

A RecSys Study in Osu!

Introduction

I enjoy playing rhythm games, and so do many others. My preferred one is also the most popular—the classic osu! game. Osu! is now approaching its 20-year anniversary. There are still around 2.5 million monthly active users and about 40 million registered accounts. By now, nearly 50,000 officially vetted (ranked) community-created beatmap sets exist, with at least a million more "loved" or "graveyard" sets.

Considering the size and age of the community, it's surprising how much room for improvement there is in how players discover beatmaps. The official repository offers only basic filters, so discovery generally boils down to:

- filtering by new or most played
- looking up specific songs (and hoping someone mapped them)
- users sharing and recommending in forums, multiplayer lobbies, or through playlists/collections

Special mentions:

Osu! Collector

<https://osucollector.com>

A third-party service for sharing beatmap collections, often organized by skillset. Currently has 42,300 users and 12,600 shared collections.

Stream/Stamina Map Bot

<https://ost.sombrax79.org/commands>

Powered by a large, community-submitted collection focused on stamina and stream maps. Its just special filters, not recommendations.

Osufy!

<https://osufy.lonke.ro>

Finds beatmaps for songs in your Spotify playlists and works reasonably well—implicitly using Spotify's recommendation system.

Note: Songs you like \neq Beatmaps you like, so this doesn't fully solve the core issue.

How to make this better

To me, there are two features that could immediately improve the process of finding beatmaps users love to play.

1. Beatmap "Type" labeling

Beatmaps are often perceived as belonging to certain types, based on one or more skillsets they demand from the player. While this is to some extent subjective, not officially attached to any map, and entirely community-driven, it's a fundamental part of the osu! experience. Almost every player has preferences for certain types, and those preferences often evolve. Collections are usually organized around these types, tournaments select maps with them in mind, and many high-level players become specialists.

According to this discussion:

https://www.reddit.com/r/osugame/comments/1kkj2zr/what_makes_you_enjoy_a_beatmap_how_well_does_osu/,

most users rank skillset enjoyment as the most—or one of the most—important factors in how much they enjoy a beatmap.

So to me, it's an obvious conclusion that beatmaps should be tagged, giving users hints about what to expect before downloading and test-playing them. This was implemented in a within the new osu lazer client recently, as a community driven feature, but there should be an algorithmically driven solution.

This issue has been raised multiple times in the past:

<https://osu.ppy.sh/community/forums/topics/1928067?n=19>

https://www.reddit.com/r/osugame/comments/1c3uo8s/map_tags/

https://www.reddit.com/r/osugame/comments/1dcz7ml/opinions_on_new_tagging_system/

- The Spanish osu! wiki offers a very advanced and technical set of classification labels:
https://osu1.roanh.dev/wiki/es/Beatmap/Beatmap_tags
- Others prefer a simpler system aimed at casual players—e.g. a spider chart with fundamental skills like "aim", "stream", "tapping", "reading", "jump", and maybe "speed":
<https://osu.ppy.sh/community/forums/topics/1928067?n=21>

The most promising project I'm aware of that attempts automatic classification is **osu_oracle**. It's extremely basic right now, but I'd like to test it myself when I get the chance:

https://github.com/token03/osu_oracle

2. A RecSys (Recommender System) for Osu! Beatmaps

Anyone familiar with recommender systems and their impact on users (when done well) won't question why this should absolutely exist—officially as well. For those less familiar, here's how it could help osu! specifically:

- Find beatmaps you'll actually enjoy, in a sea of options with poor filter support. No need to rely on chance forum posts or multiplayer lobbies—let the RecSys do it for you.
- Help lesser-known mappers and maps get discovered. Popularity bias is a real issue in osu! A RecSys could surface niche maps that match your taste.
- Avoid frequent "total misses"—no more rage-quitting maps that sound cool but when you try them, they are clearly built for DT farming, dense overlap patterns, or other things you dislike.

Previous Work:

Tillerino

A bot that can join in-game lobbies and recommend maps based on your top 10 plays.

<https://github.com/Tillerino/Tillerinobot/wiki>

Using top plays as suggestions isn't ideal—it heavily biases toward farm maps.

Dynam1cBOT

Analyzes your top plays to create a "skill fingerprint" using many features, then compares it to a large map database to recommend three maps: one easier, one matched, and one harder. It selects the skill-matched

one 80% of the time.

