MUSIC RECOMMENDATION SYSTEM USING DEEPING LEARNING AND COLLABORATIVE FILTERING TECHNIQUES

****

NAME – Gaurav Garg

STUDENT ID - 980452

Under the supervision of

AKSHITA BHANDARI

Research Proposal

Liverpool John Moores University – Master's in Data Science

September 2021

Abstract

Managing and looking for songs has become increasingly important as digital music formats have grown in popularity. Though successful music information retrieval (MIR) algorithms have been developed in the previous ten years, music recommender systems are still in their infancy. As a result, this study examines a broad framework as well as cutting-edge techniques to music recommendation. With the growth of the internet in recent decades, it has become the primary source for retrieving multimedia material such as video, literature, and music, among other things. People regard music to be a significant part of their lives, and they listen to it on a regular basis. Participants listened to music more than any of the other activities, according to previous study. Additionally, the music recommender will assist users in filtering and discovering songs based on their preferences. A smart music recommendation system should be able to detect preferences automatically and produce playlists based on them. Meanwhile, the advent of recommender systems gives the music industry a fantastic opportunity to gather users who are interested in music. More importantly, it presents us with new challenges in terms of better understanding and modelling customers' musical preferences. We explore the current state of the art in solving these problems and its limits. We go over potential future directions and visions for the field's future development. The first stage of our model is tailored for quick retrieval, while the second stage re-ranks recovered candidates to maximise accuracy at the top of the suggested list.

Table of Contents

1. Background

Recommender systems have become a necessary component of many e-commerce and Internet-based organisations such as Google, YouTube, Facebook, Netflix, LinkedIn, Amazon, and others have used recommender systems as part of their business strategies since the Web's inception and spectacular expansion over the last decade. firms in various domains like songs, movies, articles, shopping, etc. These systems are based on user sentiments about a product or service with the goal of promoting them to other users with similar interest according to their past behaviours and tastes. YouTube, for example, offers recommendations for a user based on their viewing patterns and searches for like users. The task of automatic playlist continuation (APC) is for music recommendation systems to find tracks that fit inside a specified playlist. A big part of the APC mission is to appropriately deduce the playlist's intended purpose. This is difficult not only because of the wide range of intended purposes (assuming they exist at all), but also because of the wide range of underlying attributes or characteristics that may be required to infer those purposes. A recommender system is an essential and appealing domain in information filtering systems that seeks to forecast active users' choice or rating of a particular item. A recommender system is a sophisticated engine that uses a variety of ways to suggest the most relevant products for a group of users based on their prior actions and preferences.

Therefore, relevant and appropriate recommendations are extremely important for these businesses to attract more users into using their platform for longer period of time. Even before the digital revolution, music services relied on a variety of entry points into the music catalogue: genres, decades, hit choices, new releases/trending, what's curators/influencers, playlists by context (moods/activities), and provided means for sharing content and playlists. In contrast to other creative content, a song is a three-minute experience, and the question of what to listen to next keeps coming up. As a result, the album's historic format, which ensures a minimum permissible duration, as well as its aesthetic objective, is preserved. Listeners now have devices, apps, and algorithms thanks to the digital revolution, which allow them to better capture those listening situation opportunities and adapt to each context: rich user interfaces on PCs, simplified user interfaces on mobile phones and in cars, voice control in the car, and smart speakers. Listeners can experience rich navigation even when interaction with screens is limited, to the point of inviting designers to create a zero interface where no interaction is possible. Higher granularity (the ability to provide numerous types of playlists, similar artists and

songs), higher frequency of selection updates, a much deeper dive into the collection, and personalisation are all advantages of digital. Various platforms like Rapcaviar, Discover Weekly provides listeners with customised playlist updated every week. Spotify on the other hand, creates multiple playlists based on users most played artist, popular songs in the area and lots of other playlist categorised by genres.

