Music Recommendation System Using Cosine Similarity

Contents

I. Introduction

The improvement in computerized innovation has driven the music carefully accessible to all Internet clients. The advancement of nanotechnology made capacity gadgets convenient, and these days, any handheld gadgets can store a large number of tracks. At whatever point a client has an enormous number of decisions for tuning in to music (like perusing web or individual stockpiling gadgets), the client is overpowered by alternatives.

The recommender framework goes about as a friend in need and channels the melodies that are reasonable for that client at that point. It likewise expands the client's fulfilment by playing fitting tune at the correct time, and, in the interim, limit the client's work.

The suggestion issue can be viewed as a positioning issue, and it makes a rundown of reasonable tunes for clients. Numerous music real time features for Western music arose lately, for example, Google Play Music, Apple music, Last.fm, Pandora, Spotify, etc; and some of them are not accessible in India. These music streaming applications store client inclinations and prescribe clients what they need to tune in. In India, a few music web-based features were begun as of late and those are Apple music, Gaana , Hungama, Saavn and Wynk music and so forth. Majority of them do not suggest melodies and those are only a music library. YouTube is one of the video web-based features which gives suggestions dependent on the cooperative sifting. It additionally gives office to look through utilizing title of melody and it can likewise look through a video utilizing any catchphrases inside a verse body just when full verse is free in the depiction. There is a distinct fascination for getting to music substance these days.

Accessible web crawlers or data recovery (IR) frameworks permit clients to look through a tune by the metadata, for example, melody title, craftsman, collection name. Inaccurate metadata can prompt wrongly looked through information, and with no metadata, it is absurd to expect to look through a tune. Once more, the flow IR frameworks give specific list items dependent on the question instead of comparative verses to an inquiry.

II. Literature Review

One of the most used machine learning algorithms is recommendation systems. A recommender (or recommendation) system (or engine) is a filtering system whose aim is to predict a rating or preference a user would give to an item, for example, a film, a product, a song, etc. There are two main types of recommender systems: Content-based filters and Collaborative filters. Content-based filters predicts what a user likes based on what that particular user has liked in the past. On the other hand, collaborative-based filters predict what a user like based on what other users, that are similar to that particular user, have liked. Recommendations done using content-based recommenders can be seen as a user-specific classification problem. This classifier learns the user's likes and dislikes from the features of the song. Presently, millions of music tracks are available on the web. Therefore, a music recommendation system can be helpful to filter and organize music tracks according to the need of users. To develop a recommendation system, we need an enormous amount of data along with the user preference information. The idea is to present a lyrics retrieval system for Western songs using features collected from lyrics. An English lyrics dataset is collected from the web. The collected dataset is noisy, and several forms of a single word are present in it, thus it required filtering to remove the discrepancies and make the dataset uniform.

III. Applications of Cosine Similarity

Cosine Similarity has its place in a few applications and calculations:
From the universe of PC vision to information mining, there is heaps of handiness to looking at a likeness estimation between two vectors addressed in a higher-dimensional space.
How about we go two or three situations and applications where the cosine similarity measure is utilized.

1. **Document Similarity**

    - A situation that includes the prerequisite of distinguishing the similarity between sets of an archive is a decent use case for the usage of cosine comparability as an evaluation of the estimation of likeness between two items.

    - The conventional way to deal with figure text similarity between records is to do as such by changing the information reports into genuine esteemed vectors. The objective is to have a vector space where comparative reports are "close", as per a picked similarity measure.

    - This methodology takes the name of Vector Space Model, and it's advantageous in light of the fact that it permits us to utilize straightforward direct polynomial math to process similarity.

2. **Pose Matching**

    - Posture coordinating includes looking at the stances containing key marks of joint areas.

    - Posture assessment is a PC vision assignment, and it's commonly tackled utilizing Deep Learning approaches like Convolutional Pose Machine, Stacked hourglass, PoseNet and so on.

    - Posture assessment is the interaction where the position and direction of the crucial body parts and joints of a body are gotten from a picture or succession of pictures.

3. **Bioinformatics**

    - Discovering comparative successions to an info question arrangement (DNA or proteins) from a grouping informational index is a significant issue in bioinformatics. It gives analysts an instinct of what could be connected or how the hunt space can be decreased for additional assignments.

    - An arrangement free strategy-based closeness estimates like cosine comparability and squared Euclidean distance by addressing groupings as vectors was researched. The cosine-similarity based region delicate hashing method was utilized to lessen the quantity of pairwise examinations while discovering comparative arrangements to an info inquiry.

4. Recommendation Systems

A recommender framework (or motor) is a sifting framework which point is to foresee a rating or inclination a client would provide for a thing, e.g., a film, an item, a tune, and so forth.

