Spotify Music Recommendation System

Giovanni Visco and Ege Telatar

Giovanni Visco
Engineering
University of Colorado-Boulder
Boulder CO United States
giovanni.visco@colorado.edu

Ege Telatar
Engineering
University of Colorado-Boulder
Boulder CO United States
ege.telatar@colorado.edu

Introduction

All variations of recommendation systems are present in the real world. The success of a number of tech titans can be attributed to their recommendation systems. Consider Netflix, Google, Spotify, these companies all excel at providing each unique user with recommendations. Both Ege Telatar and Giovanni Visco have a passion rooted in music, and more importantly a desire to understand how and what exactly makes a great recommendation system. This is the driving force behind the idea of the project. Given the new methods learned in CSCI 4022, the team wanted to explore them and their differences. What methods create the "best" recommendations? Of course, it is difficult to quantify a quality recommendation, so one has to logically evaluate it to ensure it makes sense, more on that later.

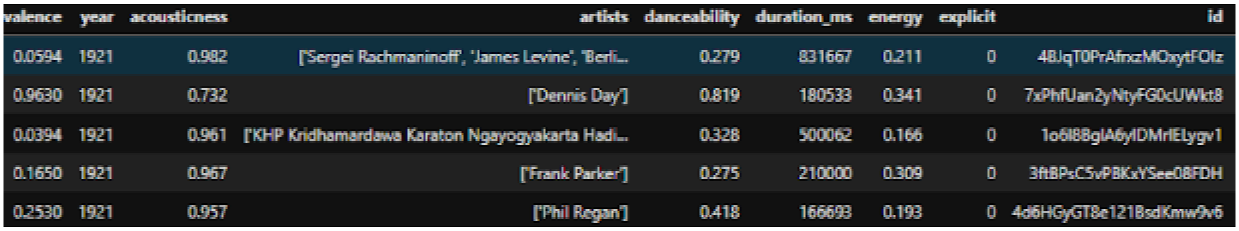
The goal of this project is to successfully create a recommendation system that provides a user with a list of Spotify song recommendations given with a single song or list of songs. This will be explored across two methods: K-Nearest-Neighbors and Collaborative Filtering. Then the differences between the performance of these two algorithms will be analyzed to understand how they effectively create recommendations and which may perform better.

Data

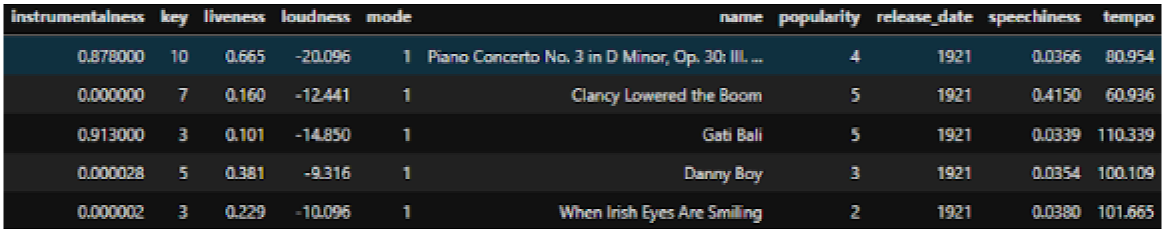
Since this project involves the exploration of two separate methods two separate datasets are needed. Both datasets have been sourced from Kaggle and will be provided as hyperlinks below.

The first dataset used for this project is a Spotify song dataset. This dataset focuses on the measurable features of each song, all of which are provided by the Spotify API. There are 170,653 unique entries each of which has 19 attributes. In this case, each entry is a unique song that contains the information on the song's valence, year, acousticness, artists, danceability, duration, energy, explicit, id, instrumentalness, key, liveness, loudness, mode, name, popularity, release date, speechiness, and tempo. This dataset breaks down the individual songs into their respective features. This will be hugely valuable for the KNN algorithm to create content-based recommendations. If a user likes a certain song, the KNN model will provide a recommendation based on the most similar songs based on the attributes of the song the user likes. The link and first five rows of this dataset are below.