The collaborative filtering algorithm has been determined to function well based on users' listening behaviour and past ratings. When combined with the use of a content-based model, the user can get a list of songs that are comparable based on low-level acoustic variables like rhythm and pitch, as well as high-level features like genre and instrument. Last.fm2, Allmusic3, Pandora4, and Shazam5 are 4 music discovery services that have successfully used these two methodologies. Meanwhile, these websites give a one-of-a-kind platform for retrieving rich and usable data for user studies. There is a lot of possibility for future expansion thanks to new discoveries in psychology, signal processing, machine learning, and musicology. As a result, this study examines a broad music recommender framework, including user profiling, item modelling, and item-user profile matching, as well as a number of state-of-the-art methodologies. We conclude and suggest a new model based on user motivation at the end of this study.

2. Problem Statement

In the last decade, recommender systems have been in the focus for many studies to develop new methods and improve their accuracy. Two types of recommender systems are: collaborative filtering and content-based filtering methods. Most recommender systems are based on collaborative filtering as it is feasible and easy to implement. There are two categories of collaborative filtering: memory-based (Li et al., 2016) and model-based strategies (Paradarami et al., 2017). Model-based methods use ratings to train a model and predict ratings on unrated items by users, whereas memory-based methods use similarities between items and users to predict. Researchers (Zarzour et al., 2018) introduced a model-based music recommendation system which clusters users using k-means clustering and reduce dimensionality with SVD. One of the earliest model-based collaborative filtering methods was proposed 2 decades ago (Chen and George, 1999), which was based on the idea that users (or judges) who tend to give similar ratings can be modeled as having identical rating probabilities.

Content-based methods take into account both the profile of user’s interest and description

of item. Bayesian classifiers, for example, are used to assess the likelihood of a user loving an item based on its content (Pazzani and Billsus, 2007) and another research (Chen et al., 2008) introduced contest-based recommender systems that employs profitability into CF-based systems. A text similarity method was introduced (Robertson and Walker, 2000) that sets a threshold for text similarity in item description and employs adaptive filtering. These systems suffer from many challenges such as misinterpret songs that rarely appear, completing playlist with very few tracks, neglecting sequence of tracks in a playlist. To address these challenges, there are hybrid methods proposed (Nicholas, 1999) and (Ungar and Foster, 1998) which combines collaborative features and content together. Recently, researchers (Xiong et al., 2018) proposed a deep learning-based hybrid method by combining collaborative filtering and textual content. Furthermore, explainable recommendation has also been a field of research interest (Zarzour et al., 2020) where trustworthiness score which depends on users' reactions after watching recommended videos is combined with collaborative filtering.

These collaborative filtering and content-based methods are also applied in music recommender systems. Some authors (Chen et al., 2012) and (Mcfee and Lanckriet, 2011) have developed ways for modelling the transitions between songs in playlists using Markov chains. Although, recently researchers have found that sequence of songs hardly matters to listeners while the song-to-song transitions (Kamehkhosh et al., 2018) and ensemble of songs in playlist (Chen et al., 2018) seems to matter. An online competition (ACM RecSys Challenge 2018) was held for the task of Automatic playlist continuation (APC), which involves recommending songs that are likely to be added to an existing playlist. For example, the recommended songs for a list of new year songs should be other new year songs. As part of the challenge and to promote this type of research at scale, Spotify publicly released the Million Playlist Dataset (MPD) in 2018. The data is now publicly available at (AIcrowd | Spotify Million Playlist Dataset Challenge | Dataset_files, 2021) . Every playlist has a title, last modified time, description, total duration, list of tracks with details like track IDs, artist name, track uri etc. for every track. Playlists are filtered to remove tracks with offensive content.

The methods used by participating team in the challenge were mostly based on collaborative filtering combined with dimensionality reduction or CNN models. For instance, (Yang et al., 2018) proposed a 2-phase collaborative filtering model where in