There are two fundamental kinds of recommender frameworks:

- Content-based filters:
- Collaborative filters:

Content-based techniques are computationally quick and interpretable. Also, they can be productively adjusted to new things or clients. Notwithstanding, perhaps the greatest constraint of substance-based suggestion frameworks is that the model just figures out how to suggest things of the very kind that the client is now utilizing or, for our situation, tuning in to. Despite the fact that this could be useful, the worth of that proposal is fundamentally less on the grounds that it comes up short on the unexpected part of finding something totally new.

Collaborative-based methods work with an association framework, likewise called rating lattice. The point of this calculation is to become familiar with a capacity that can foresee if a client will profit by an item-meaning the client will probably purchase, tune in to, watch this thing.

Content-based methods gives suggestions based with respect to the similarity of two tune substance or characteristics while cooperative strategies make a forecast on potential inclinations utilizing a framework with appraisals on various tunes.

IV. Methodology

i. Proposals done utilizing content-based recommenders can be viewed as a client explicit characterization issue. This classifier learns the client's preferences from the highlights of the tune. The clearest methodology is keyword matching.

ii. In a couple of words, the thought behind is to extricate significant catchphrases present in a melody portrayal a client likes, look for the keywords in other tune depictions to assess similarity among them, and dependent on that, prescribe those tunes to the client.

iii. For our situation, since we are working with text and words, Term Frequency-Inverse Document Frequency (TF-IDF) can be utilized for this coordinating with measure. The occasions a term happens in an archive is called its term recurrence (TF).

iv. tf(t,d) = include of t in d/number of words in d

v. While figuring TF, all terms are considered similarly significant. Anyway, it is realized that specific terms, for example, "is", "of", and "that", may seem a great deal of times yet have little significance. Along these lines we need to burden the regular terms while increase the uncommon ones, by processing IDF, a reverse archive recurrence factor is consolidated which reduces the

heaviness of terms that happen habitually in the report set and builds the heaviness of terms that happen infrequently.

vi. **idf(t) = log(N/(df + 1))**

tf-idf presently is the correct measure to assess how significant a word is to a report in an assortment. It doles out a load to each word in the record, which is determined utilizing the recurrence of that word in the report and recurrence of the archives with that word in the whole corpus of reports.

vii. After this we utilize the cosine similarity library to discover the similarity between verses of various tunes. Cosine Similarity is an estimation that measures the closeness between at least two vectors. The cosine similarity is the cosine of the point between vectors.

viii. A situation that includes the necessity of distinguishing the similarity between verses of a tune is a decent use case for the usage of cosine similarity as an evaluation of the estimation of similarity between two articles.

ix. Evaluation of the similarity between two verses can be acquired by changing over the words or expressions inside the sentence into a vectorised type of portrayal. The vector portrayals of the records would then be able to be utilized inside the cosine comparability equation to acquire a measurement of closeness.

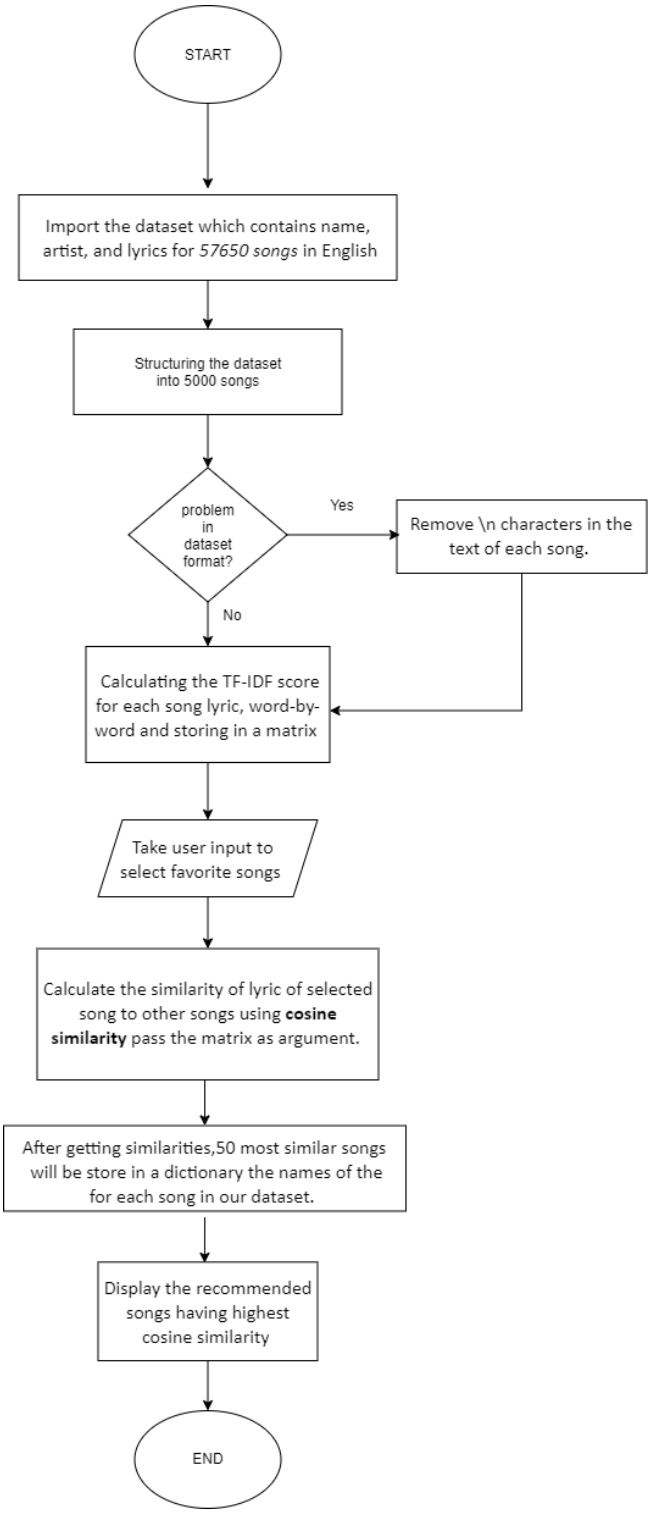
x. The cosine similarity of 1 infers that the two verses are by and large indistinguishable and a cosine similarity of 0 would highlight the end that there are no similarities between the two verses.

