Background

Indie games, created by small teams or individuals, offer innovation and creativity in the gaming landscape. Comprising only 40% of total game revenue with over 12,300 contributing to this share while 60% of total game revenue are coming from 100 AAA titled games. Limited marketing budgets, with 80% of developers lacking dedicated funds, pose challenges, leading indie studios to rely on grassroots efforts.
However, indie games consistently excel in quality and critical acclaim, with 8 out of the top 10 highest-rated games on platforms like Steam being indie titles. This success underscores the talent and dedication of indie developers. Yet, proving that indie-game market has potential to become a significant and distinct sector within the broader gaming industry.

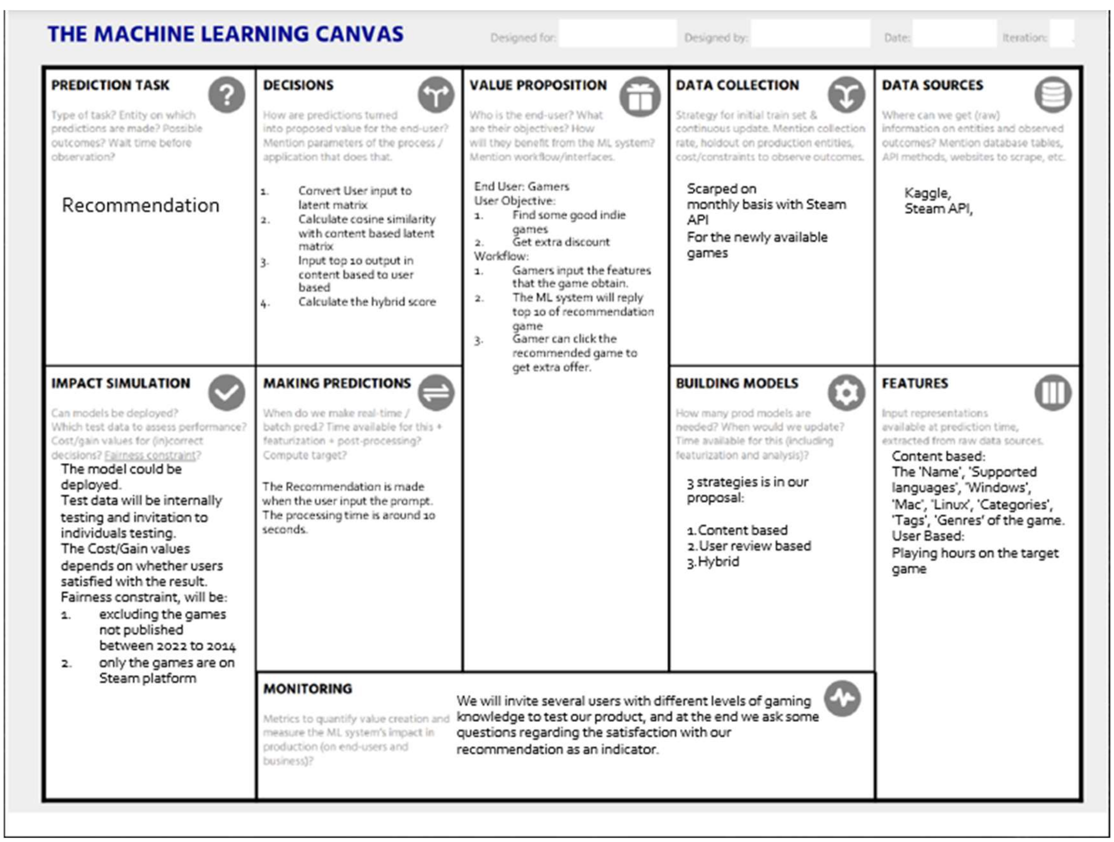
Problem statement

1. Players tend to stick with famous genres & IP, limiting their exploration of new games.
2. Current game recommendation algorithm creates exposure blockades to games with low marketing budget.
3. Indie games struggle to enter the market.