A hybrid recommendation approach, as far as I can tell.

AlphaOsu!

<https://alphaosu.keytoix.vip/self/pp-recommend>

Also based on top plays, fully focused on competitive players—recommending maps to maximize pp gain, not enjoyment.

Work by m1tm0

Around the time I concluded, that there is serious room for improvement, a user named *m1tm0* gained attention by announcing his own RecSys project on Reddit. He discussed both plans and challenges:

1. https://www.reddit.com/r/osugame/comments/1kkj2zr/what_makes_you_enjoy_a_beatmap_how_well_does_osu/ – Discussion on the difficulty of capturing “enjoyment” and a proposal for an enjoyment metric.
2. Same thread – Further elaboration on how ratings are constructed via explicit and implicit signals, what model was used at the time, and some example recommendations.
3. https://www.reddit.com/r/osugame/comments/1kqjags/osu_recsys_looking_for_alpha_testers/ – Model was switched to Bayesian Personalized Ranking for scalability; call for alpha testers.

I’m in contact with the user, and a lot of groundwork is being laid through this project. It will be referenced in later sections. At this point, I can confirm that the first results are promising—already demonstrating the feasibility of this whole endeavor. It already takes domain-specific details like mod sensitivity into account and currently offers a Discord bot with many useful tuning options.

Motivations for Contributing:

- Contribute to the refinement of third-party recommendation tools, increase awareness and adoption in the community, and demonstrate interest in the feature, hopefully encouraging the osu! developers (or rhythm game developers in general) to adopt such systems.
- Unlock a unique domain to run RecSys experiments and test research questions, not least as part of a project for my RecSys course at the University of Innsbruck.

Discussion of available data:

We have two datasets: one with 10,000 random users and one for the top 10,000 users. They vary in the number of “interactions” (scores). The numbers below are based on the top player dataset, the random palyer has a little less.

Due to the massive size, I collected and organized all data into a MySQL database. We’ll need to set up a robust system to extract only what’s needed.

Items (Beatmaps) [~200,000 ranked beatmaps—should be most or all in existence]

Ranked only—vetted beatmaps. Loved, graveyard, etc., not included.

- **.osu** files in formats v1–v14, with metadata and geometric hit-object data [~6 GB]
- Info file with 25 basic attributes
- **Beatmap Set** table with Set Information [48,000, all ranked sets]
(Beatmaps are always grouped in a set - same mp3, multiple difficulties)

- Fail/exit mask for each map (split into 100 chunks: where do users quit/fail?) [~4 million lines—not all mod combinations]
- Mod difficulty mask [~7 million lines]
- Meta-attribute features for each **beatmap + mod** combo, e.g., *aim, strain, speed,..* [~64 million lines]
- MP3 files (not yet)

Users [10,000]

- User stats table with ~25 stats
- Full profile (not pulled yet—doesn't seem to add much)
- Liked beatmaps (not pulled yet—valuable as explicit feedback)

Interactions [at least 50,000,000 — exact count TBD]

- Play count table for how often a user played each map [~75 million lines]
- *scores_high* with basic performance stats [~43 million entries]
- *scores* table with similar features [~34 million entries]
(Likely partial import—stopped at ~65%. Need to check difference vs. *scores_high*)

Basically for each Beatmap (+ Mod Combination that changes difficulty) the best score of each user seems to be included.

Discussion of (to me) reasonable approaches:

1. Collaborative Filtering

Already exists in several approaches with varying degrees of success

Main Challenges:

- **Beatmap/Mod interaction** – Mods alter both the experience and especially the difficulty of a map. Should each Map + Mod combination be treated as a separate item?
- **Farm Bias** – Especially present in the top 10,000 user set, but generally common. Maps that give high pp for relatively low effort—so-called farm maps—are disproportionately overrepresented due to repeated plays and grinding.
- **Popularity Bias** – Covered earlier, but worth repeating: very pronounced in osu!.
- **Sequential Nature** – The data is inherently sequential and tied to player progression. Changes in skill and preference over time should ideally be captured.
- **Enjoyment** – How do we interpret interactions? What defines an enjoyable recommendation? User mindsets vary from recreational to competitive. As discussed in https://www.reddit.com/r/osugame/comments/1kkj2zr/what_makes_you_enjoy_a_beatmap_how_well_does_osu/, enjoyment is complex, subjective and varies across skill levels. A good metric is needed. Something like *#Interactions = Enjoyment* will **not** suffice.
- **Hard/Soft Constraints** – To be discussed in the following Research Questions section.