V. Algorithm

- Import the dataset which contains name, artists, and verses for 57650 tunes in English by utilizing pandas library and store it in an information outline.
- On account of the dataset being so huge, we are going to resample just 5000 random tunes.
- Eliminate '\n' characters in the content of every tune.
- Utilizing TF-IDF vectorizer we will figure the TF-IDF score for every melody verse, word-by-word and store in a grid.
- Figure the similarity of one verse to another utilizing cosine comparability pass the framework as contention.
- When we get the similarity, we'll store in a word reference the names of the 50 most comparable melodies for every tune in our dataset.
- Take client contribution to choose a melody and for number of proposals he needs to see.

- Show the suggested melodies having most noteworthy cosine similitude.

VI. Flowchart



VII. Implementation

1. **Tools:**
   Jupyter Notebook

2. **Library:**
   - NumPy
   - Pandas
   - Sklearn
     - TfidfVectorizer
     - cosine_similarity

Dataset

songs.csv

	A	B	C	D
1	artist	song	link	text
2	ABBA	Ahe's My K	/a/abba/ahes+my+kind+of+girl_20598417.html	Look at her face, it's a wonderful face
3	ABBA	Andante, A	/a/abba/andante+andante_20002708.html	Take it easy with me, please
4	ABBA	As Good A	/a/abba/as+good+as+new_20003033.html	I'll never know why I had to go
5	ABBA	Bang	/a/abba/bang_20598415.html	Making somebody happy is a question of give and take
6	ABBA	Bang-A-Bo	/a/abba/bang+a+boomerang_20002668.html	Making somebody happy is a question of give and take
7	ABBA	Burning M	/a/abba/burning+my+bridges_20003011.html	Well, you hoot and you holler and you make me mad
8	ABBA	Cassandra	/a/abba/cassandra_20002811.html	Down in the street they're all singing and shouting
9	ABBA	Chiquitita	/a/abba/chiquitita_20002978.html	Chiquitita, tell me what's wrong
10	ABBA	Crazy Wor	/a/abba/crazy+world_20003013.html	I was out with the morning sun
11	ABBA	Crying Ove	/a/abba/crying+over+you_20177611.html	I'm waitin' for you baby
12	ABBA	Dance	/a/abba/dance_10001507.html	Oh, my love it makes me sad.
13	ABBA	Dancing Q	/a/abba/dancing+queen_20002554.html	You can dance, you can jive, having the time of your life
14	ABBA	Disillusion	/a/abba/disillusion_20002786.html	Changing, moving in a circle
15	ABBA	Does Your	/a/abba/does+your+mother+know_20003035.html	You're so hot, teasing me
16	ABBA	Dream Wo	/a/abba/dream+world_20003014.html	Agnetha We're not the stars of a Hollywood movie
17	ABBA	Dum Dum	/a/abba/dum+dum+diddle_20002555.html	I can hear how you work, practising hard
18	ABBA	Eagle	/a/abba/eagle_20002818.html	They came flying from far away, now I'm under their spell
19	ABBA	Every Goo	/a/abba/every+good+man_20336091.html	Every good man needs a helping hand
20	ABBA	Fernando	/a/abba/fernando_20002896.html	Can you hear the drums Fernando?
21	ABBA	Fernando (	/a/abba/fernando+in+spanish_20930420.html	Puedes escuchar Fernando?
22	ABBA	Free As A E	/a/abba/free+as+a+bumble+bee_20003016.html	I'm down and I feel depressed
23	ABBA	From A Tw	/a/abba/from+a+twinkling+star+to+a+passing+angel_21001	Twinkle, Twinkle little star
24	ABBA	Gimme Gir	/a/abba/gimme+gimme+gimme_10001506.html	Half past twelve
25	ABBA	Givin' A Lit	/a/abba/givin+a+little+bit+more_20003017.html	There's a gentleness to everything you do
26	ABBA	Gonna Sing	/a/abba/gonna+sing+you+my+lovesong_20002968.html	You say she's been mad at you
27	ABBA	Hamlet III	/a/abba/hamlet+iii_20598413.html	When the autumn leaves are falling to the ground