Significances

This business project holds significant potential in several key areas. Firstly, it aims to extend the spectrum of consumer habits by encouraging users to explore a broader range of gaming options. By providing a platform that supports low-budget games, it not only boosts sales for developers operating on a limited budget but also offers consumers access to a diverse array of titles they might not encounter through traditional marketing channels.
Furthermore, it serves as an alternative marketing platform, addressing the limitations faced by developers on larger platforms like Steam, where visibility often favors titles with substantial marketing budgets. By offering a low-cost solution for developers to promote their games, this project enhances industry diversity and reduces sales barriers for underrepresented groups within the gaming community.

ML Canvas



Data Set Analysis

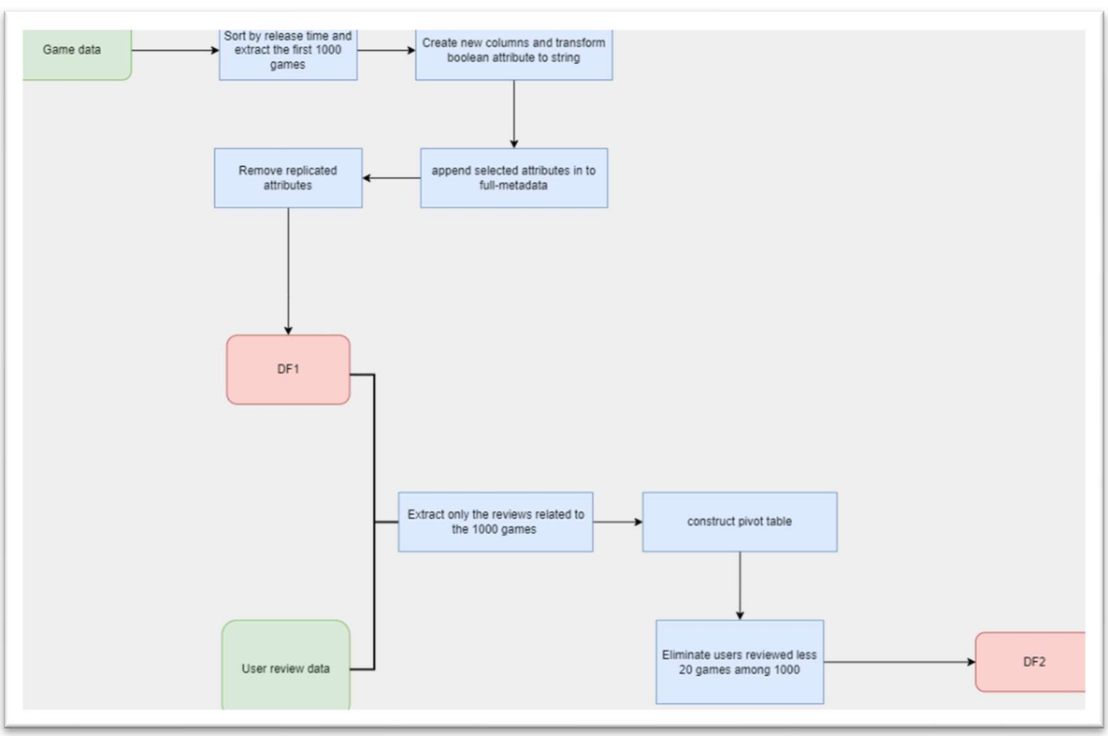
The data set we picked has 85103 entries which represent 85103 games on the steam platform, we extract 1000 entries considering the computational constraint in this project after sorting the entries with their publishing time. Every entry in the dataset has 38 attributes and we select the following as the attributes we will use:

1. AppID : ID of the game in Steam Database
2. Name: Name of the game
3. Supported language: What languages does the game support.
4. Windows: if the game supports Windows OS
5. Mac: if the game supports Mac OS
6. Linux: if the game support Linux
7. Categories: the categories identifier strings
8. Genres: the genre's identifier strings, which is the upper level of the Categories
9. Tags: the tags that have been given to the game on steam

Then, since Categories, Genres and Tags are not a compulsory metadata while the game is launched on steam platform, there will be null value, therefore we perform an empty string imputation to these 3 columns. Then we use define a function that we learn in the ML2 lectures to turn the Boolean values while they are True. Then we appended all the attributes except Name and AppID into the "full_metadata" column and output as the CSV.