first phase autoencoders is used to encode and decode playlist features and CNN is used for training just playlist titles. Another team (Teinemaa et al., n.d.) used a three-stage model, where in first stage 4 collaborative filtering models are built on 4 context track-track, word-track, album-track and artist-track. First stage is then followed by taking weighted sum of all 4 models and then completing playlist for a given number of tracks based on same album and popular tracks. The winning team (Volkovs et al., 2018) also applied a two-stage model with additional node for cold start problem. Cold start problem occurs when there are only 2 playlist features available: name & length of the playlist. The first stage employs a WRMF collaborative filtering method which is one of most known methods for implicit/binary collaborative filtering (Hu et al., 2008), since WRMF ignores order of songs in the playlist it is clubbed with a temporal CNN and Neighbour-based CF models (song-song, playlist-playlist). The second stage contains an xgboost classifier used to re-rank prediction from 1st stage and additional playlist features like average song/artist/album, song features like artist/album, pairwise playlist song features like similarity between target song and songs in the playlist and furthermore, creative features like danceability, energy, mode etc. taken from spotify audio api.

With the recent improvement in deep learning techniques, many types of powerful news-based (Okura et al., 2017), apps-based (Cheng et al., 2016) and video-based (Covington and Adams, 2016) recommendation systems are based on deep learning. To address the issue of data sparsity and cold-start recently many deep learning techniques are proposed. Another strong technique, Collaborative Deep Learning (CDL), is given, which uses Bayesian formulation to do deep representation learning for the ratings matrix. CDL allows for two-way communication between content information and ratings matrix feedback.

However, there is no study that demonstrates how to embed people and products together. with taking order of songs into account. Also, there are no studies that combine recent deep learning models with cold start problem to build a versatile collaborative filtering model. For example, by clubbing a neighbour-based collaborative filtering model with deep learning and employing a CNN trained on playlist titles to solve for cold-start problem can produce a more effective result. There are also no studies that take number of times a song appears in playlist into consideration.

3. Research Questions

The aim of this research is to develop a recommender system that performs Automatic Playlist Continuation (APC). Since playlist continuation is required to be done in production environment, the recommender system should be versatile enough to work on various limitations such as a smaller number of tracks in the playlist, only playlist title etc. and hence will be tested on such a test set.

Based on the literature review, below research questions have been formulated:

- What are the most and least important playlist and track features during APC?
- What are the best performing models to recommend songs for playlist?
- What solves cold-start problem for playlist with very less features and tracks?
- What can be laid as a foundation for future work in music recommendation systems?

4. Aim and Objectives

The research aims to understand factors influential in music recommendation systems with the task of Automatic Playlist Continuation. The research targets development of an efficient music recommender system to be used by online music streaming websites and eventually help listeners find more tracks they may like.

Below are the research objectives:

- To perform literature review of researches in recommendation systems.
- To perform exploratory data analysis on track and playlist features, find correlation between them and perform feature generation.
- To employ deep learning, clustering and collaborative filtering models for Automatic playlist completion.
- To compare results and identify most efficient models in creating a music recommender system.
- To consolidate results from finalized models.

5. Significance of Study

The study is directed to develop a music recommender system that can be used in production by music streaming platforms to give useful and relevant recommendations to its users. This study will focus on developing a solution that combines recent deep learning models with cold start problem to build a versatile collaborative filtering model which as not been explored yet. The research would also be able to answer which features contribute most and least to music recommendation, this can also greatly help streaming platforms to better understand profile of the user. This study introduces a new collaborative filtering

recommendation algorithm based on dimensionality reduction and clustering techniques with the goal of increasing the program's recommendation performance. Music recommendation is relatively a new area of research and is gaining researchers interest over the years since the business of online music streaming platforms entirely depend on their ability to keep user attracted to the platform for as long as possible and at the heart of it sits the recommendation system.

6. Scope of the study

Recommendation is all about broadening a listener's musical horizons beyond what they already know and enjoy. It gives listeners more navigation speed once they've exhausted all of their song/artist search capabilities. Scope of this study is to make use of the existing deep learning, collaborative filtering and clustering algorithms and to develop new such algorithms that best suits the recommender system. The evaluation of model would be carried out on the separate test set which has playlists of varying lengths, the evaluation metrics used can tell us about recall and top k accuracy of the model. The suggested models will be trained and tested to get consolidated table and arrive at a conclusion on which models performs the best and if there is a scope of further development in the models.

