International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 12 | December - 2022 Impact Factor: 7.185 ISSN: 2582-3930

APPLICATION OF MUSIC RECOMMENDATION SYSTEM

T. Sreekar, J. Rahul
Mr. D. Rambabu (Guide)
Sreenidhi Institute of Science and Technology
Hyderabad

ABSTRACT

A music recommendation system using content-based filtering relies on the characteristics of the music itself to make recommendations to users. This approach involves analyzing the attributes of each song in the system's database, such as the genre, artist, and tracks. When a user inputs a song name, the recommendation system uses these attributes to identify similar songs and make recommendations based on the user's input. This type of recommendation system is useful for users who have specific music preferences and want to discover new artists or songs within their preferred genres. It is also effective at personalized recommendations, as it takes into account the individual characteristics of each user .

INTRODUCTION

Music recommendation systems play a significant role in helping listeners discover new music and artists that align with their preferences. These systems use various approaches to generate recommendations, including collaborative filtering, which relies on the listening history of other users, and content-based filtering, which focuses on the characteristics of the music itself.

In this paper, we will explore the use of content-based filtering for music recommendation systems. This approach involves analyzing the attributes of each song in the system's database, such as the genre, artist, and lyrics, and using these attributes to identify similar songs and make recommendations based on a user's input. Content-based filtering is particularly useful for personalized recommendations,

as it takes into account the individual characteristics and preferences of each user.

CONTENT-BASED FILTERING

Content-based filtering is a technique used in recommendation systems that relies on the characteristics of the items being recommended to make recommendations. In the context of music recommendation systems, this means that the system uses the attributes of each song in its database, such as the genre, artist, and lyrics, to identify similar songs and make recommendations to users.

Content-based filtering is based on the idea that a user's preferences can be determined based on the characteristics of the items they have liked in the past. For example, if a user consistently listens to songs by a particular artist or in a particular genre, the recommendation system might suggest other songs by that artist or in that genre.

Content-based filtering is often contrasted with collaborative filtering, which relies on the behavior of other users to make recommendations. Collaborative filtering systems make recommendations based on the items that similar users have liked in the past, rather than the characteristics of the items themselves. Both approaches have their own strengths and limitations, and recommendation systems may use a combination of both approaches to generate recommendations.

Cosine Similarity

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. It is defined as the dot product of the vectors divided by the product of the magnitudes of the vectors.

In the context of recommendation systems, cosine similarity is often used to measure the similarity between different items based on the characteristics of those items. For example, in a music recommendation system, cosine similarity might be used to compare the characteristics of different songs, such as their genre and artist order to identify similar songs.

To calculate cosine similarity, the vectors representing the items being compared must be represented in a common space, such as a high-dimensional space where each dimension corresponds to a specific characteristic of the item. The dot product of the vectors is then calculated, and the result is divided by the product of the

magnitudes of the vectors to obtain the cosine similarity.

Cosine similarity is often used in recommendation systems because it is relatively easy to calculate and can handle sparse data, meaning that it can work well even when some characteristics of the items being compared are not present or are unknown. It is also relatively robust to changes in the scale of the vectors, meaning that it can work well even when the characteristics of the items are measured on different scales.

cosine similarity = (A * B) / (||A|| * ||B||)

where:

- A and B are the two vectors being compared
- (A * B) is the dot product of the vectors, which is calculated by summing the product of the corresponding elements in the vectors
- ||A|| and ||B|| are the magnitudes of the vectors, which are calculated as the square root of the sum of the squares of the elements in the vectors

For example, if we have two vectors A and B with elements [a1, a2, a3] and [b1, b2, b3] respectively, the cosine similarity would be calculated as:

cosine similarity = ([a1, a2, a3] * [b1, b2, b3]) / (sqrt(a1^2 + a2^2 + a3^2) * sqrt(b1^2 + b2^2 + b3^2))

The resulting value will be a number between -1 and 1, with a value of 1 indicating that the vectors are perfectly similar, a value of 0 indicating that the vectors are orthogonal (perpendicular) to each other, and a value of -1 indicating that the vectors are perfectly dissimilar.

