

# TEMPORAL GRAPHS FOR MUSIC RECOMMENDATION SYSTEMS

## MOTIVATION

"Nearly 616.2 million people listen to their favorite artists or discover new ones via online streaming platforms" [1]. Hence, streaming companies seek to enhance the user experience by offering personalized music recommendations, moreover, users value personalization as a top feature [3]. At the same time, music streaming services are able to track individual musical choices meticulously, thus, a growing volume of data on multiple user's musical preferences is available. In response, we aimed to develop a recommendation model that predicts the genres a user is expected to like

## MODELING APPROACH

We worked with a temporal evolving graph from Temporal Graph Benchmark (TGB) [2] datasets. TGB provides two data sets: First, a graph with 992 users and 513 music genre represented as nodes and weighted edges that indicates a user listens to a music genre at a given time. The temporal graph evolves over the span of 4 years. The dataset posed a challenge since the description given by the creators of the dataset did not match it. Additionally, it has anomalies and multiple duplicates. A second dataset derives from the first, a dynamical matrix of users and genres where the interaction of each user and each genre is normalized over the span of a week. We train our models on this second dataset. Our goal is predicting the top 10 genres a user is likely to listen to in the following weeks of the training set.

**Train-test-split:** We split chronologically into the train, validation and test set with 70%, 15% and 15% of the edges respectively.

**Models:** We experimented with several time series models. For the baseline model we chose the latest node label, i.e., for each user and genre in the test dataset, we look at the latest weight of that user for that genre in the training set and use that as the prediction. We looked at the time signal of the user genre weights for many users and did not find any seasonality or trend. Moreover, it was not feasible to detect seasonality or trend for all 992 users with a scalable algorithm.

Then we tried rolling average and did hyperparameter tuning with the window parameter. Finally we tried exponential smoothing with hyperparameter tuning on the smoothing parameter.

**Key performance indicator:** We use the normalized discounted cumulative gain with 10 items (NDCG@10). It is a holistic metric that accounts for both the relevance of the item and the rank of the prediction compared to the ground truth.

## RESULTS

Model	NDCG@10 (Validation)	NDCG@10 (Test)
Latest genre rating (Baseline)	0.18068	0.1575
Mean genre rating	0.2310	0.2034
Rolling Average (Window= 7 days)	0.2242	0.1951
Rolling Average (Window= 14 days)	<b>0.2333</b>	0.2008
Rolling Average (Window= 21 days)	0.2345	<b>0.2014</b>
Exponential Smoothing ( $\alpha = 0.8$ )	0.1941	0.1662
Exponential Smoothing ( $\alpha = 0.4$ )	0.1827	0.1619

## NEXT STEP & BENEFITS TO STAKEHOLDERS

We want to exploit the graph structure of our data to train graph neural networks and learn the interaction of users and music genres. Also, train classical and deep learning multivariate time series models to analyze the music genre preferences of one user. Ultimately, offer streaming services a recommendation system that can create a great and unique experience for each user.

[1] Götting M. "Music streaming worldwide - statistics & facts". 10 Jan 2024, <https://www.statista.com/topics/6408/music-streaming/#topicOverview>. Accessed May 28, 2024.

[2] Huang, S., et al. "Temporal graph benchmark for machine learning on temporal graphs." *Advances in Neural Information Processing Systems*, 2023. Preprint: [arXiv:2307.01026](https://arxiv.org/abs/2307.01026), 2023.

[3] Spotify. "Understanding recommendations on Spotify". <https://www.spotify.com/us/safetyandprivacy/understanding-recommendations>. Accessed May 30, 2024.