DR. AKHILESH DAS GUPTA INSTITUTE OF TECHNOLOGY & MANAGEMENT

(Shastri Park New Delhi-110053)

MUSIC RECOMMENDATION SYSTEM

SUBMITTED BY:

KUNAL [CSE 3rd YEAR - 06515602721]
HARSHIT [CSE 3rd YEAR - 06615602721]
MOHIT BADHANI [CSE 3rd YEAR - 14715602721]
MAYANK SAGAR [CSE 3rd YEAR - 05615602721]

ACADEMIC YEAR 2023/24

DECLARATION

We, the undersigned members of this project conducted as part of the course at **Dr. Akhilesh das Gupta Institute of Technology & Management**, collectively declare that this project is the culmination of our original work. It has been developed in accordance with the requirements for the modern era, and it has not been previously submitted for any other assessment or purpose.

We jointly declare the following:

The project work presented in this document is the result of our combined effort and original input. We have duly acknowledged the contributions of each team member where applicable.

Any assistance, guidance, references, or resources obtained from external sources have been properly cited and acknowledged in this document.

The project work has not been submitted by any of us or any other individuals for assessment in any academic institution.

All sources of information, data, code snippets, and other materials utilized in this project have been appropriately referenced and cited.

Any tools, frameworks, software libraries, or third-party resources employed in the project have been cited and credited accordingly.

We are fully aware that any form of plagiarism, academic dishonesty, or misconduct in this project could result in severe consequences, as outlined by the academic policies of [ADGITM].

CERTIFICATE

This is to certify that the following project has been successfully completed and verified by the undersigned. The project titled predict diabetes using machine learning was undertaken by **Kunal, Harshit Verma, Mohit Badhani, Mayank Sagar** as part of their academic pursuits at **Dr.Akhilesh Das Gupta Institute of Technology &Management**. The project was completed in accordance with the guidelines and requirements set forth by the academic institution.

PROJECT DETAILS:

TITLE: MUSIC RECOMMENDATION SYSTEM

STUDENTS: Kunal, Harshit Verma, Mohit Badhani, Mayank Sagar

COLLEGE: ADGITM

PROGRAM & BRANCH : B. TECH (Computer Science And Engineering)

The project was thoroughly reviewed and assessed by the faculty members. The verification process included an examination of the project documentation, codebase, design artifacts , and a presentation or demonstration of the project's functionality and outcomes.

This verification certificate is issued for the purpose of submission to the college and is subject to the policies and procedures of the academic institution.

ACKNOWLEDGEMENT:

We would like to express our sincere gratitude to all those who have contributed to the successful completion of the project. This endeavour would not have been possible without the support, guidance, and assistance of various individuals and resources. We extend our heartfelt thanks to the following:

Faculty Advisor: We are deeply thankful to Mr. Himanshu, our project guide, for providing us with invaluable guidance and mentorship throughout the project. Their insights and suggestions were instrumental in shaping our project.

The Project based on Unsupervised Machine Learning required the deep knowledge of clustering, k means algorithm, pandas and many other libraries and modules. It was made possible by proper guidance of the faculty and references from internet.

We are sincerely grateful for the contributions of our fellow group mates and everyone else for making it possible.

ABSTRACT:

The growing use of Internet as an information source, has led to the proliferation of technologies to deploy rich web-based applications. Among these applications are present the music service providers. These systems allow users to listen to music without downloading it to the computer. Some use recommendations techniques to improve the user experience. The objective of this project is to develop a music recommendation system.
The system will determine the musical preferences of the users based on the analysis of their interaction during use. This way the system can estimate what artist or group would match user preferences to the user at a given time. It has been considered the fact that we do not always want to hear the same artists or genres, we do have favourite bands, but sometimes we appreciate a surprise, a new discovery.

The system uses music information collected from online music services that make available their music catalogues for developers' community to be used inside new applications. It has been created a web system that connects to the music service providers to obtain these musical catalogues. This system implements the necessary communication features to use this information in the client web browser.
This system helps users discover new artists, albums or songs making the musical catalogue available for listening.