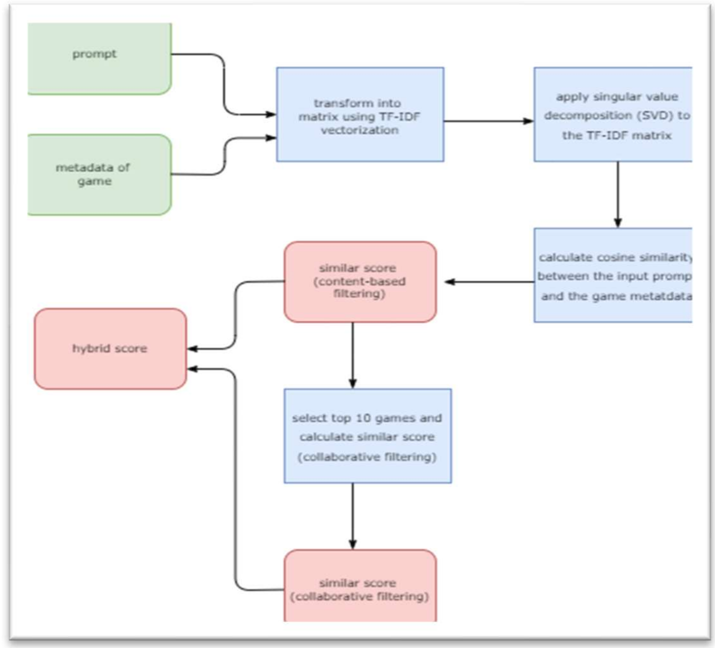
For the user review dataset, it has 41154794 entries of 13781059 individual players, on 37610 individuals games, and we extract the entries related to the 1000 games, and we also set a threshold on picking users, which is they need to at least comment on 10 games to be considered as effective entries. After cleaning, we perform a pivot table on column is "user_id", row is "app_id" and value will be 'playing_hour', and we will have 1000 row with 540 columns as the review dataset and output as CSV.

Data Preprocessing

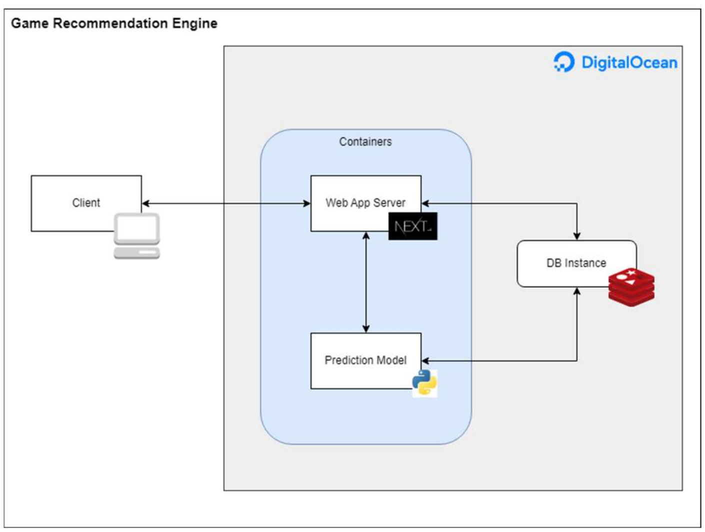


Model Benchmark

We implement a metric called Hybrid Score as the resultant prediction of the model. Hybrid Score consists of 2 sub-metrics, Content-based filtering, which is computed using cosine similarity, and Collaborative filtering, computed with varying weights based on recommended games' rankings from Content-based filtering. We take a 0.5 weight from each filtering and obtain the Hybrid Score result.



Deployment



We use Digital Ocean as our cloud platform. A Redis instance was used as database to store dataset and user data for the web server. 2 containers are created, one is initialized with Next.js image, this container focuses on rendering web-based GUI. The prediction model was contained in another image with Python environment. Those 2 containers communicate via RESTful APIs, while delegate APIs was used when communicating with the Redis instance.