7. Research Methodology

7.1 Introduction

Automated Playlist Continuation (APC) is getting more popular as people listen to music online. While there has been substantial progress in recent years, the majority of the methodologies presented are tested on exclusive datasets, making open collaboration and convention impossible. By undertaking standardised assessments of playlist continuance trends, the 2018 ACM RecSys Challenge intends to close this gap. At the centre of this ambition is the Million Playlist Dataset (MPD), which was launched with the support of Spotify. MPD is the largest publicly accessible dataset of its sort, with over 2.2 million songs and 1 million playlists. The study's goal is to create a playlist continuation version that can suggest suitable following songs for each playlist. For the evaluation, a different set of 10K test playlists is used, with a few songs withheld. The duration of the test playlists ranges from zero (cold start) to a hundred tracks. This mimics a real-world scenario in which the APC model must perform correctly at all degrees of playlist completion. This study's methodology is mostly based on a two-level structure. The first stage seeks to build a big selection of tracks with high recall value for each playlist, and the second stage rates

only the tracks generated from first set and select top k. Below are the steps involved for this research:

Step1: Data Cleaning & Understanding

Data understanding and cleaning is important part of a machine learning research. In most research works, data is collected from publicly available sources, which in this case is the MPD published by Spotify and available publicly as an open contest at AiCrowd. Data cleaning steps will be carried out on MPD like removing data sparsity, filling or removing missing values, feature reduction, correlation analysis between features etc.

Step 2: Exploratory Data Analysis

Exploratory Data Analysis (EDA) is carried out to gain hidden insights from the dataset. Univariate, bivariate and multivariate analysis will help in understanding relation between multiple features w.r.t target feature. For visual examination, various charts will be used such as bar chart, heat map, scatter plot, etc. These charts will help to understand trends between different features of the dataset.

Step 3: Pre-processing and Transformation

Pre-processing is required to make data ready for modelling. In the pre-processing steps categorical features will be transformed using one-hot encoding. Features data-type will be restored to the one supported by machine learning model. In addition, outliers will be found and removed from every column in the dataset.

Step 4: Models:

Deep learning, Clustering and Collaborative Filtering are some models to be trained for this research. Every model will be tuned using grid search cross validation. The approach followed will be to first build a baseline model and keep iterating the parameters until we achieve the best results. Different models like xgboost, random forest etc. will also be explored to make sure we use best fit model.

Step 5: Evaluation:

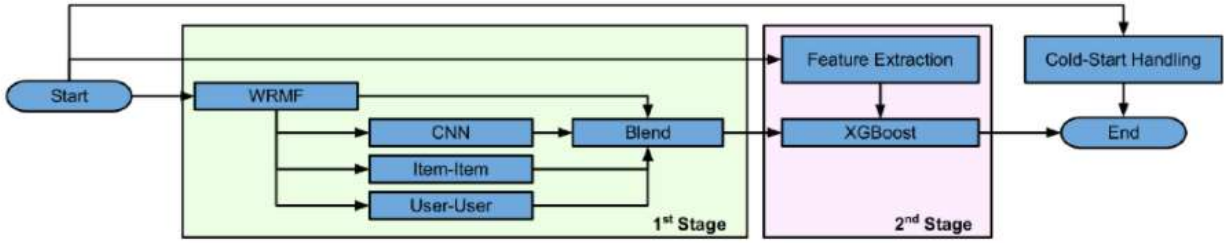
Evaluation is done on test dataset containing 10k playlists of varying lengths. Three different metrics for evaluation are used to test different features of the solution: Recommended songs clicks, Normalized discounted cumulative gain (NDCG) and R-precision. More detail about each of the evaluation metric and what they test is present in

"Evaluation Metrics" below.

Step 6: Results & Conclusion

Consolidated results from deep learning, clustering, xgboost and collaborative filtering models will be generated. These results will give insights about which set of models performs the best and will be compared with actual test set results available to evaluate at AiCrowd (AIcrowd | Spotify Million Playlist Dataset Challenge | Dataset_files, 2021).