The dynamic characteristics of the interface allows the user to browse music collections while listening to a song or playing a video. The user will receive information related to her interaction patterns in form of recommendations of items. These items will probably match user preferences and they are shown as the user interacts with the system and only when it has enough information about user preferences.

1.1 INTRODUCTION:

The continue increasing in connection speed and trends in web development technologies, given rise to large web systems nowadays visited daily. Among these advanced systems, there are systems that allow users to listen music online without the need of downloading it to their personal computer. This issue solves a big problem originated by peer-to-peer software. There is a wide variety of these systems and new alternatives are constantly emerging increasingly improved.
Some are simple players providing playlist functionality (www.prostopleer.com), others accompany the player with a recommendation system of similar artists (www.spotify.com), also there are complex collaborative systems in which hundreds of people leave comments on songs (www.pandora.com,www.lastfm.com) and have the chance of interact with each other as in the newly emerging social networks. This project has been designed in order to meet some logic-based recommender features. In one hand, due to the need of overcoming the numerical representation of data, needed to use other mathematical approaches, in other hand because it allows the closest data representation domain to this problem.

1.2 MOTIVATION

The motivation of this project is to create an algorithm that can provide a curated playlist based on user's preference resulting in a better Music listening experience to the users. This can help users to discover new music similar to what they are used to listen to but providing something new at the same time.

1.3 STATEMENT OF PROBLEM:

The growing use of Internet as an information source, has led to the proliferation of technologies to deploy rich web-based applications. Among these applications are present the music service providers.
It has been considered the fact that we do not always want to hear the same artists or genres, we do have favourite bands, but sometimes we appreciate a surprise, a new discovery.

1.4 OBJECTIVE OF PROJECT:

The objective of this project is to develop a music recommendation system. The system will determine the musical preferences of users based on the analysis of their interaction during use.

1.5 SIGNIFICANCE OF THE PROJECT:

The project primarily aims at recommending music to user but with emphasis on new music rather than repeating the daily playlist of user, the user have higher chance of loving the music with same genre and mood as they have been listening daily but also want change.

2.1 LITERATURE WORK:

Clustering

In this assignment we performed clustering on the Spotify dataset and develop a recommendation system for the users. We started with a dataset of songs and their features like danceability, energy, etc. Then performed clustering on the dataset and then recommended songs to the users based on the clusters.

Music Recommendation System

In this assignment submission contains a main python code file, new_project.ipynb, we used Jupyter Notebook for the project code file. It takes the path to user preferences (a csv file) and outputs:

● A single playlist file containing all the songs recommended to the users. It is not important how the songs are sampled from the clusters.

->Acousticness: number<float>

A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.

->Danceability: number<float>

Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.

->Duration_ms: integer

The duration of the track in milliseconds.

->Energy: number<float>

Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general
Entropy

->Track_id: string
The Spotify ID for the track.

->Instrumentalness: number<float>
Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content.

->Key: integer
The key the track is in. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C♯/D♭, 2 = D, and soon. If no key was detected, the value is -1.

->Liveness: number<float>
Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.

->Loudness: number<float>
The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typically range between -60 and 0 db.

->Mode: integer
Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.

->Speechiness: number<float>
Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value.

->Tempo: number<float>

The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.

->Time_signature: integer

An estimated time signature. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).

->Track_name: string

Name of the track

->Artist_name: string Name of the artist.

->valence: number<float>

A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track.

Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

->genre: string

Genre of the song.

