**Release Notes**

**“Music Recommendation Systems”**

**By**

**Team: Vornoi Practitioners**

| **Sl.No** | **Team Member** | **Email ID** |
| --- | --- | --- |
| 1 | Mrinal Bhardwaj (Captain) | mrinal.nbd@gmail.com |
| 2 | Ashutosh Kumar Bharti | ashubharti65@gmail.com |
| 3 | Ankitha R Desai | ankitha.desai0816@gmail.com |
| 4 | Muskan A | muskan.af18@gmail.com |
| 5 | Rutvik Jaiswal | rutvikjaiswal195@gmail.com |

# **1. Introduction:**

Music recommender system is a system that learns from the user's past listening history and recommends songs that they would probably like to hear in the future. People sometimes feel it is difficult to choose from millions of songs. Moreover, music service providers need an efficient way to manage songs and help their customers to discover music by giving quality recommendations. Thus, there is a strong need for a good recommendation system.

The aim of this project is to design and develop a “Music Recommendation System” that will detect preferences and produce a playlist based on them. We are using the Spotify dataset provided by Kaggle and building this recommendation system using Content-based recommendation and Nearest Neighbor algorithm.

# **2. Compatible Products:**

This project was tested on the Windows OS and Jupyter platform using the unit test module. A song index is chosen randomly or as per user’s choice, and the Recommender system will check for similar song indices based on features such as loudness, duration\_ms, tempo, and so on, and ‘n’ number of similar songs are recommended.

# **3. Upgrades:**

When implementing recommender systems, researchers generally adopt two methods: Content-Based Filtering (CBF) and Collaborative Filtering (CF).

# **4. Content-based filtering:**

Content-based filtering uses the features of each item to find the similarities items. By assigning a score to how similar each item is, we can recommend an item based on how similar it is to all other items in the dataset.

In the context of Spotify playlists, we use the features (loudness, tempo, etc.) of each song in a playlist to find the average score of the whole playlist. Then, we recommend a song that has a score similar to the playlist but is not in the playlist.

# **5. New Features:**

* The attributes of a song are used in this recommender system.
* Tempo, energy, liveliness, loudness, danceability, and many other important characteristics are utilized.
* Recommender system takes into account all the relevant features of a song that practically describes all the qualities of a song.
* Correct recommendation is rolled out based on all the features of the song, user listens to.