Figure 1: Process Flow Diagram



7.2 Dataset description

This research will consume the Million Playlist Dataset (MPD) published by Spotify and available publicly as an open contest at AiCrowd. This data is appropriate for the research because it contains playlist of varying lengths and cold-start problem as well. Spotify being the primary choice for online music streaming for many users all over the world, it has ample number of tracks for all popular languages. This collection of 1 million playlists, drawn from Spotify's over 4 billion public playlists, has over 2 million distinct tracks by approximately 300,000 artists, making it the world's largest publicly available dataset of tracks and playlists. Between January 2010 and November 2017, public playlists generated by Spotify users in the United States were included in the dataset. Every playlist has a title, last modified time, description, total duration, list of tracks with details like track IDs, artist name, track uri etc. for every track. Playlists are filtered to remove tracks with offensive content. Below table has description for features of playlist and track.

Table 1: Playlist attributes

Attribute	Description	Type
pid	The MPD ID of this playlist is its playlist id. This is a number that ranges from 0 to 999,999.	Integer

name	This is the title of the playlist.	String
description	This is the playlist's description. Note: Because for playlist created by users, description is Spotify's new feature, the majority having missing description.	String
modified_at	When was the last time the playlist was updated? Times are rounded to nearest hour on date the playlist was last updated, which is midnight GMT.	Timestamp
num_artists	The number of distinct artists represented in the playlist's songs.	Integer
num_albums	The total number of albums that each track in the playlist has.	Integer
num_tracks	The playlist's total amount of tracks.	Integer
num_followers	The number of people who followed this playlist when the MPD was generated.	Integer
num_edits	The total number of times playlist was edited. Tracks updated into playlist within a two-hour time frame are assumed to be part of a single edit.	Integer
duration_ms	In milliseconds, the duration of all the tracks added in playlist.	Numeric
collaborative	Whether the playlist is collaborative or not. A collaborative playlist can have tunes contributed by multiple people.	Boolean

Table 2: Track attributes

Attribute	Description	Type
pid	The MPD ID of playlist the track belongs to. This is a number that ranges from 0 to 999,999.	Integer
track_name	The title of the song.	String
track_uri	The track's URI on Spotify.	String
album_name	The album's name is the title of the track.	String
album_uri	The album's URI on Spotify.	String

artist_name	The major artist on the track's name	String
artist_uri	The primary artist's Spotify URI for the track.	String
duration_ms	The track's duration in milliseconds	Numeric
pos	The track's position on the playlist.	Integer

This challenge's test set consists of ten separate playlist groups, each having 1,000 playlists. Each group represents a distinct task in terms of completing a playlist:

i. Recommend songs for a playlist based solely on its title
ii. Recommend songs in a playlist based on the title and first track.
iii. Given a playlist's title and the starting five songs, predict the playlist's tracks.
iv. Recommend songs for a playlist based on the first five songs (without title)
v. Recommend songs for a playlist based on the title and the first 10 songs.
vi. Recommend songs for a playlist based on the first ten songs (without title)
vii. Recommend the first 25 songs for a playlist based on the title.
viii. Recommend songs for a playlist based on its length.
ix. Recommend the first 100 songs for a playlist based on the title.
x. Given a playlist's title and 100 random music, predict the tracks for it.

The purpose of this division is to imitate the playlist continuation model's production environment. These playlists will put the model's ability to handle all stages of playlist construction to the test. We will also create a validation set similar to the test set by sampling playlist from train set.

7.3. Data pre-processing and Transformation

In this step we pre-process the data and make it ready for modelling. First step to pre-process MPD is to remove tracks and artist that appear very rarely, because analysing rare samples does not provide statistically significant patterns. For reducing this data sparsity, we filter out tracks that appear less than 6 times and artists that appear less than 4 times. After pruning, the number of tracks is 26% of original count which is 598,293 and number of artists is 155,942 (53%). We would also validate our model on full data to test the hypothesis that if these rare samples provide any significant improvement in accuracy. For playlist titles, we set maximum length to 25 and include only alpha-numeric as well as few special characters (/<>+-).