	artist_name	track_name	track_id	popularity	year	genre	danceability
0	Jason Mraz	I Won't Give Up	1F56cjZA9RTuuMZDrSA6	68	2012	acoustic	0.483
1	Jason Mraz	93 Million Miles	1s8tP3jP4GZcyHDsjvw218	50	2012	acoustic	0.572
2	Joshua Hyslop	Do Not Let Me Go	1Ca8MPiyuvr2VU3O9W0F	57	2012	acoustic	0.409
3	Boyce Avenue	Fast Car	1wsZUhUZLIh1OsyrZq7sz	58	2012	acoustic	0.392
4	Andrew Belle	Sky's Still Blue	16nXlYClvJAfi6ujLiKqEq8	54	2012	acoustic	0.43
5	Chris Smither	What They Say	1vptbNKGs6sPy1Vh1O0v	48	2012	acoustic	0.566
6	Matt Wertz	g in a Winter Wonderland	1SvLAG3URGrEvNNbGM	48	2012	acoustic	0.575
7	Green River Ordinance	Dancing Shoes	1uzQCg9p4yH347Nn8OW	45	2012	acoustic	0.586
8	Jason Mraz	Living in the Moment	17k1L4EkZppZPz1EJWTS	44	2012	acoustic	0.65
9	Boyce Avenue	Heaven	1mYmUdAVXlaHCnnW13o	58	2012	acoustic	0.619
10	Tristan Prettyman	Say Anything	1EDlhTp2iJXWPdLpa8OM	45	2012	acoustic	0.657

energy	key	loudness	mode	speechiness	acousticness	instrumentalness
0.303	4	-10.058	1	0.0429	0.694	0.0
0.454	3	-10.286	1	0.0258	0.477	1.37e-05
0.234	3	-13.711	1	0.0323	0.338	5e-05
0.251	10	-9.845	1	0.0363	0.807	0.0
0.791	6	-5.419	0	0.0302	0.0726	0.0193
0.57	2	-6.42	1	0.0329	0.688	1.73e-06
0.606	9	-8.197	1	0.03	0.0119	0.0
0.423	7	-7.459	1	0.0261	0.252	5.83e-06
0.628	7	-7.16	1	0.0232	0.0483	0.0
0.28	8	-10.238	0	0.0317	0.73	0.0
0.43	1	-10.202	1	0.0314	0.534	0.000238

liveness	valence	tempo	duration_ms	time_signature
0.115	0.139	133.406	240166	3
0.0974	0.515	140.182	216387	4
0.0895	0.145	139.832	158960	4
0.0797	0.508	204.961	304293	4
0.11	0.217	171.864	244320	4
0.0943	0.96	83.403	166240	4
0.0675	0.364	121.083	152307	4
0.0976	0.318	138.133	232373	4
0.119	0.7	84.141	235080	4
0.103	0.292	129.948	250063	4
0.12	0.335	91.967	236379	4

2.2 RELATED WORK:

DATA PREPROCESSING:

1)DATA CLEANING

Removing unnecessary columns:

Columns 'artist_name', 'popularity', 'year', 'duration_ms' and 'genre' are not going to be useful for us, even genre will not appear in our input so it doesn't help that much. We're going to drop all mentioned columns.

Handling missing data:

In columns 'song_name', 'Unnamed: 0' and 'title' we see lots of NaNs, in addition to that they are not going to be helpful for the clustering part, so we dropped these columns too.

It was previously mentioned that if the 'key' is not detected, its value is -1, it might be a good way of handling the missing data, but it can ruin our upcoming clustering.
So, we're going to drop all the rows whose **'key' is -1**.

