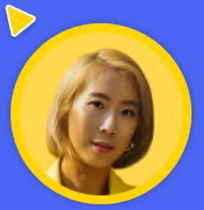
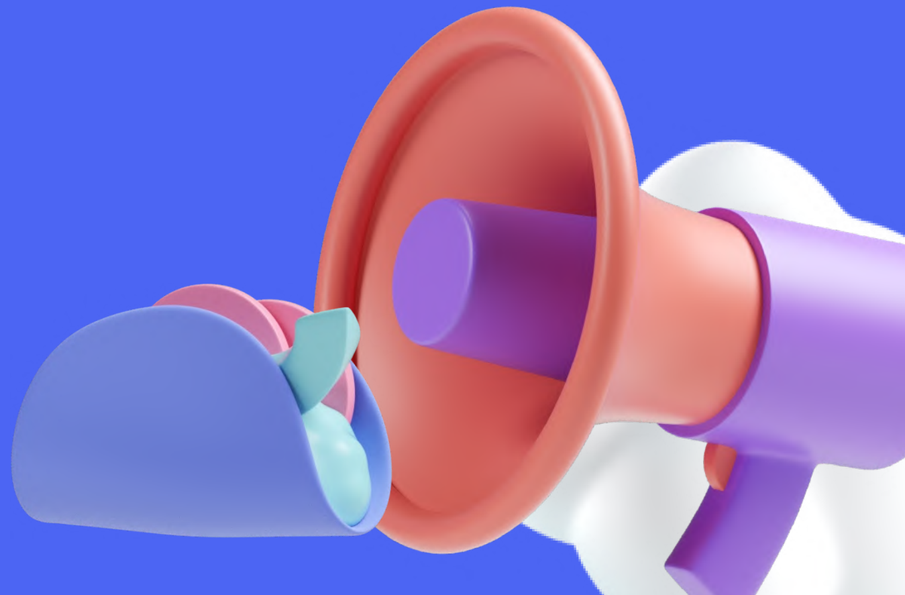


Steam Video Games



Context

Steam is the world's most popular PC gaming hub, with over 6,000 games and a community of millions of gamers. With a massive collection that includes everything from AAA blockbusters to small indie titles, great discovery tools are a highly valuable asset for Steam.

Dataset

The original dataset has 199,999 observations and contains the following values:

- ***'User ID'*** (numeric) - A collection of unique user ID .
- ***Game Name*** (string) - The name of each game played by the user, unordered.
- ***'Purchase'*** (string) - containing two values: A variable containing two values: *purchase*, or *play*.
- ***'1.0'*** **-** Hours that the user played each game.
  - Values equal to *1.0* represent a user purchasing a game.
- ***'0'*** **-** An extra variable containing only 0's, holding no information.

Dataset Preparation

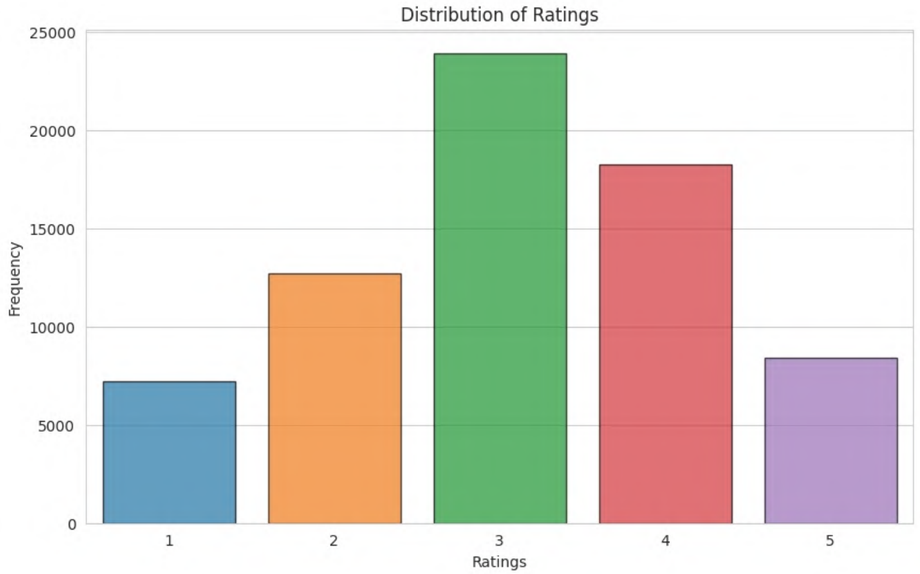
- '***Ratings***' variable was generated using base-weights to create a somewhat normal distribution with a left skew - to represent real-life results.

