GameAdo

A Game Recommendation System

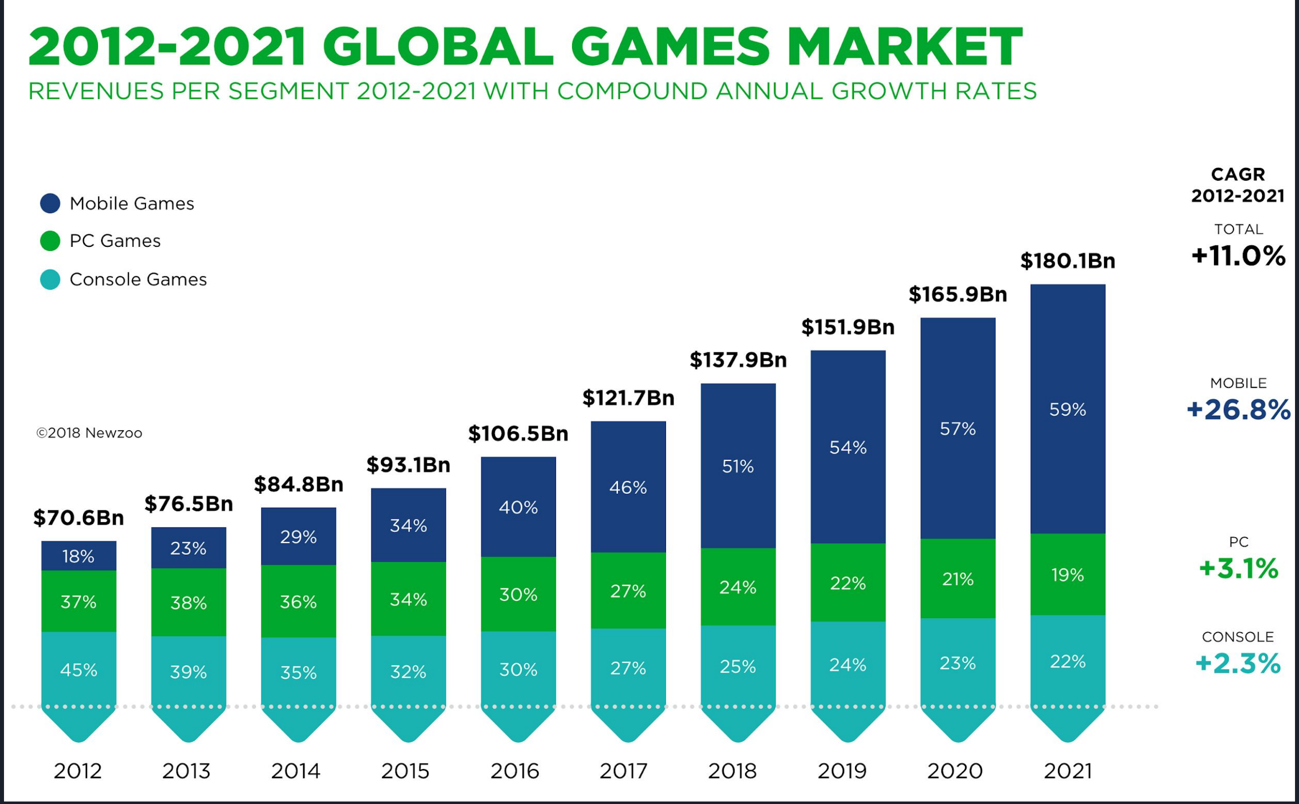
Objective

- To present a recommendation system for the community of gamers.
- Enhance the user satisfaction.
- Adds value to both the gamers and the game distributors.
- Need of a good recommender system

Need of A Recommendation System

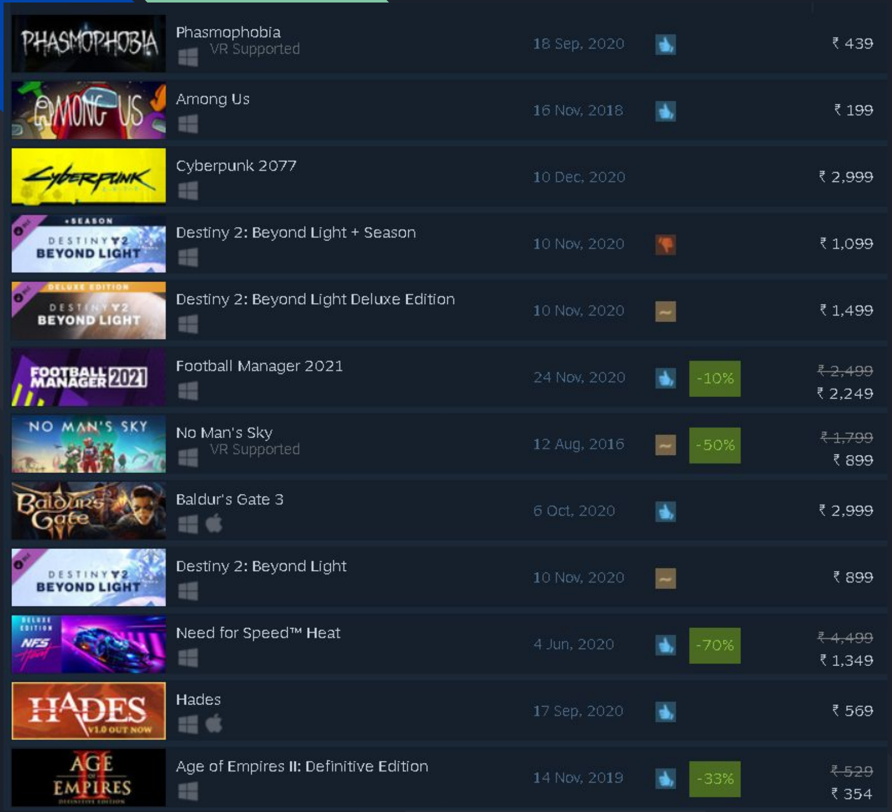
- The global video game market size was valued at USD 151.06 billion in 2019 and is expected to grow at a Compound Annual Growth Rate (CAGR) of 12.9% from 2020 to 2027.
- Rising inclination from physical games to online games.
- The growing penetration of internet services coupled with the easy availability.
- Adoption of gaming as an educational tool.
- The Booming Gaming Industry.

