranking. If the data was available, we could have increased the scope of the dataset by average scoring or some other metrics.

Recommending Music

For our project, we have drawn inspiration from Spotify’s existing system and have created Content based recommender systems and an approach to evaluate them.

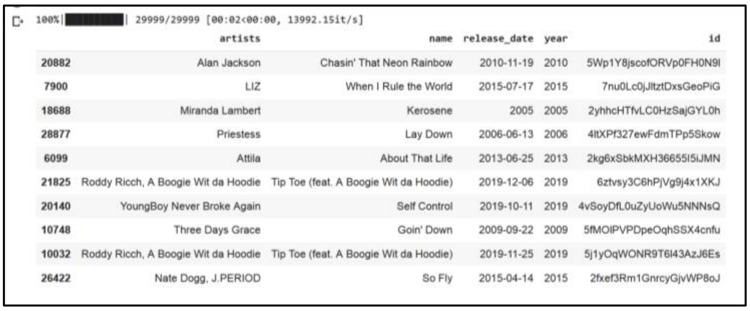
Recommenders using different similarity measures

We have made two recommenders. They are as follows –

1. Recommender based on Manhattan Distance

In the recommender, we use the numerical features of a given song to find similar songs using Manhattan distance. Quoting from the paper, “On the Surprising Behavior of Distance Metrics in High Dimensional Space” (by Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Kiem), ”For a given problem with a fixed (high) value of the dimensionality d, it may be preferable to use lower values of p. This means that the L1 distance metric (Manhattan Distance metric) is the most preferable for high dimensional applications”. Thus, Manhattan Distance is preferred over the Euclidean distance metric as the data has multiple features.

Below are the songs recommended by plugging – “Harleys in Hawaii” by Katy Perry as input.



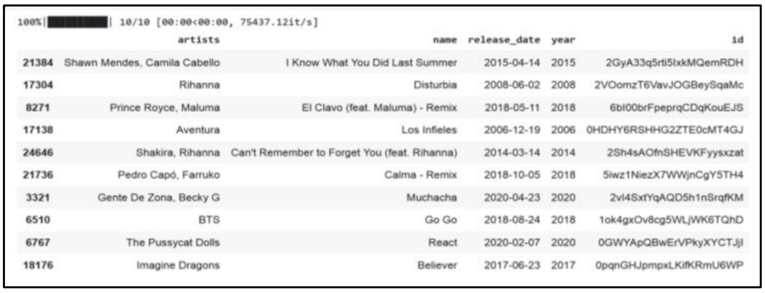
```
100%|██████████| 29999/29999 [00:02<00:00, 13992.15it/s]
```

	artists	name	release_date	year	id
20882	Alan Jackson	Chasin' That Neon Rainbow	2010-11-19	2010	5Wp1Y8jscofORVp0FH0N9I
7900	LIZ	When I Rule the World	2015-07-17	2015	7nu0Lc0jJltztDxsGeoPiG
18688	Miranda Lambert	Kerosene	2005	2005	2yhhcHTfvLC0HzSajGYL0h
28877	Priestess	Lay Down	2006-06-13	2006	4ltXPf327ewFdmTPp5Skow
6099	Attila	About That Life	2013-06-25	2013	2kg6xSbkMXH36655I5iJMN
21825	Roddy Ricch, A Boogie Wit da Hoodie	Tip Toe (feat. A Boogie Wit da Hoodie)	2019-12-06	2019	6ztvsy3C6hPjVg9j4x1XKJ
20140	YoungBoy Never Broke Again	Self Control	2019-10-11	2019	4vSoyDfL0uZyUoWu5NNNsQ
10748	Three Days Grace	Goin' Down	2009-09-22	2009	5fMOIPVPDpeOqhSSX4cnfu
10032	Roddy Ricch, A Boogie Wit da Hoodie	Tip Toe (feat. A Boogie Wit da Hoodie)	2019-11-25	2019	5j1yOqWONR9T6l43AzJ6Es
26422	Nate Dogg, J.PERIOD	So Fly	2015-04-14	2015	2fxef3Rm1GnrcyGjvWP8oJ

2. Recommender based on Cosine Similarity

For this recommender, we import the cosine similarity functionality from the ‘sklearn.metrics’ library. We use this function to create a matrix of cosine similarity measures between all songs in the database. Next, we sort this matrix based on these values to get the top recommendations for a selected song.