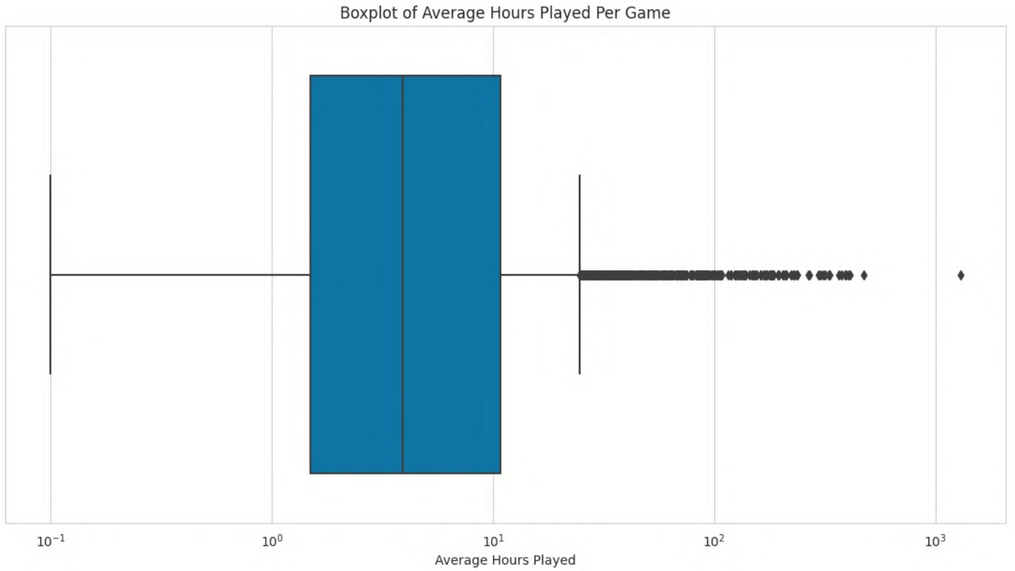
- After, we used two target variables that show user interaction/ratings: '***Ratings***' and '***HoursPlayed***' variables.

Extra variables from another dataset were added in order to give our recommendation engines more depth. These variables are:

- '***Metadata.Genres***' **-** the genre of each video game (string).
- '***Release.Year***' -the release year of each video game (integer).

Exploratory Data Analysis





'Ratings' Variable:

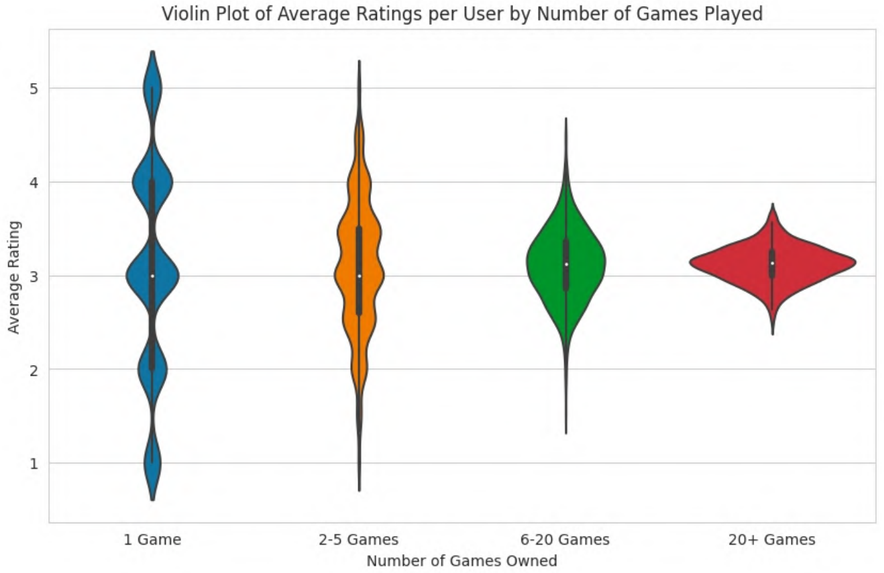
- Slightly left-skewed distribution.
- Majority of ratings fall between 2-4 - a central tendency bias and avoidance of extreme ratings (1 or 5).