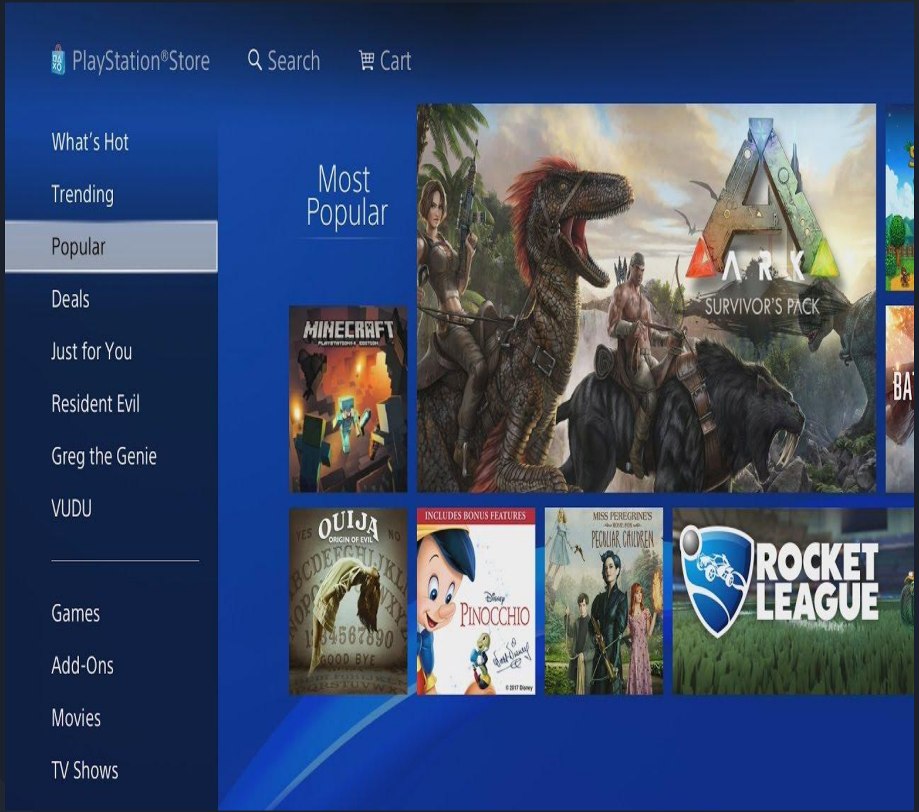
The Booming Gaming Industry



Reference: Check [here]

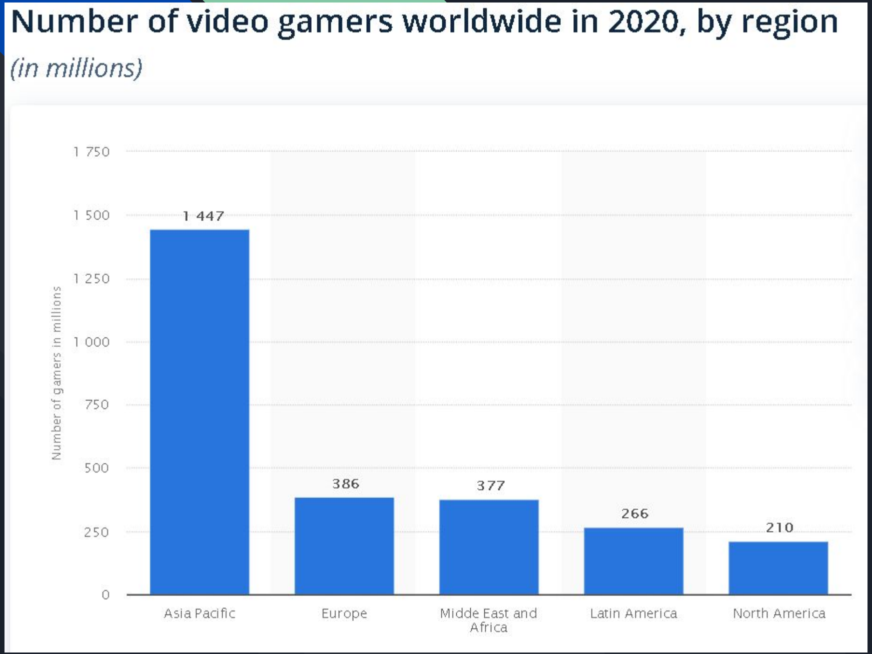
So many choices...!!