7.4. Models

We will focus on modelling deep neural networks that include an input layer, a collection of hidden layers, and an output layer in this research. Deep learning model will be used with scores/utility from neighbour-based collaborative filtering models. For activation ReLu function may be used because of its simplicity in optimization, in addition exploring Sigmoid and tanh functions as well. For solving cold-start problem, we plan to explore 2 approaches: dimensionality reduction with SVD or CNN for playlist titles only. Adam optimizer can be used for the deep learning model and CNN model with learning rate of 0.005, which will be tuned using cross validation. The size of filters for CNN model can be 3,5,6 and 9. For Collaborative filtering, 4 neighbour-based (track-track, word-track, album-track, artist-track) or WRMF model can be used. These set of models are expected to produce best results as they are state-of-the-art and are proven to perform better than older ones. Algorithms will be redefined and updated to best suit the use case and will be compared with original to see if they bring any significant changes in accuracy.

7.5. Evaluation metrics

The R-precision, defined as:

$$R - precision = \frac{|T \cap R|}{|T|}$$

where T is the set of ground truth (holdout) tracks, and R is set of recommended songs. The notation |.| denotes the number of elements in the set.

The Normalized discounted cumulative gain (NDCG), defined as:

$$NDCG = \frac{DCG}{IDCG}$$

where:

$$DCG = rel_1 + \sum_{i=2}^{|R|} \frac{rel_i}{log_2(i+1)}$$

$$IDCG = 1 + \sum_{i=2}^{|G|} \frac{1}{log_2(i+1)}$$

The Recommended Songs CLICKS metric, that mimics a Spotify feature for track recommendation where ten tracks are presented at a certain time to the user as the suggestion to complete the playlist. These metric captures required number of clicks or

refreshes before a relevant track is found, and is defined as:

$$CLICKS = \frac{argmin_i\{R_i \ : \ R_i \ \in G\} - 1}{10}$$

While NDCG and CLICKS were calculated based on the track-level agreement between the holdout tracks and the recommended tracks, R-precision was calculated on the artist level agreement. In other words, it was considered sufficient if the artist of a recommended track matched the artist of a holdout track.

8. Required Resources

Required hardware and software resources for this research are:

Hardware or Environment Requirements:

- Data Storage – ASW S3 or Google Drive
- Version control system – Bitbucket or Github
- A Laptop
- Processor – Intel Core i3 or higher
- RAM – 6 GB or more
- GPU – Nvidia Graphics Card or equivalent
- OS – 64-bit Windows 10/macOS Catalina or equivalent
- Kernel - AWS Sagemaker or Google Colab or equivalent

