Microsoft Intern Engage: Algorithm

~Sunaina Rustagi

- **My Introduction:** I am Sunaina Rustagi, currently pursuing B.Tech in Information Technology from Maharaja Surajmal Institute of Technology and also pursuing B.Sc. in Programming and Data Science from Indian Institute of Technology, Madras.

  I have been awarded with the Google Generation Scholarship for the year 2022 and Microsoft Learn Student Ambassador as well.

  I generally work with 2 languages, Java and Python and is currently working on C++ as well. I am also a graphic designer proficient in handling the Adobe Creative Suit. However, I am still exploring the field of Machine Learning and Web Development.

- **Project:** Algorithm based project
  - Trying to implement and understand the algorithms used for recommendation by audio streaming apps like spotify.
  - UI building for the audio streaming app (clonning Spotify)

- **Spotify Algo:** Bandits for Recommendations as Treatments (BaRT)
  - Responsible for 2 things:
    - Explore - Explores the interests of new users through studying the data received from other people and the recommend.
    - Exploit - Recommends on studying the user history, activities, social media or even location of the user.
  - BaRT registers the songs liked by the user if the users listen it upto 30 secs at least, otherwise it is marks as unliked by the algorithm.

- **Structure:**
  - Public

- Music Recommendation System
  - Model.ipynb
  - Recommender.py
  - Song_data.csv
  - triplets_data.csv
- Src:
  - Components:
    - Body
    - CurrentTrack
    - Footer
    - Login
    - Navbar
    - PlayerControls
    - Playlist
    - Sidebar
    - Spotify
    - Volume
  - Utils:
    - Constants
    - Reducer
    - StateProvider
  - App.jsx
  - Index.css
  - index.js

- **Recommendation Engine:** Collaborative filtering engine
  - Recommends on the basis of users activity, past history, preserenses etc.
    - Model form :
      - Contains 2 files: Triplet_file - user_id, song_is, listen_time & Metadata_file - song_id, title, release, year, and artist_name
  - Language used: Python
  - Input : User & Item
  - Output: Predictions

    - Recommender.py: cooccurence_matrix will calculate the values which will calculate the value of few parameters like listen count, etc.
    - Co-occurence matrix formation.
    - Disadvantage: only limited number of songs can be analyzied as they are being studied on the basis of dataset obtained from websites.

- **Features still working on:**
    - Speech recognition
    - Inbuilt Radio FMs
    - Making playsits on analysing user interests