Average 'HoursPlayed' per Game Box Plot:

- Half of the observations show users that spend from around 2-10 hours on average per game.
- Usual observations reach up until approximately 30 hours. Values higher than that will be considered '***heavy users***'.

Exploratory Data Analysis

	Game	Number of Players	Average Rating
3	Dota 2	4841	3.089857
7	Team Fortress 2	2323	3.106759
1	Counter-Strike Global Offensive	1377	3.076979
9	Unturned	1069	3.077643
5	Left 4 Dead 2	801	3.086142
2	Counter-Strike Source	715	3.146853
8	The Elder Scrolls V Skyrim	677	3.031019
4	Garry's Mod	666	3.124625
0	Counter-Strike	568	3.140845
6	Sid Meier's Civilization V	554	3.095668



Top 10 Most Played Games and their Average Ratings.

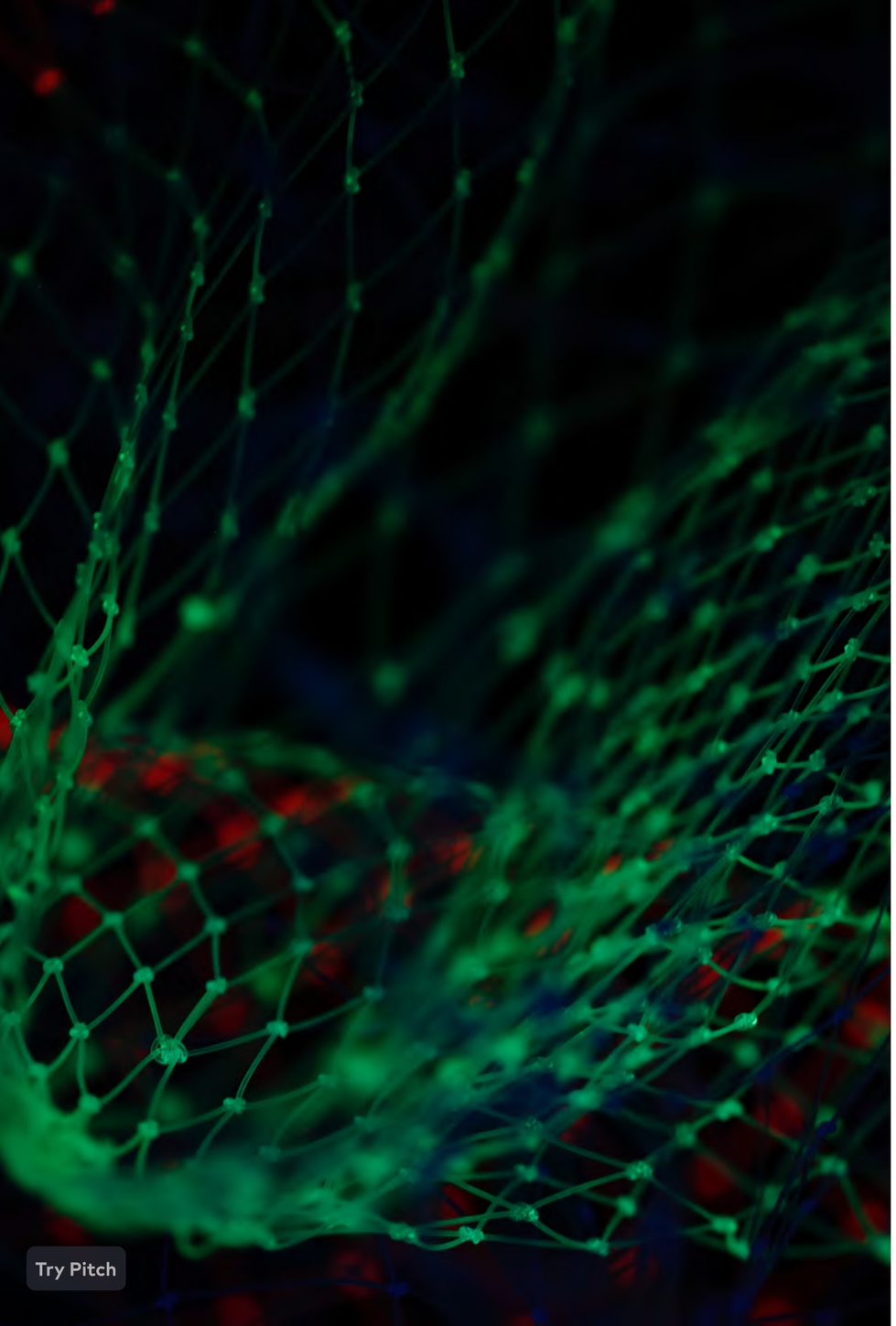
- Despite games being the most played, their average ratings is around ~3.
- Could be due to the large number of observations.