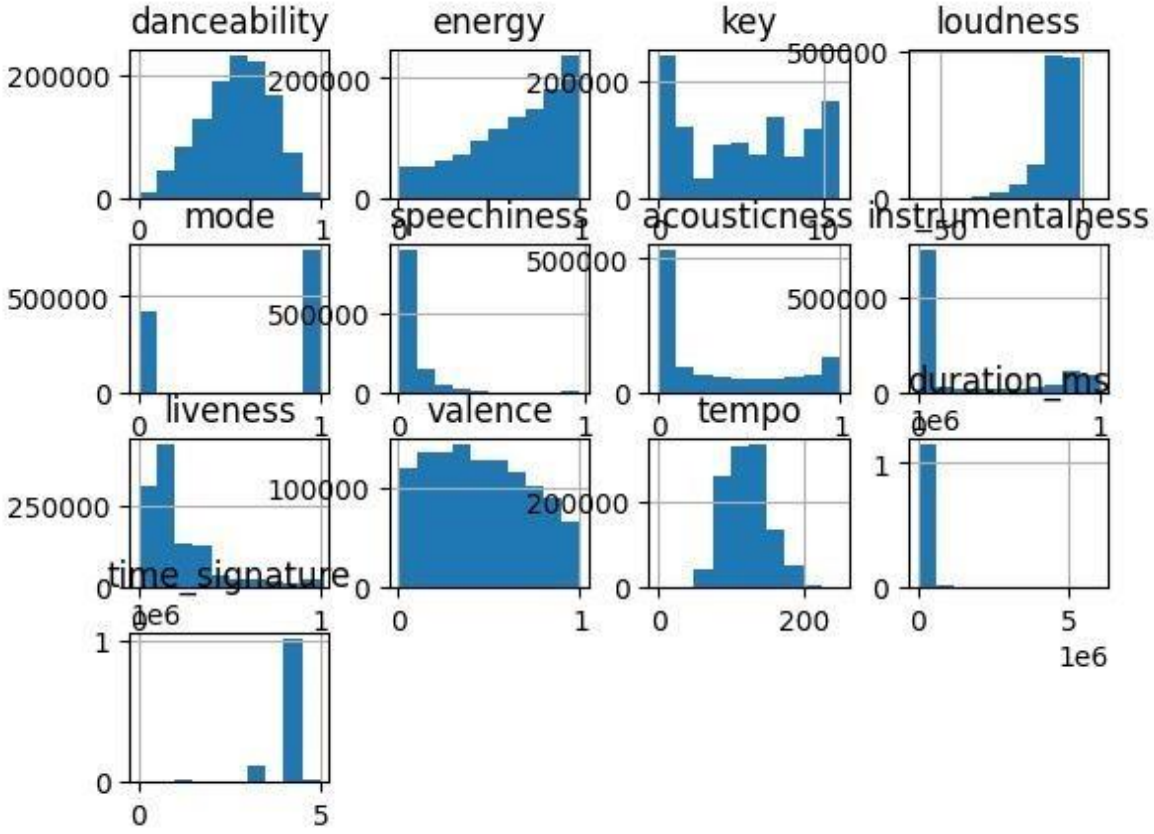
Removing Duplicates:

It seems that we have a lot of repeated rows too. We're going to drop them as well. After all these data cleanings, the columns are numerical but 'id' which is string. We keep this column because when we want to recommend a song to a user, we're going to use it, But when clustering we'll temporarily drop it, so that it doesn't have an effect on the result.