Dataset for K Nearest Neighbors



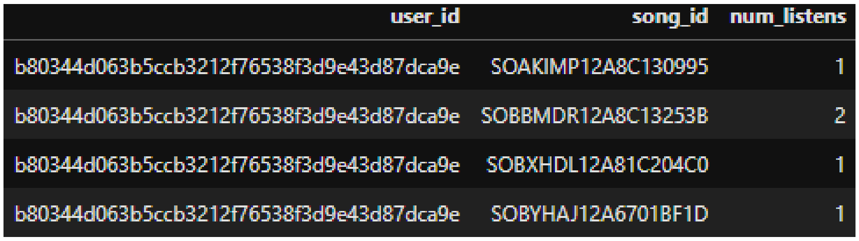
valence	year	acousticness	artists	danceability	duration_ms	energy	explicit	id
0.0594	1921	0.982	['Sergei Rachmaninoff', 'James Levine', 'Berli...	0.279	831667	0.211	0	4BJqT0PrAfrxzMOxytFOIz
0.9630	1921	0.732	['Dennis Day']	0.819	180533	0.341	0	7xPhfUan2yNtyFG0cUWkt8
0.0394	1921	0.961	['KHP Kridhamardawa Karaton Ngayogyakarta Hadi...	0.328	500062	0.166	0	1o6I8BglA6ylDMrIELygv1
0.1650	1921	0.967	['Frank Parker']	0.275	210000	0.309	0	3ftBPsC5vPBKxYSee08FDH
0.2530	1921	0.957	['Phil Regan']	0.418	166693	0.193	0	4d6HGyGT8e121BsdKmw9v6



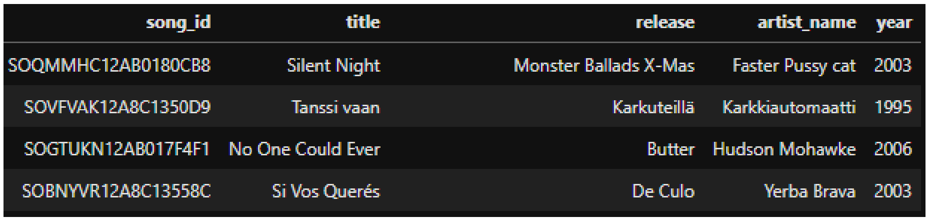
instrumentalness	key	liveness	loudness	mode	name	popularity	release_date	speechiness	tempo
0.878000	10	0.665	-20.096	1	Piano Concerto No. 3 in D Minor, Op. 30: III. ...	4	1921	0.0366	80.954
0.000000	7	0.160	-12.441	1	Clancy Lowered the Boom	5	1921	0.4150	60.936
0.913000	3	0.101	-14.850	1	Gati Bali	5	1921	0.0339	110.339
0.000028	5	0.381	-9.316	1	Danny Boy	3	1921	0.0354	100.109
0.000002	3	0.229	-10.096	1	When Irish Eyes Are Smiling	2	1921	0.0380	101.665

The second dataset used for this project is a Spotify listening history dataset. Two files exist in this dataset. The first is a text file that has the user id, song id, and the number of listens for that song by that user. This text file has over one million entries, however, through the data analysis, it was broken down into the listening histories of 76,353 unique users for 10,000 unique Spotify songs. The second file included in this dataset is the Spotify song data CSV. This file provides the song title, release, artist name, and year released given the unique song id that will be extracted from the text file. This file has nearly one million unique Spotify songs. Of course, since the text file has a listening history of only 10,000 songs the full size of this dataset will not be utilized. The link and the first five rows of each file in this dataset are below.

