Project Checkpoint 2

Team Python

Chirag Bhatt 2018MT10750

Parth Singhal 2018CH10231

Rohan Sharma 2018TT10909

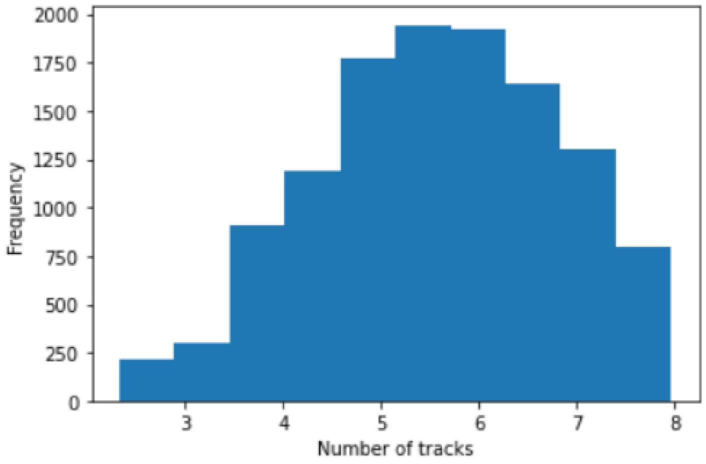
For storing the data, we have created 3 pandas dataframe, playlist_lookup_table, track_lookup_table and master table. Each row in playlist and track lookup table is characterized by unique playlist and track respectively and each row in master_table is characterized by track and playlist and column represents track uri, album uri, artist uri, duration, etc. This data frames contains all the information of the dataset.

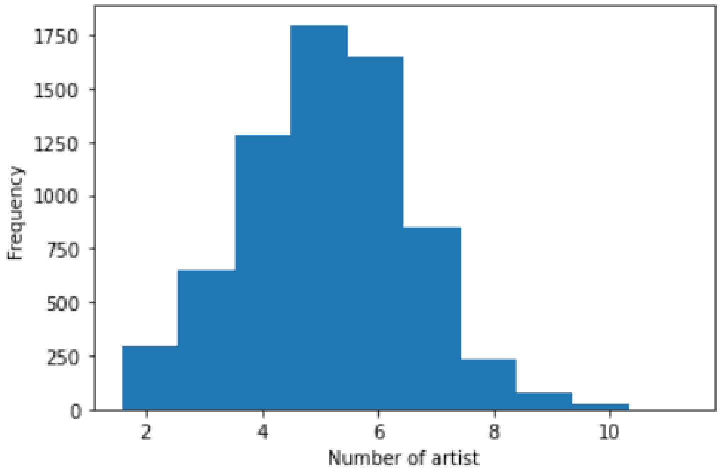
Our initial approach consists of track based and playlist based collaborative filtering. We have created Playlist Track matrix where each row represents unique playlist and each column represents a track and value of a cell at index i,j is 1 if j^{th} track is contained in i^{th} playlist and 0 otherwise. This approach is based on the paper :

https://www.researchgate.net/publication/328086857_Artist-driven_layering_and_user%27s_behaviour_impact_on_recommendations_in_a_playlist_continuation_scenario

We have then applied BM25 normalization on the matrix and used KNN algorithm to find similarity between tracks and between playlists. Currently we are considering cosine and Tversky similarity models

For the analysis of the dataset, we plotted number of frequencies of log base 2 of unique tracks and unique artists per playlist.





From the frequency plot we can infer that computing artists similarity can be alternative way of computing similarity between playlists.

For fast implementation of normalization, KNN and other similarity algorithms we are using similaripy library (https://pypi.org/project/similaripy/) (the creator of this library also got inspired from RecSys Challenge) and SciPy. Sparse library in python.

We are running initial scripts in our laptop on a smaller subset of dataset, will use hpc server to run the script on the whole dataset.

Currently we aim to run the script on whole dataset on hpc server for different values of neighbours in KNN algorithm and evaluate on the test set. Since the matrix dimensions are very high and matrix is very sparse we also aim to apply PCA algorithm to bring down the dimension.