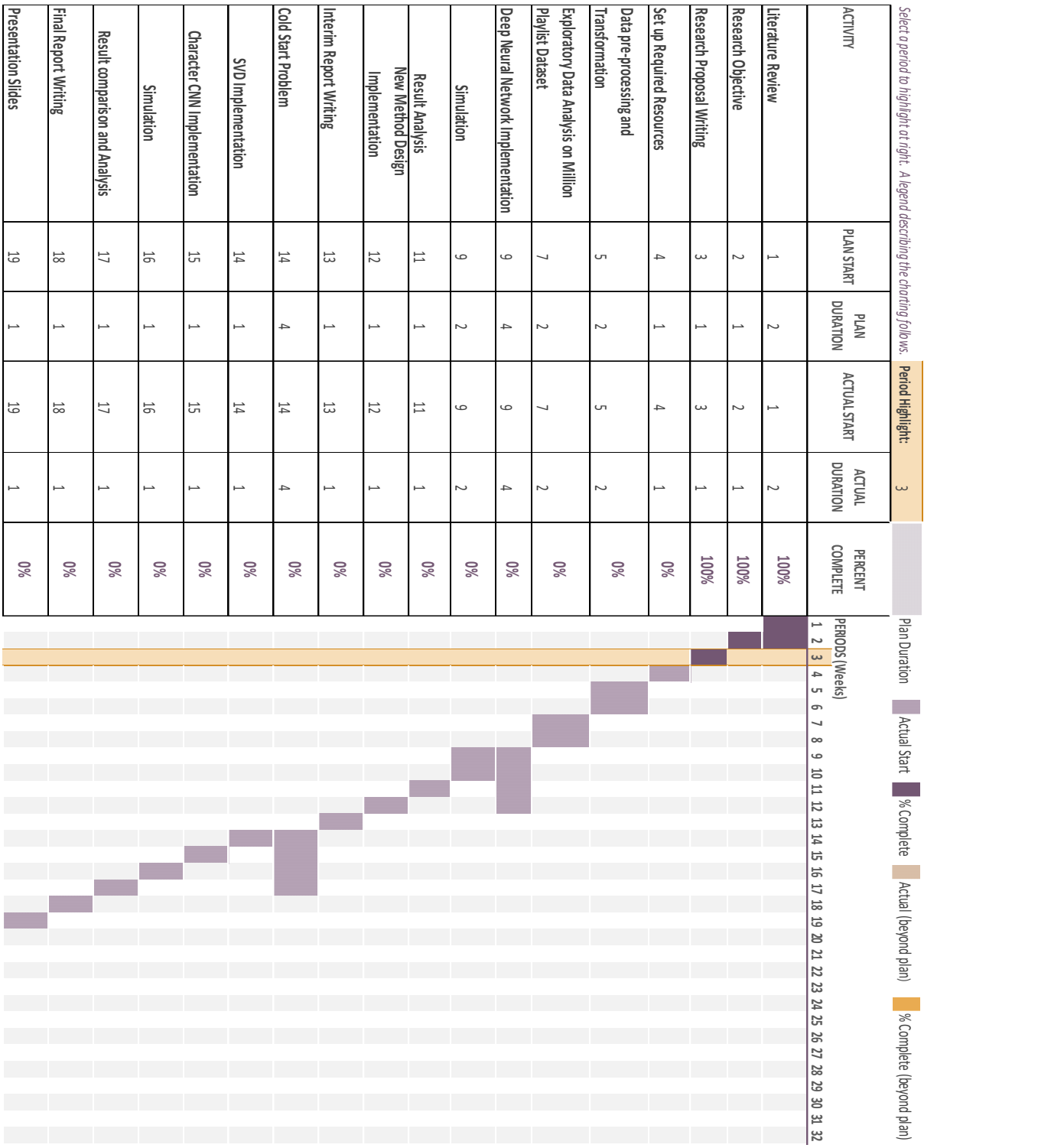
Software Requirements:

- Python
- Python libraries
    - Pandas, Numpy, Seaborn, Matplotlib, Scipy, Statsmodel, Keras, Scikit-learn and Tensorflow
- Anaconda
- Jupyter
- Git

* Note: all latest available versions

9. Research Plan

The project plan for this research is summarised in the Gantt Chart below.



ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Literature Review	1	2	1	2	100%
Research Objective	2	1	2	1	100%
Research Proposal Writing	3	1	3	1	100%
Set up Required Resources	4	1	4	1	0%
Data pre-processing and Transformation	5	2	5	2	0%
Exploratory Data Analysis on Million Playlist Dataset	7	2	7	2	0%
Deep Neural Network Implementation	9	4	9	4	0%
Simulation	9	2	9	2	0%
Result Analysis	11	1	11	1	0%
New Method Design Implementation	12	1	12	1	0%
Interim Report Writing	13	1	13	1	0%
Cold Start Problem	14	4	14	4	0%
SVD Implementation	14	1	14	1	0%
Character CNN Implementation	15	1	15	1	0%
Simulation	16	1	16	1	0%
Result comparison and Analysis	17	1	17	1	0%
Final Report Writing	18	1	18	1	0%
Presentation Slides	19	1	19	1	0%

References

ACM RecSys Challenge 2018. [online] Available at: http://www.recsyschallenge.com/2018/ [Accessed 9 Nov. 2021].

AIcrowd | Spotify Million Playlist Dataset Challenge | Dataset_files. [online] Available at: https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge/dataset_files [Accessed 9 Nov. 2021].

Chen, C.-W., Lamere, P., Schedl, M. and Zamani, H., (2018) RecSys Challenge 2018: Automatic Music Playlist Continuation. [online] Available at: https://doi.org/10.1145/3240323.3240342 [Accessed 9 Nov. 2021].