Dataset for Collaborative Filtering



user_id	song_id	num_listens
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOAKIMP12A8C130995	1
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBBMDR12A8C13253B	2
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBXHDL12A81C204C0	1
b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBYHAJ12A6701BF1D	1



song_id	title	release	artist_name	year
SOQMMHC12AB0180CB8	Silent Night	Monster Ballads X-Mas	Faster Pussy cat	2003
SOVFVAK12A8C1350D9	Tanssi vaan	Karkuteillä	Karkkiautomaatti	1995
SOGTUKN12AB017F4F1	No One Could Ever	Butter	Hudson Mohawke	2006
SOBNYVR12A8C13558C	Si Vos Querés	De Culo	Yerba Brava	2003

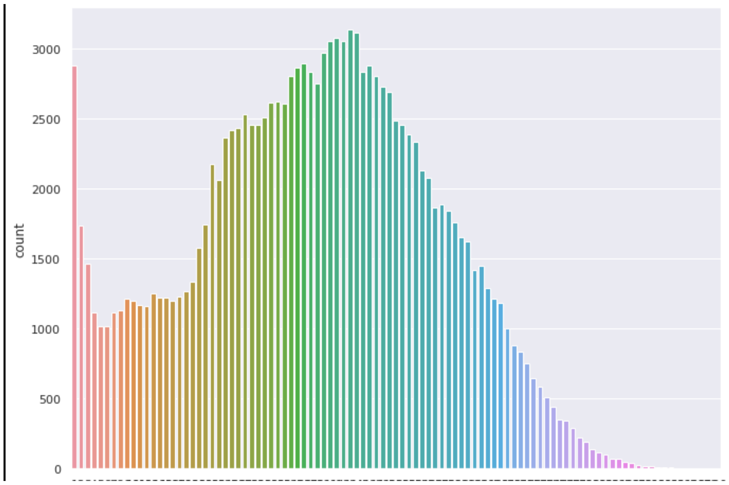
Real World

As briefly covered in the introduction there are numerous real-world applications of this particular idea. The amount of data in the world is ever-growing and methods that can accurately navigate this data to provide the end-user with meaningful results are needed now more than ever. Recommendation systems currently exist everywhere, making life for people more convenient on a daily basis. There are endless resources available for information regarding recommendation systems. The beauty of a recommendation system, however, is the growth potential. There is no “right” answer when providing a recommendation. Metrics and methods will constantly be explored to try to improve recommendation systems, they will be an ever-growing and expanding concept because their real-world applications are so important to people on a daily basis.

Exploratory

Exploratory analyses were performed on each of the used datasets. This allowed the team to dive deeper into the data and understand key features and how things are distributed.

Let us begin with the exploratory analysis of the Spotify song dataset used for the K Nearest Neighbors method. The analysis began with some simple calculations to view the means and standard deviations of each of the attributes in the dataset. Nothing jumped out from these observations, the only notable measurements were the standard deviations of both the instrumentalness and spechiness, both around two times the size of their average. These metrics refer to the instruments and vocals present in a song, this makes the relatively high standard deviation understandable given that there are songs with no vocals or no instruments and visa versa. Then an analysis of the popularity was performed. The distribution of all the songs’ popularity had a resemblance to a bell curve, peaking around a popularity of 40-45, on a scale from 0-100. The exception was the large number of songs that had zero popularity, over 27,000 or 16% of songs in the dataset had a popularity of zero. The distribution of the popularity can be seen on the graph to the right. The next step was observing the distribution of songs considering the decade of their release. There was a very even distribution of songs from the 1950s to the 2010s, each decade having slightly under 20,000 songs. The remaining three decades in the dataset fell short of this: the 1920s with ~5,000 songs, the 1930s with ~10,000, and the 2020s with ~2,000.