Average Rating per User by Number of Games Played:

- Players of only 1 game show variety in their ratings.
- We see that the more games someone plays, the more ratings are centralized around middle values (instead of extremes.

Our Recommendation Engines

Date



Cold Start

The cold start problem in the context of gaming hubs occurs in two instances:

- User-Based Cold Start: When a new user signs up and there is no known info about the user for the recommendation engine to promote games.
- Content-Based Cold Start: When a new game is posted and there is no ratings or reviews about the game

How will these be solved?

- User-Based Cold Start:
  - The games with the highest cumulative gameplay hours are selected. Then, out of those games, the top 10 games with the highest average rating are selected. These games will be advertised to the new user
- Content-Based Cold Start:
  - The newest games will be released in the 'new-games' section. It is the responsibility of the creator to advertise the games on external sources.

Non-Personalized Rec. Engine

A **non-personalized recommendation** engine provides general recommendations that are not tailored to the individual preferences of a user.

In the context of Steam, our non-personalized recommendation engine takes into account three variables that are converted into a rating matrix:

- *'User ID'*
- *'Game Title'*
- '*Rating (1 - 5)*'

This rating matrix is then fed into a recommendation engine using the "Popular" method. This engine simply recommends the most popular games. Additionally, this engine had a true positive rate of 71%, meaning that it recommended correctly about 7 out of 10 games that it should have recommended.

User-Based Collaborative Filtering Engine

A user-based collaborative filtering engine in the context of a video game vending service would be finding similar users and recommending them non-mutual video games in their respective libraries. In order to provide more information to our recommendation engine, the two datasets previously were merged.

Our engine functioned as such:

- The variables Game Title, User ID, Hours Played, Ratings, Genres and Release Year were taken into account
- A rating matrix was made, and then split into train and test sets
- The recommender model was made under the "UBCF" model, and predictions were run

RMSE in recommendation engines is a measure of the average magnitude of errors in the predictions of the engine