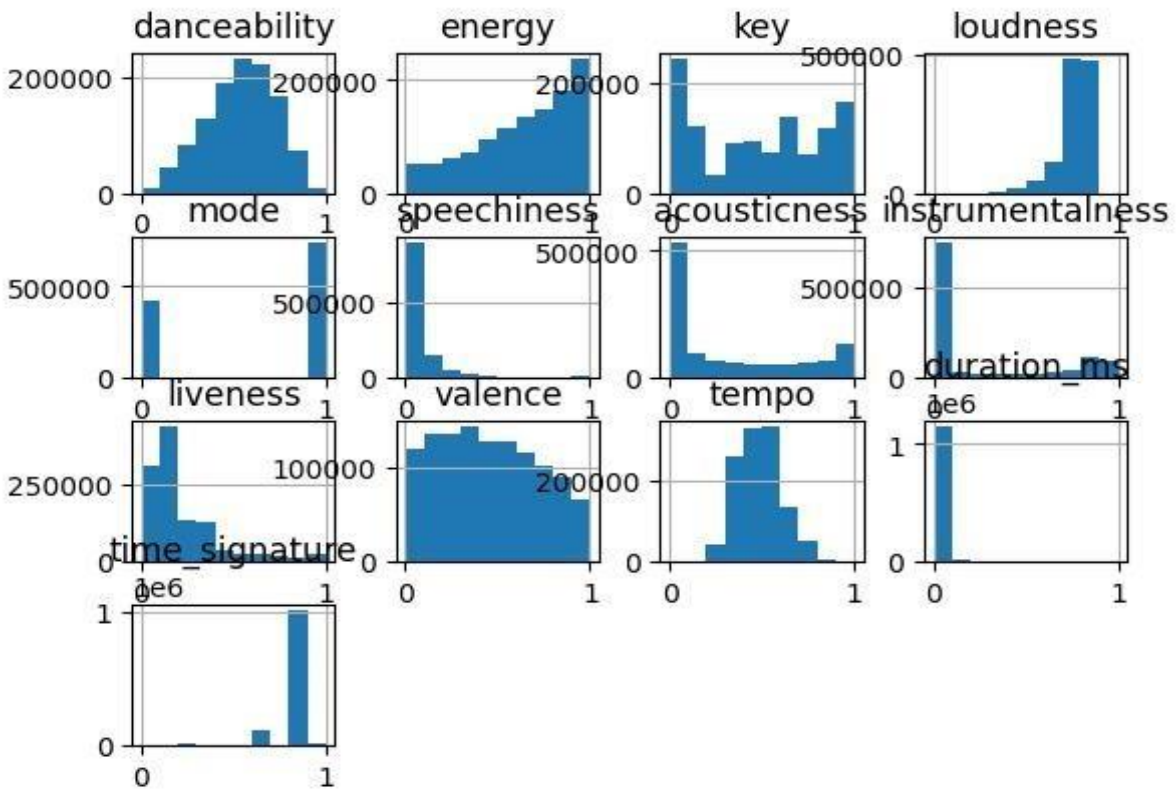
```python
def music_recommender():

    raw_data = pd.read_csv('spotify_data.csv')
    #print(raw_data.shape)
    #raw_data = raw_data.iloc[:1000,:]
    #print(raw_data)

    pd.set_option('display.max_columns', None)
    raw_data.info()

    # data cleaning ----------------------------------------------------------------------------
    nulls = raw_data.isnull().sum()
    #print(nulls)

    training_data = raw_data.drop(['artist_name','popularity','year','genre','Unnamed: 0'], axis=1, inplace=False)

    print(training_data.shape)
    training_data = training_data[training_data.key != -1]
    print("after dropping some rows:\n", training_data.shape)
    print(training_data.head())

    print(training_data.shape)
    print(training_data.duplicated().any())
    training_data.drop_duplicates(inplace=True)
    print(training_data.shape)
```

2) DATA TRANSFORMATION

Now Let's take a look at histogram plots of our data:



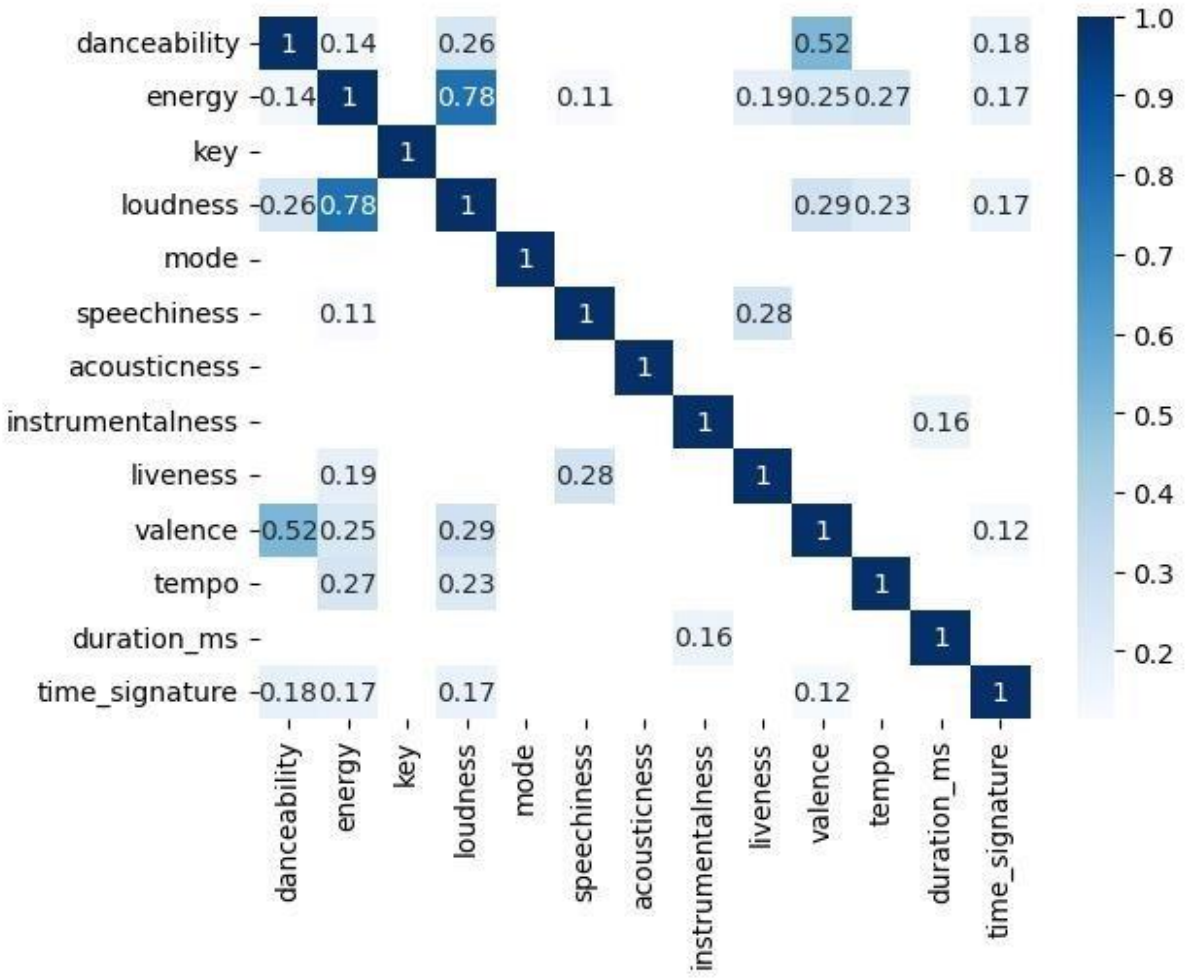
If we want to use K-means, the distribution should be symmetric. As we can observe from the plot above, most of the features have a skewed distribution. So, we should normalize them.
As many of our variables have left skewness a method like logarithm cannot help.
It's better now, not completely normal but it's better than before.
What about scales of the variables?
To use K-means algorithm, all the variables should have the same mean and variance (be on the same scale). We want to transfer all of them to the **range of 0.0 to 1.0**. Which means we're going to normalize them.
An Important point that we should be careful about, is that: We should fit a scaler on the training data and then use this scaler to scale both training data and test data. they shouldn't have separate scalers. Because separate scalers will cause data leak.
After normalizing the result is as follow:




```
# this global scalar will be fitted on training data and will be used for both training and test data
global_scalar = MinMaxScaler()
id_column = training_data['track_id']
track_name_column = training_data['track_name']
training_data.drop(['track_id','track_name'], axis=1, inplace=True)
global_scalar.fit(training_data)
training_data = pd.DataFrame(global_scalar.transform(training_data),
                             index=training_data.index,
                             columns=training_data.columns)


training_data.hist()
plt.show()

training_data.info()

corr = training_data.corr()
sns.heatmap(corr[corr > 0.1], cmap="Blues", annot=True)
plt.show()

## End of data visualisation...
```

3) DATA REDUCTION

Before moving on to clustering let's just make sure about another thing: For a better result, we should find highly correlated variables, and if there are any, we should drop them.



None of the correlations are not very noticeable. So we keep all ofthem. Now, Our data is ready to be clustered…

CLUSTERING:

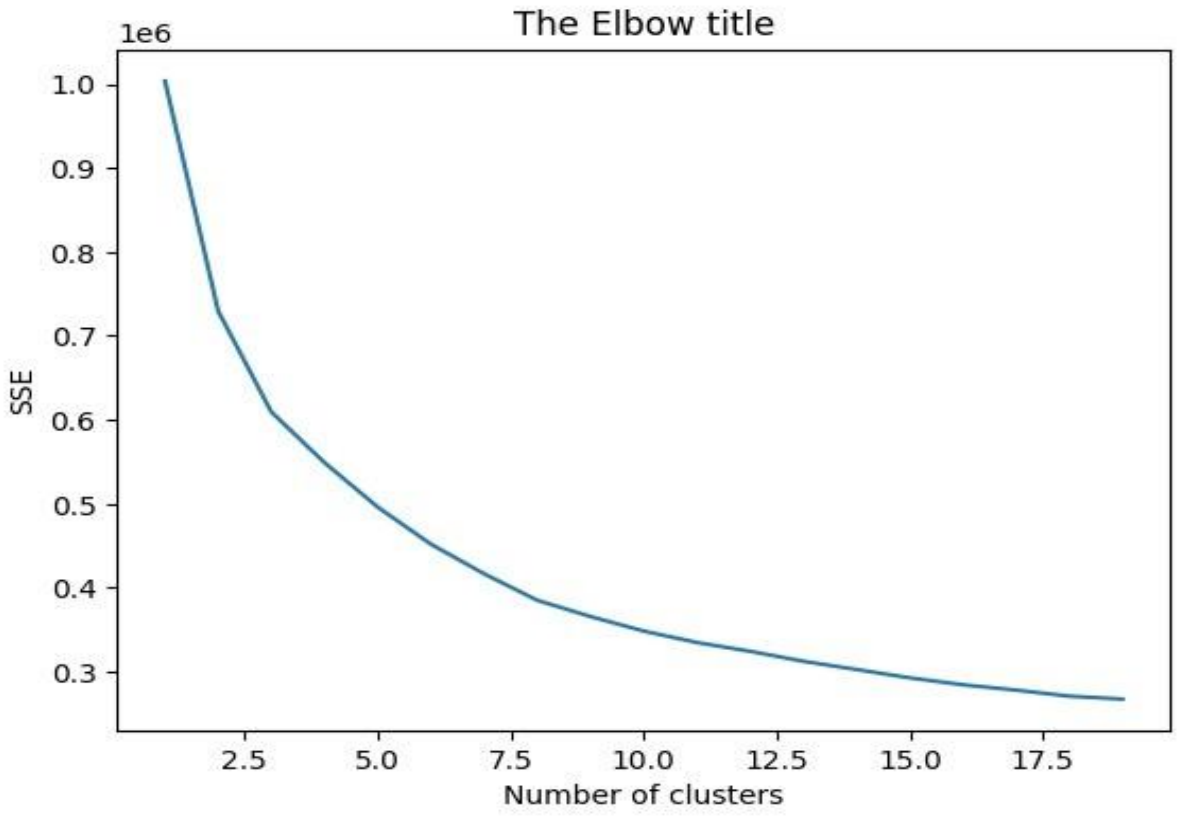
We're going to use K-means algorithm which is an unsupervised one.
An important point about K-means, is that before running, it should be told about the value of k, which is the number of clusters that we want to have.
But how do we select a meaningful number for the clusters?
To answer this question we use the elbow method.
It can be concluded that 10 clusters can work well.
After clustering now I can handle the input/output part of the assignment...

```
# run block of code and catch warnings
with warnings.catch_warnings():
    # ignore all caught warnings
    warnings.filterwarnings("ignore")
    # execute code that will generate warnings
    # clustering(Using K means clustering)----------------------------------------------
    training_data['track_id'] = id_column
    training_data['track_name'] = track_name_column
    wcss = []
    for i in range(1, 20):
        kmeans = KMeans(i)
        kmeans.fit(training_data.drop(['track_id','track_name'], axis=1, inplace=False))
        wcss_iter = kmeans.inertia_
        wcss.append(wcss_iter)

    number_clusters = range(1, 20)
    plt.plot(number_clusters, wcss)
    plt.title('The Elbow title')
    plt.xlabel('Number of clusters')
    plt.ylabel('SSE')
    plt.show()
```

NORMALIZING THE INPUT:

The input file contains preferences of the user. Before anything we have to normalize them, they should be normalized with the same scaler that we made based on our training data.

```
kmeans = KMeans(n_clusters=10,n_init = 10)
training_data_clustered = kmeans.fit(training_data.drop(['track_id','track_name'], axis=1, inplace=False))
training_data["cluster"] = training_data_clustered.labels_
centroids = training_data_clustered.cluster_centers_
print(training_data.head())

#static input
userPreferences = pd.read_csv("input_tracks.csv")

# making output.....................................................................................
userPreferences.drop(userPreferences.columns.difference(["danceability", "energy", "key", "loudness", "mode",
                                                         "speechiness", "acousticness", "instrumentalness",
                                                         "liveness", "valence", "tempo", "duration_ms",
                                                         "time_signature"]), axis = 1, inplace=True)

# input normalizing
userPreferences = pd.DataFrame(global_scalar.transform(userPreferences),
                               index=userPreferences.index,
                               columns=userPreferences.columns)

fields = ["track_id", "track_name", "cluster"]
```



3.1 METHODOLOGY OF PROJECT:

The selected methodology used for developing this work is the Unsupervised Machine Learning. Its characteristics fit very well to the needs identified after the planning of project execution. These characteristics are:

- Clustering
- Based on goals and use cases
- Using Machine Learning
- Periodic testing system
- Track Changes

The nature of the problem makes necessary several iterations over a changing pattern. This model will be continuously improved, as we proceed through the knowledge of the data with which it works, more reliable will be its behaviour.

4. RESULT AND ANALYSIS:

Output contains a csv file which are as follow:

single_playlist.csv

A complete recommender system about the state-of-the-art of recommender systems, specially focusing on music recommender systems has been

performed. Aimed to settle the project analysis and design strategies, as well as define correctly the work outlines.

	track_id	track_name	cluster
1	lxw9a5A2fDvcBTBVy0tEJ	It's Gonna Be Me	1
2	ghAx3U7gjFqKnVXhCKfO	ypnotized – Club Dub Mix	7
3	OpYCnX2rBNGTWagexTI	Giant Mantis vs. TURT nip	2
4	6gzVwzZDcbzUELRP8Vl	The Idol	8
5	q4Jd66Pn6xzE5CWH5fP	Será Que Foi Saudade	8
6	AhWtQnpoL5Vl9VQPl9fU	Senorita	1
7	BkstDXtKf2eKXmrp2yJw	Cao	8
8	lzjL2GUQxTLjHVLxGMxp	The glass globes	3
9	1zmUMkJkne2fG5hawGt	ke Up Baby - Remastered	7
10	5gXorSMH6WlVpaPdlnU	Cut the Strings - 7" Remix	0
11	d57RvhSrREibo2sXE7QP	Protection	8
12	2G9o9rLBBH7pQnTSH2w	d Cities of the Heart - Live	2
13	l3nwlynB8jaZRSVu57oFk	Poor Born	2
14	l78CAyXn0HqGF5TOGb3	Paixão de um Homem	1
15	5guIneSyc5l30ibdysyy6O	Prejudice	8
16	4pevalltBeg3qpiXjl4enB	Kiss Me on the Phone	8
17	aYBrU6ME2jgZXXwlaHLI	Exacto	7
18	Se6isMePzQ9YFTocvl3bk	Bali	8

5. CONCLUSION & FUTURE SCOPE:

The system overview itself is an innovative approach to music browsing and discovering including efficient recommendation features. The system is completely implemented using open-source tools and modules. It is convenient to remark that the system exploits some resources freely provided by commercial music systems required to achieve its functionality.

It was observed that the development of a recommendation system with commercial features requires an extensive relational database to store the music previously catalogued . Creating a good relational database of music, making relations between artists, albums, musical genres, and époques, could greatly expand the capabilities of a recommender algorithm to help users discover new music.

REFERENCES:

- https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.hist.html
- https://realpython.com/k-means-clustering-python/
- https://towardsai.net/p/data-science/how-when-and-whyshouldyounormalize-standardize-rescale-your-data-3f083def38ff
- https://medium.com/@evgen.ryzhkov/5-stages-of-data-preprocessingfor-k-means-clustering-b755426f9932
- https://developer.spotify.com/documentation/webapi/reference/#/operations/get-audio-features
- https://silo.tips/download/music-recommender-system
- https://github.com/AnitaSoroush/MusicRecommendationSystem/blob/main/musicRecommender.py