As per above, 16% of the songs in the dataset have a popularity of zero. The means and standard deviations were recalculated for this subset of data to compare with the dataset as a whole to try to understand what features may be causing this low popularity rating. Compared with the entire dataset the songs with zero popularity had much higher averages for acoustic ness, instrumentallness, and speechiness, with a much lower average for energy. To better understand these observations we further analyzed the distribution of popularity across the decades. There was a clear upwards trend with songs released in the 2020s having an average rating of around 65 while songs from the 1920s - 1940s had an average rating of around 2. This made it clear that some normalization was needed before creating any viable recommendations. This preliminary analysis of the dataset provided a foundation for sanity checks when creating recommendations with K Nearest Neighbors.

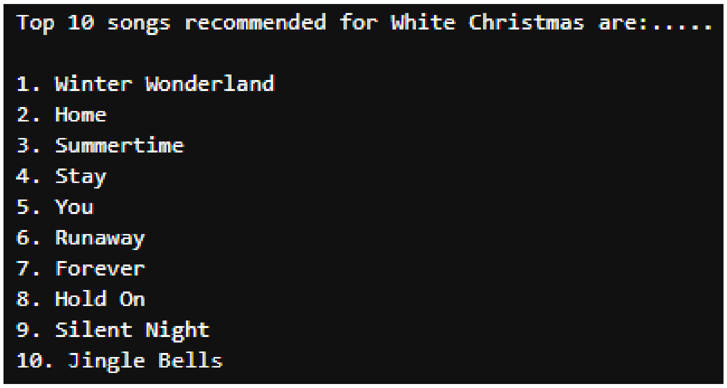
As for the second dataset we first dived into the text file containing information about the users' listening histories. Some basic numbers were gathered, the dataset contains 76,353 unique Spotify users, 10,000 unique Spotify songs, and 6,090,969 song listens. The most popular songs in this dataset were then looked at. The most popular song in this dataset, with over 54,000 listens is You're The One by Dwight Yoakam. The next most popular songs came in with listening counts at around 49,000, 41,000, 31,000, and 26,000. Following this, the most active users in the dataset were looked at. The most active user came in with a listening count of ~4,400, this was 1,000 more than any other user in the dataset and well above the average of 79 listens per unique user. Additionally, we observed that a single user listened to the song "Starshine" by Gorillaz 2,213 times. This is over five days straight of listening to a single song. This made it clear that some outliers existed in this dataset. These observations provided the necessary information to normalize the dataset as necessary. This ensured we could create the best system for creating good song recommendations considering users' listening histories.

Methods

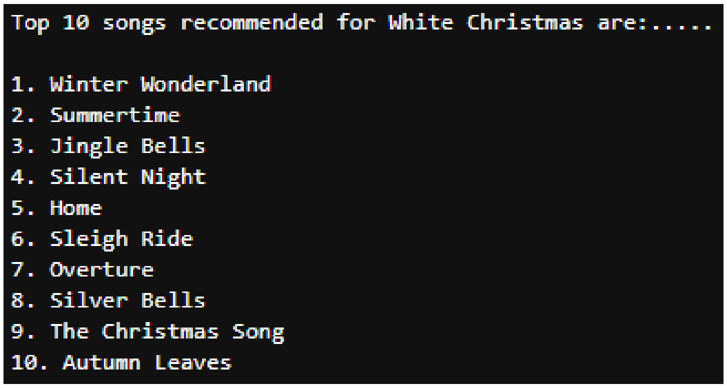
For this project, two Advanced-Data Science methods are being explored. The first method is the K Nearest Neighbor classifier. Generally, this method is applied with classification problems and assigning labels to unknowns. However, for this implementation, we can observe how similar certain songs are to one another and use that as a basis for recommendations. This is considered a Content-Based recommendation. Where if a user likes a certain song, we analyze that song's attributes and metrics, provided by our first dataset, and recommend the user songs that have the most similar attributes. This is the perfect model choice for content-based recommendations. The second method is Collaborative Filtering. This method is well known for its popularity and efficiency in recommendation systems. It is particularly strong for user-based recommendations. A collaborative filter can effectively analyze the listening history of a user, find a user with the most similar listening history, and provide recommendations based on songs that the similar user has listened to. Logically it is very straightforward and highly effective for how it is going to be implemented on this dataset. Both of these models can be accompanied by different measures for similarity/distances. For the KNN we are choosing to operate with the euclidean distance and for the Collaborative Filter we are working with the cosine similarity, both the centered and uncentered variants.