The songs recommended for the same input are as follows –



```
100%|██████████| 10/10 [00:00<00:00, 75437.12it/s]
```

	artists	name	release_date	year	id
21384	Shawn Mendes, Camila Cabello	I Know What You Did Last Summer	2015-04-14	2015	2GyA33q5rti5IxkMQemRDH
17304	Rihanna	Disturbia	2008-06-02	2008	2VOomzT6VavJOGBeySqaMc
8271	Prince Royce, Maluma	El Clavo (feat. Maluma) - Remix	2018-05-11	2018	6bI00brFpeprqCDqKouEJS
17138	Aventura	Los Infieles	2006-12-19	2006	0HDHY6RSHHG2ZTE0cMT4GJ
24646	Shakira, Rihanna	Can't Remember to Forget You (feat. Rihanna)	2014-03-14	2014	2Sh4sAOfnSHEVKFyysxzat
21736	Pedro Capó, Farruko	Calma - Remix	2018-10-05	2018	5iwz1NiezX7WWjnCgY5TH4
3321	Gente De Zona, Becky G	Muchacha	2020-04-23	2020	2vl4SxtYqAQD5h1nSrqfKM
6510	BTS	Go Go	2018-08-24	2018	1ok4gxOv8cg5WLjWK6TQhD
6767	The Pussycat Dolls	React	2020-02-07	2020	0GWYApQBwErVPkyXYCTJjI
18176	Imagine Dragons	Believer	2017-06-23	2017	0pqnGHJpmpxLKifKRmU6WP

Evaluating Recommenders

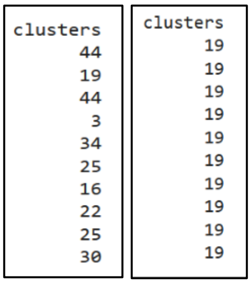
In order to evaluate these recommenders and determine if one is better than the other, we decided to test the following methods –

Convert the unsupervised problem to a supervised problem - Get the recommended songs from both the recommenders and compare them with what the user has chosen. The recommender with the most gaps compared to the User's choice will be considered inferior. But the problem is that we do not have any supervisory signal (i.e., we do not know what the user would have chosen after a particular song). Therefore, we cannot use this approach which is widely accepted. Also, if we review the recommendations, there are no similar songs in any of the top 10 songs recommended.

Use another Heuristic measure - After we make the recommendations using the two recommenders, we save the recommendations. Next, we use K-means clustering (i.e., Euclidean distance) to divide the data into 'k' clusters. Let's think of these clusters as Genres. Then, for the recommendations saved, we check which cluster are the songs classified into and if the recommendations are largely in the same cluster for a recommender. If we observe that most songs are classified into the same cluster, we may be able to infer that the set of recommendations is similar as per two similarity measures and probably the better recommender. (Note - It may not actually be the better recommender, it is just similar by 2 measures)

When we use the second (heuristic) approach to our recommenders, we notice that all songs recommended by the second recommender (Cosine similarity) are in the same cluster whereas, the recommendations made by the first recommender (Manhattan Distance) are spread into multiple clusters. Thus, we can assume that recommendations made by Recommender – 2 are similar as per two measures and may be the better recommender.

The figure below shows the top 10 songs from each of the recommender and their respective clusters.



```
clusters
      44
      19
      44
       3
      34
      25
      16
      22
      25
      30
```

```
clusters
      19
      19
      19
      19
      19
      19
      19
      19
      19
      19
```

Conclusion

We have taken inspiration from Spotify's current system, which is a combination of exploratory analysis, popularity prediction and recommendation. While Spotify uses a variety of tools and metrics to create their business moat (Unique business offering) through data and algorithms, we have used Tableau for our exploratory data analysis and python to create the predictors & recommenders. Throughout our project, we aim to replicate Spotify's approach at a smaller scale. Our objective was to understand the complexities of analyzing such data and understand the non-trivial challenges that surround this approach of using data and machine learning.

References

[1] https://gist.github.com/hktosun/d4f98488cb8f005214acd12296506f48

[2] https://medium.com/analytics-vidhya/analyzing-spotify-dataset-with-python-6ba9cb82a486

[3] https://www.kaggle.com/vatsalmavani/music-recommendation-system-using-spotify-dataset

[4] https://towardsdatascience.com/song-popularity-predictor-1ef69735e380

[5] Uncovering How the Spotify Algorithm Works | by Hucker Marius | Towards Data Science

[6] https://www.popsci.com/technology/spotify-audio-recommendation-research/

[7] https://towardsdatascience.com/part-iii-building-a-song-recommendation-system-with-spotify-cf76b52705e7

[8] https://medium.com/@kunal_gohrani/different-types-of-distance-metrics-used-in-machine-learning-e9928c5e26c7

Task Ownership

Tasks	Owner
Literature Review	All
EDA	Shubham, Prerna
Visualization	Shubham, Prerna
Data Cleaning, Pre-processing, and Feature Engineering	All
Modelling	Shubhangi
Recommendation	Anirudha

Appendix

Spotify's Systems

In 2008 Spotify started changing the world around music by introducing music streaming. Since then, music on CDs and DVDs has left all our lives and the music industry changed.

Now, Spotify is the biggest player (with 365 million users and 165 million subscribers) in the music streaming market but must maintain its position between Tech giants like Apple (Apple Music), Amazon (Amazon Music), and Google (YouTube Music). To do so, two peer groups must be in the focus of Spotify: artists and users. To deliver the best service to them, there is one thing at the heart of Spotify: Algorithms & Machine Learning. The better Spotify understands the users and the greater the customer experience is, the more users can be convinced, converted to paying customers, and held as customers. In other words, data and algorithms are Spotify's opportunity to not be crunched between Apple, Amazon, and Google. The data gathered and used by Spotify is quite extensive. At Spotify, almost everything is tracked.