2. A Content-Based Approach

(I'm especially excited about this—maybe not best suited for this project)

Motivation:

Beatmaps are extremely feature-rich items: they combine a music track, gameplay mechanics, competitive attributes, and (in theory) leaderboard context. It's clear players can develop very specific tastes. True Content-based recommendation (Capturing Mapping or musical features, and not just the superficial attributes) has never been seriously attempted in this domain. Recommending Music alone is already complex enough.

A proper hybrid system is likely out of reach for now, but an isolated content-based approach could be a first start. It's also a great way to avoid the strong biases present in the interaction data.

And someone has to make the first attempt, right?

Proposal:

A "Beatmap Radio" (like Spotify's)

Given a sample beatmap, return a set of similar ones. Some more similar in music, some more in gameplay.

To capture the complexity of beatmaps, a deep learning approach seems most viable. **Maybe something like this:**

1. Scrape labels for each beatmap using osu_oracle, osu!lazer client tags, and user collections from osu!Collector.
2. Create beatmap classes from these labels. *(Heavy preprocessing—this won't be easy.)*
3. Do the same for musical features.
4. (Optional) Combine musical and gameplay features—if possible. *(Still hard to imagine, but maybe doable.)*
5. Use contrastive learning to transform beatmaps into an embedding space, where proximity aligns with shared labels (for gameplay, music, or both).
6. To find similar beatmaps: locate one in embedding space, return nearest neighbors.
7. Evaluate using a test set—e.g., maps frequently appearing together in osu!Collector label-based collections should be close neighbors.

Discussion of possible Research Questions:

Achieving solid recommendations in such a complex and new domain is a challenge in itself, but beyond that we want to answer some research questions of course, and this field definitely opens up some interesting ones.

Some ideas:

Applicable to CF-Approach

1. Study on Implementing User-Specific Hard/Soft Recommendation Constraints

Recommendations in osu! must fall within a narrow window of what's viable for a user.

- Difficulty range must avoid both boredom and frustration. Static difficulty ratings (Star-Rating) might not be enough, as players often have vastly different skill levels across skillsets (see Beatmap

Labeling).

- Physical constraints matter: players have limits in approach rate (reading) and max hitrate (especially sustained), and others aswell. These limits improve gradually but can be pushed only by a marginal amount. From the Outside, I bet you would be surprised how hard these limits are for most players at any given time.

Outline:

1. Establish a recommendation baseline.
2. Implement constraints in various ways (filtering, cost functions) across different models.
3. **Do user-specific constraints based on domain knowledge improve recommendations?** If so, what's the best way to implement them?
4. Explore transferability to other domains with similar constraint dynamics.

2. Multidimensional Recommender System to Satisfy Players with Different Motivations

(Dont know much about these systems yet)

Players are generally speaking on a spectrum from recreational to competitive. For this approach use the *random_users_dataset*. All top 10000 Players would be pretty competitive.

1. Start with a single-score baseline (e.g. some combination of: #interactions, pp, accuracy). It won't fit all user types equally.
2. Try using a multidimensional recommender (they learn user-specific weights for each dimension (I think)).
3. Evaluate using a reward function, different from the training metrics:
 - For farm-heavy maps: high pp achieved and fewer retries are good;
 - likes are a strong signal always.
 - For non-farm maps: more retries might signal enjoyment; maybe aim for 95% accuracy as an engaging soft target (from personal experience).
4. Try transferring to another dataset where user motivations are fundamentally different aswell.

Applicable to CB-Approach

The approach itself is complex enough that just testing feasibility could be a valid research question. Beyond that, a few extensions:

These aren't very generalizable yet, still working on it.

1. Does Contrastive Learning Capture the Multilabel Nature of Beatmaps?

While a beatmap is often mainly one archetype, it's more accurate to think of it as a distribution across multiple labels—especially with finer-grained types (see: https://osu1.roanh.dev/wiki/es/Beatmap/Beatmap_tags).

2. Is the Embedding Space Meaningful Enough to Allow Transformations?

Do dimensions in the embedding space align with how players perceive maps? Can we move in the space in useful ways—e.g., similar music but a jump map instead of a stream map (combined space), or “similar map but more technical patterns”?