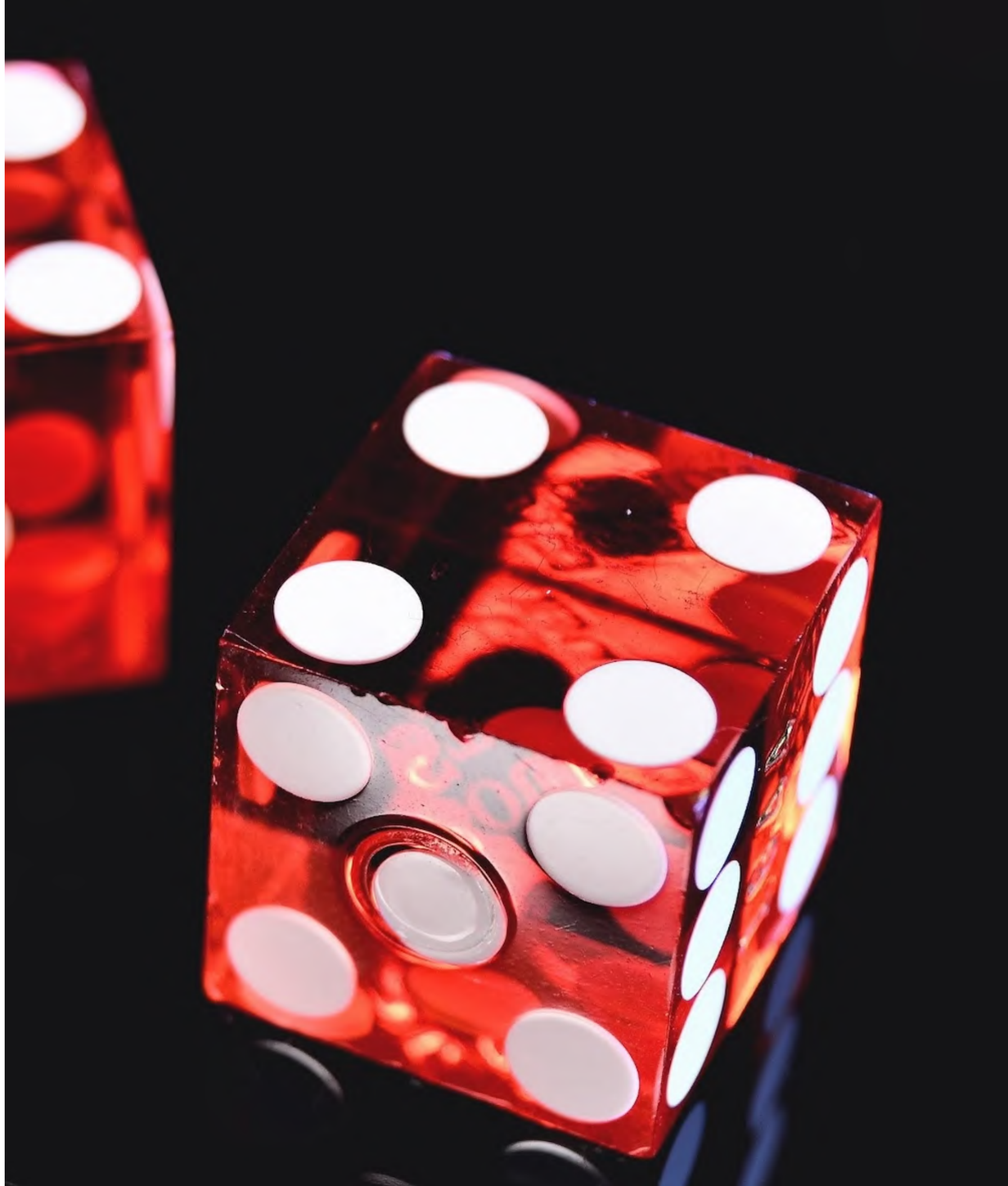
- After much trial and error, the best RMSE score returned was a 68.03.

Item-Based Collaborative Filtering Engine

An **item based-collaborative filtering** engine is similar to the user-based one, but it finds similar items instead of users.

The same variables were used as the previous engine.

After many trials, the best RMSE was 70.



Content-Based Recommender

Content-Based Recommender suggests items to users based on their preferences and the features of items. With the following models, we aimed to predict the variables 'Ratings' and 'HoursPlayed'.

Tested ML Models: Linear Regression, Random Forest, XGBoost, and Neural Network.

Performance Metrics

- Hours Played → user engagement & Ratings → Item
- Hours Played prediction models show overall better performance on MSE and MAE.
- Linear Regression has performed the best out of all other models, as it has the least negative R-squared value along with the lowest MSE and MAE.
- Predictions through rating hinder the models'performances to negative R-squared values.

Top Performers

1. Linear Regression for predicting "Ratings".
2. Neural Networks for predicting "HoursPlayed".
3. XGBoost for predicting "HoursPlayed".

Thank you!