VIII. Results and Discussion

1. We executed music recommendation framework in python utilizing cosine similarity that suggests tunes dependent on the lyrics. Series of words are changed over into vectors that are then coordinated with the data set of melodies to suggest the most fitting tune that the client will like.

2. Cosine Similarity is more helpful for occurrences when you don't need extent to slant the outcomes. This is generally valuable in word vectorization on the grounds that normalizing the information makes a long line of text practically identical to a short book. The Euclidian distance will be huge when between writings of various word length which would slant your outcomes.

3. Coming up next are our decisions dependent on explore results. To start with, music recommender framework ought to consider the music type data to expand the nature of music suggestions. Second, CRNNs that considers both the recurrence highlights and time succession designs has generally speaking better execution. It demonstrates the viability of its mixture design to separate the music highlights. In view of our examinations, we can propose for future exploration to add other music highlights to improve the precision of the recommender framework, for example, utilizing rhythm gram for catching neighbourhood beat at a specific time.

4. As you can notice, content-based techniques are computationally quick and interpretable. In addition, they can be versatile and effectively adjusted to new things or clients.

5. In addition, the produced suggestions are somewhat not one-sided towards notable or well-known things. Nonetheless, perhaps the greatest limit of substance-based proposal frameworks is that the model just figures out how to suggest things of the very kind that the client is now utilizing or, for our situation, tuning in to. Despite the fact that this could be useful, the worth of that suggestion is altogether less on the grounds that it comes up short on the unexpected segment of finding something totally new. Last, this kind of proposal framework can't catch any relevant or social data. Just data that can be gotten from the substance of a melody can be acquired and used to create proposals.

6. In future, the inter-cluster cosine similarity can be utilized for programmed assessment. A weighted score can be doled out to each part of verses (beginning, center, and end) for assessment.

7. The temperament words from verses can be gathered utilizing solo methodology, for example, word embeddings and the inferred disposition data can be utilized for positioning the aftereffects of comparative verses recovery framework.

8. Positioning the recovered verses is another critical factor for suggestion framework and isn't considered during the assessment of current framework. It is one of the restrictions of current created framework and positioning based assessment can be carried out in future.

IX. References

1. S Bandyopadhyay, D S Sharma and R Sangal, “Retrieving Similar Lyrics for Music Recommendation System” (2017), Proc. of the 14th Intl. Conference on Natural Language Processing, pages 290–297.
2. Maturu, Srikanth, "Application of Cosine Similarity in Bioinformatics" (2018). Computer Science and Engineering: Theses, Dissertations, and Student Research. 153
3. Harika Abburi, Eswar S. A. Akkireddy, Suryakanth Gangashetti, and Radhika Mamidi. 2016. Multimodal sentiment analysis of telugu songs. In Proceedings of the 4th Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2016), pages 48–52.
4. Jayson Beaster-Jones and Natalie Sarrazin. 2016. Music in Contemporary Indian Film: Memory, Voice, Identity. Taylor & Francis.
5. James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, et al. 2010. The youtube video recommendation system. In Proceedings of the fourth ACM conference on Recommender systems, pages 293–296. ACM.
6. Yong Han, Li Min, Yu Zou, Zhongyuan Han, Song Li, Leilei Kong, Haoliang Qi, Wenhao Qiao, Shuo Cui, and Hong Deng. 2015. Lrc sousou: A lyrics retrieval system. In Proceedings of the International Conference of Young Computer Scientists, Engineers and Educators, pages 464–467. Springer.
7. Xiao Hu, J Stephen Downie, and Andreas F Ehmann. 2009. Lyric text mining in music mood classification. In Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR 2009), pages 411–416.
8. Yajie Hu. 2014. A Model-Based Music Recommendation System for Individual Users and Implicit User Groups. Ph.D. thesis, University of Miami.