Results

For the K Nearest Neighbors content-based recommendations we observed promising results. The model we created allows the user to specify the specific song features/attributes in order to create a recommendation. This resulted in the exploration of recommendations for the same unique song but with different features of the song each time. Let’s consider a song most people are familiar with, “White Christmas”. The first list of recommendations for “White Christmas” only considers the song’s popularity. These results have a few songs that logically make sense to recommend. The other Christmas songs like Winter Wonderland, Silent Night, and Jingle Bells. However, a few songs seem they shouldn’t be on this list. Nevertheless, this is the recommendation only considering the popularity of “White Christmas”. Now let's look at the results of the recommendations when using all of the song’s attributes (popularity, valence, acousticness, etc.). This set of results is much more appealing. The other Christmas songs on the list are bummed up as higher recommendations. Even some new Christmas songs have appeared on this list compared to the previous recommendation: Sleigh Ride, Silver Bells, and The Christmas Song. We consistently observed a better-performing recommendation system when considering all of the attributes of a single song. This behavior was expected and acted as a sanity check for the recommendations.

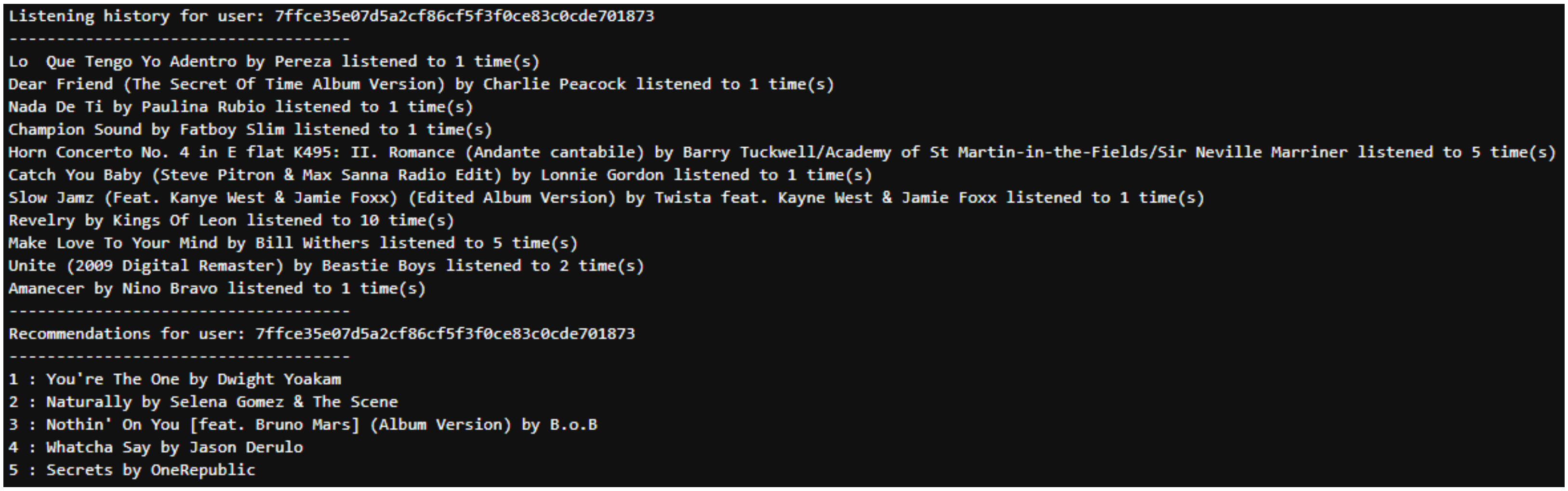


```
Top 10 songs recommended for White Christmas are:.....

1. Winter Wonderland
2. Home
3. Summertime
4. Stay
5. You
6. Runaway
7. Forever
8. Hold On
9. Silent Night
10. Jingle Bells
```



```
Top 10 songs recommended for White Christmas are:.....

1. Winter Wonderland
2. Summertime
3. Jingle Bells
4. Silent Night
5. Home
6. Sleigh Ride
7. Overture
8. Silver Bells
9. The Christmas Song
10. Autumn Leaves
```