Essentially, the music industry today has become a competitive but an uncertain industry. Millions of records are downloaded every day and thousands of records are created daily but only a select few are considered popular. Identification of popular music records through analysis of features and user preferences helps streaming platforms to create an efficient feedback loop.

About the Datasets

After searching for publicly available datasets, we shortlisted 3 datasets. They are

1. Spotify 1.2M - https://www.kaggle.com/rodolfofigueroa/spotify-12m-songs
2. Spotify 160k - https://www.kaggle.com/ektanegi/spotifydata-19212020
3. Spotify API: Spotipy - https://spotipy.readthedocs.io/en/2.19.0/

We explored the use of these different datasets and concluded that using the second dataset is most beneficial as it has a good spread of data, almost the same number of features as other datasets and most importantly it does not strain the computing resources.

Exploratory Data Analysis & Feature Understanding

[1] Tableau Link for Spotify Artists, Songs, and Streams visualization: https://public.tableau.com/app/profile/shubham.khode/viz/SpotifyArtistsSongsandStreams/ArtistSongbyRegionandYear

[2] Tableau Link for Spotify Top Artists and Songs for all regions (2017-2022) visualization:

https://public.tableau.com/app/profile/shubham.khode/viz/SpotifyTopArtistsandSongsforallregions2017-2022/SpotifyTopArtistsandSongsforallregions2017-2022

[3] Tableau Link for Spotify Artist Word Cloud (Global Chart) visualization:

https://public.tableau.com/app/profile/shubham.khode/viz/SpotifyArtistWordCloudGlobalChart/SpotifyArtistWordCloudGlobalChart

[4] Tableau Link for Visualization to understand Artist Popularity:

https://public.tableau.com/app/profile/prerna.prasad8880/viz/SpotifyEDATableau/SongsTrendsbyArtists

Recommenders

Recommender Systems are nothing new at all and we encounter them on every corner of the Internet: Netflix, Google, Amazon, E-Commerce Shops, and so on. Whenever we get a recommendation on possibly fitting products, services, or persons we talk about recommendation systems. In the case of LinkedIn, the recommender system suggests persons based on our network, working history, and interests, the Netflix algorithm recommends movies and series that fit best to our film flavor and Amazon provides us with similar products or complementary products that other customers bought together with the product at hand.

So, what a recommender system simply does is deliver suggestions based on behavior or characteristics that have been tracked by the system. As the Spotify Research Team states, “Users are overwhelmed by the choice of what to watch, buy, read, and listen to online” and hence recommender systems are necessary to help navigate and facilitate the decision process.

Recommender systems can basically be divided into 3 types:

- **Collaborative-filtering:** Collaborative filtering assumes that people who agreed in the past will agree in the future and that they will like similar kinds of objects as they liked in the past.

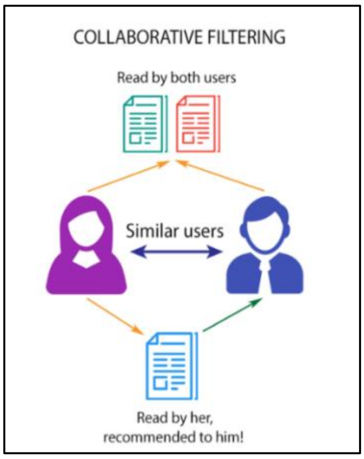


Figure # – Collaborative Filtering

- **Content-based:** Content-based recommender systems rely on the available “features” of the user-item relationship to build a model.

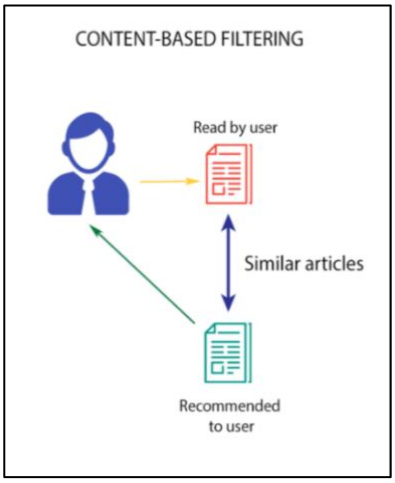


Figure # – Content Filtering

- **Hybrid:** This type obviously combines both, content-based and collaborative filtering.

Spotify's Recommendation system

Spotify uses a Hybrid filtering with an NLP system. The NLP system helps to understand recent trends by crawling web pages. Additionally, Spotify uses a reinforcement learning technique called "BaRT" (Bandits for Recommendations as Treatments) to include a reward – action function. If a song is heard for more than 30 seconds, it is registered as reward and the action is to provide similar songs as recommendation.

Similarity Metrics

- Manhattan Distance –

The Distance formula for Manhattan distance is given as follows –

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

- Cosine Similarity –

The formula for cosine similarity is given below –

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$