Chen, L.S., Hsu, F.H., Chen, M.C. and Hsu, Y.C., (2008) Developing recommender systems with the consideration of product profitability for sellers. *Information Sciences*, 1784, pp.1032–1048.

Chen, S., Moore, J.L., Turnbull, D. and Joachims, T., (2012) Playlist prediction via metric embedding. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.714–722.

Chen, Y.-H. and George, E.I., (n.d.) A BAYESIAN MODEL FOR COLLABORATIVE FILTERING.

Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X. and Shah Google Inc, H., (2016) Wide & Deep Learning for Recommender Systems. [online] Available at: https://arxiv.org/abs/1606.07792v1 [Accessed 9 Nov. 2021].

Covington, P. and Adams, J., (n.d.) Deep Neural Networks for YouTube Recommendations. [online] Available at: http://dx.doi.org/10.1145/2959100.2959190 [Accessed 9 Nov. 2021].

Hu, Y., Volinsky, C. and Koren, Y., (2008) Collaborative filtering for implicit feedback datasets. *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp.263–272.

Kamehkhosh, I., Jannach, D. and Bonnin, G., (n.d.) How Automated Recommendations Affect the Playlist Creation Behavior of Users. [online] Available at: http://www.midiaresearch.com/downloads/ [Accessed 9 Nov. 2021].

Li, T., Liu, A. and Huang, C., (2016) A Similarity Scenario-Based Recommendation Model with Small Disturbances for Unknown Items in Social Networks. *IEEE Access*, 4, pp.9251–9272.

Mcfee, B. and Lanckriet, G., (n.d.) THE NATURAL LANGUAGE OF PLAYLISTS.

Nicholas, I.S.C., (1999) Combining Content and Collaboration in Text Filtering. *undefined*.

Okura, S., Tagami, Y., Ono, S. and Tajima, A., (n.d.) Embedding-based News Recommendation for Millions of Users. *KDD*, [online] 17. Available at: http://dx.doi.org/10.1145/3097983.3098108 [Accessed 9 Nov. 2021].

Paradarami, T.K., Bastian, N.D. and Wightman, J.L., (2017) A hybrid recommender system using artificial neural networks. *Expert Systems with Applications*, 83, pp.300–313.

Pazzani, M.J. and Billsus, D., (2007) Content-Based Recommendation Systems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, [online] 4321 LNCS, pp.325–341. Available at:

https://link.springer.com/chapter/10.1007/978-3-540-72079-9_10 [Accessed 9 Nov. 2021].

Robertson, S. and Walker, S., (2000) Threshold setting in adaptive filtering. *Journal of Documentation*, 563, pp.312–331.

Teinemaa, I., Tax, N. and Bentes, C., (n.d.) Automatic Playlist Continuation through a Composition of Collaborative Filters.

Ungar, L.H. and Foster, D.P., (1998) Clustering Methods for Collaborative Filtering. [online] Available at: http://www.sims.berkeley.edu/resources/collab/ [Accessed 9 Nov. 2021].

Volkovs, M., Rai, H., Cheng, Z., Wu, G., Lu, Y. and Sanner, S., (2018) Two-stage model for automatic playlist continuation at scale. *ACM International Conference Proceeding Series*.

Xiong, R., Wang, J., Zhang, N. and Ma, Y., (2018) Deep hybrid collaborative filtering for Web service recommendation. *Expert Systems with Applications*, 110, pp.191–205.

Yang, H., Jeong, Y., Choi, M. and Lee, J., (2018) MMCF: Multimodal collaborative filtering for automatic playlist continuation. *ACM International Conference Proceeding Series*.

Zarzour, H., Al-Sharif, Z., Al-Ayyoub, M. and Jararweh, Y., (2018) A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques. *2018 9th International Conference on Information and Communication Systems, ICICS 2018*, 2018-January, pp.102–106.

Zarzour, H., Jararweh, Y. and Al-Sharif, Z.A., (2020) An Effective Model-Based Trust Collaborative Filtering for Explainable Recommendations. *2020 11th International Conference on Information and Communication Systems, ICICS 2020*, pp.238–242.