LIMITATIONS

There are several limitations to using cosine similarity for recommendation systems:

- Limited context: Cosine similarity only considers the characteristics of the items being compared, and does not take into account any additional context or information about the user or the items. This means that it may not be able to make recommendations that are tailored to a specific user or situation.
- High-dimensional space: In order to compare items using cosine

similarity, they must be represented in a high-dimensional space, where each dimension corresponds to a specific characteristic of the item. This can be computationally expensive, especially if the number of dimensions is large.

- Sparsity: Cosine similarity can handle sparse data, but it can be less effective in situations where the vectors being compared have few common dimensions. In these cases, the resulting similarity scores may not accurately reflect the similarity of the items.
- Scale issues: Cosine similarity is relatively robust to changes in the scale of the vectors, but it may still be affected by differences in the scale of the characteristics being compared. For example, if one characteristic is measured in a different unit than another, the resulting similarity scores may not be meaningful.
- Non-linear relationships: Cosine similarity assumes that the relationship between the dimensions of the vectors is linear, which may not always be the case. This can lead to inaccurate similarity scores in situations where the relationship between the dimensions is non-linear.

SOFTWARE REQUIREMENTS

- Jupyter Notebook
- Python

Operating System: Windows 7 and above or Linux based OS or MACOS.

Modules used: numpy, pandas, scikitlearn, wordcloud, matplotlib.

HARDWARE REQUIREMENTS

- RAM: 4 GB
- Storage: 500 GB
- CPU: 2 GHz or faster
- Architecture: 32-bit or 64-bit

CONCLUSION

The output is mostly dependent on the features like acousticness, liveliness, danceability, loudness , tempo etc, and also it is dependent on popularity, genres and release date. The data set's available genres include pop, edm, indie , jazz, hip-hop, rock etc. Based on content-based filtering, the

system suggests songs the user might enjoy and adds such songs to the user's playlist.

FUTURE SCOPE

- Improving the accuracy of recommendations: There is still room for improvement in the accuracy of recommendations made by content-based music recommendation systems. Researchers could explore new algorithms or techniques for more effectively analyzing and comparing the characteristics of different songs to identify similar songs.
- Incorporating additional data sources: Currently, music recommendation systems often rely on a limited set of characteristics for making recommendations, such as the genre, artist, and lyrics of a song. Adding additional data sources, such as information about the production or arrangement of the song, could improve the accuracy and personalization of recommendations.
- Enhancing user control: Some users may prefer to have more control over the recommendations they receive, such as the ability to exclude certain artists or genres from the recommendations. Future recommendation systems could allow users to customize their recommendations in more granular ways.
- Combining content-based and collaborative filtering: Content-based and collaborative filtering are often used as separate approaches for recommendation systems, but combining the two approaches could potentially lead to even more accurate and personalized recommendations. Researchers could explore ways to effectively combine the two approaches to create more powerful recommendation systems.
- Incorporating context and real-time factors: Recommendation systems could be improved by incorporating real-time context and other factors, such as the user's location, the time of day, and the user's mood. This could allow the system to make recommendations that are more relevant and timely for the user

ACKNOWLEDGMENTS

We would like to express our sincere gratitude to our mentor Mr. D. Rambabu for providing us with the fantastic chance to work on this amazing project on this subject. This project also enabled us to do extensive research, which allowed us to learn a lot of brand-new information. They have our sincere appreciation.

REFERENCES

- Michael D Ekstrand, John T Riedl, and Joseph A Konstan.Content filtering recommender systems. Foundations and Trends in Human-Computer Interaction, 4(2):81–173, 2011.

- Yifan Hu, Yehuda Koren, and Chris Volinsky. Content filtering for implicit feedback datasets. In IEEE International Conference on Data Mining (ICDM 2008, pages 263–272,2008.

- Paul B Kantor, Lior Rokach, Francesco Ricci, and Bracha Shapira. Recommender systems handbook. Springer, 2011.

- Brian McFee, Thierry Bertin-Mahieux, Daniel PW Ellis, andGert RG Lanckriet. The million song dataset challenge. In Proceedings of the 21st international conference companion on the World Wide Web, pages 909–916. ACM, 2012.

- J Ben Schafer, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. In Proceedings of the 1st ACM conference on Electronic commerce, pages 158–166. ACM,1999.

AUTHORS

First Author - T. Sreekar,
B-tech Final year , CSE
Second Author - J. Rahul,
B-tech Final year , CSE