For the Collaborative Filter user-user-based recommendations our results were not as promising as the KNN. The quality of these results can be attributed to the dataset itself. We learned that creating good recommendations considering the number of times a user listened to the song is not optimal. Ideally, for a collaborative filter, you want to make recommendations based on a user’s rating of an item. This is because a rating is usually on a hard set scale, for example, 0-5. Meanwhile, a user can listen to a song however many times they like, the scale for song listens can be potentially zero to infinity. This results in an interesting scaling problem for providing recommendations considering listening count. There are two potential solutions to this problem: find a new dataset with user ratings or come up with a method to scale the listening counts. Firstly, we were unable to find any Spotify datasets that contain user ratings, likely because a user cannot rate a song on Spotify. Secondly, we could not come up with an effective way to scale the listening counts for the users. That being said, our results still have some success, however, it is clear that they could be better considering these factors. Below is the listening history of a user in the database and the recommendations provided by our collaborative filter.



```
Listening history for user: 7ffce35e07d5a2cf86cf5f3f0ce83c0cde701873
------------------------------------
Lo  Que Tengo Yo Adentro by Pereza listened to 1 time(s)
Dear Friend (The Secret Of Time Album Version) by Charlie Peacock listened to 1 time(s)
Nada De Ti by Paulina Rubio listened to 1 time(s)
Champion Sound by Fatboy Slim listened to 1 time(s)
Horn Concerto No. 4 in E flat K495: II. Romance (Andante cantabile) by Barry Tuckwell/Academy of St Martin-in-the-Fields/Sir Neville Marriner listened to 5 time(s)
Catch You Baby (Steve Pitron & Max Sanna Radio Edit) by Lonnie Gordon listened to 1 time(s)
Slow Jamz (Feat. Kanye West & Jamie Foxx) (Edited Album Version) by Twista feat. Kayne West & Jamie Foxx listened to 1 time(s)
Revelry by Kings Of Leon listened to 10 time(s)
Make Love To Your Mind by Bill Withers listened to 5 time(s)
Unite (2009 Digital Remaster) by Beastie Boys listened to 2 time(s)
Amanecer by Nino Bravo listened to 1 time(s)
------------------------------------
Recommendations for user: 7ffce35e07d5a2cf86cf5f3f0ce83c0cde701873
------------------------------------
1 : You're The One by Dwight Yoakam
2 : Naturally by Selena Gomez & The Scene
3 : Nothin' On You [feat. Bruno Mars] (Album Version) by B.o.B
4 : Whatcha Say by Jason Derulo
5 : Secrets by OneRepublic
```

Conclusion

In conclusion, the goal of this project was met. A Spotify recommendation system was successfully implemented while exploring the different types of recommendations (content-based and user-based). While we ran into some problems and in-efficiencies with our dataset selection. We understood the mistakes we made and how we could improve upon them in future endeavors. Through this project, we concluded that a mixture of content-based and user-based recommendations will likely yield the best results. Future work for this project could include the combination of our two datasets to find similar users and then use content-based recommendations on the users' listening histories. Additionally, the team succeeded in learning more about recommendation systems, how they work, and how important they are in the current world we live in. Further exploration is necessary to really answer the "what methods create the "best" recommendations?" question.