PLAYERUNKNOWN'S
BATTLEGROUNDS
FORTNITE
ENDGAME
COUNTER STRIKE
LEAGUE of LEGENDS
MINECRAFT
DOTA 2
HEARTHSTONE
APEX
LEGENDS
THE DIVISION 2
Splatoon 2

HUGE user base



- An average Gamer spends about **9 minutes per week** searching for a proper game to play.
- With over **2 billion players** across various platforms.
- So that sums up to **15.6 billion hours** wasted **per year** searching for games.

Reference:
Click here.

Top Players in The Gaming Industry

By Valve
95 Million

By EA sports
40 Million

By Ubisoft
50 Million

By Sony interactive Entertainment
110 Million

By Microsoft
90 Million

By Epic games.Inc
61 Million

These are some of the top Gaming Distribution Companies in the gaming industry.

Why Steam?

- We Choose Steam for all our Data Collection.
- Steam is an online, cross-platform game distribution system, with around
    - Over 95 million active users
    - Around 1 billion total accounts
    - hosting over 40000 games
- The dataset contains records from over 100,000 games and applications.
- Steam has also expanded into an online web-based and mobile digital storefront.

Data Collection

- We used the Steam Web API to collect data for all the applications present on Steam platform.
- We also used two freely available datasets from Kaggle, (i.e. user and application datasets) as temporary datasets.
- Similar to Kaggle we extracted two datasets using the API:
  - Application Dataset (details regarding application and games)
  - User Dataset (user reviews, hours played, ratings etc.)

Application Dataset

	app_id	name	type	platforms	categories	genres	recommendations
0	5	NaN	NaN	NaN	NaN	NaN	NaN
1	7	NaN	NaN	NaN	NaN	NaN	NaN
2	8	NaN	NaN	NaN	NaN	NaN	NaN
3	10	Counter-Strike	game	windows:mac:linux	Multi-player:PvP:Online PvP:Shared/Split Scree...	Action	94916.0
4	20	Team Fortress Classic	game	windows:mac:linux	Multi-player:PvP:Online PvP:Shared/Split Scree...	Action	3597.0
...	...	...	...	...	...	...	...
104004	1456550	The Tower Of TigerQiuQiu Soapbubble	dlc	windows	Single-player:Downloadable Content	Action:Casual:Indie	NaN
104005	1456850	Solicitude Wake-up Demo	demo	windows	Single-player:Game demo	NaN	NaN
104006	1457260	Masters of Puzzle - Halloween Edition: Undeadl...	dlc	windows:mac	Single-player:Downloadable Content:Steam Achie...	Casual:Indie:Simulation	NaN
104007	1457270	Masters of Puzzle - Halloween Edition: Pumpkin...	dlc	windows:mac	Single-player:Downloadable Content:Steam Achie...	Casual:Indie:Simulation	NaN
104008	2028850	Bioshock Infinite: Columbia's Finest	dlc	windows:linux	Single-player:Downloadable Content:Steam Achie...	Action	NaN

104009 rows × 7 columns

- Contains more than 1 Lakh application data, which includes games, demos, trailers, softwares, downloadable contents (dlc) etc.
- Only half (around 48,000) of the apps in the dataset are games.

User Dataset

	app_id	steam_id	playtime_forever	language	voted_up
0	10	76561198116800044	1025.0	english	True
1	10	76561198897181049	1995.0	english	True
2	10	76561198272044528	7070.0	english	True
3	10	76561199042823737	614.0	english	True
4	10	76561198992482648	6143.0	english	True
...	...	...	...	...	...
4166536	2028850	76561198150653411	0.0	english	True
4166537	2028850	76561198181450110	0.0	english	True
4166538	2028850	76561198253602926	0.0	english	True
4166539	2028850	76561197966220495	0.0	english	True
4166540	2028850	76561198115096726	0.0	english	True

4166541 rows × 5 columns

- Contains more than 4 Million user reviews.
- We collected around 1000 reviews for each application.
- Each individual user review contains:
  - The application id for which the user has given the review.
  - The user's unique Steam ID.
  - The number of minutes the user has played the game (also include inactive minutes).
  - The language in which the review is written.
  - And finally whether the user recommended the game or not.

Data Cleaning

- For both the datasets we remove the application ids that are anything other than games.
- For the user dataset we removed the entries for which the total playtime is zero.
- We also dropped columns which are of no use for this project (such as language, type etc.), converted the data types and manipulated the data accordingly.

Anomaly detection

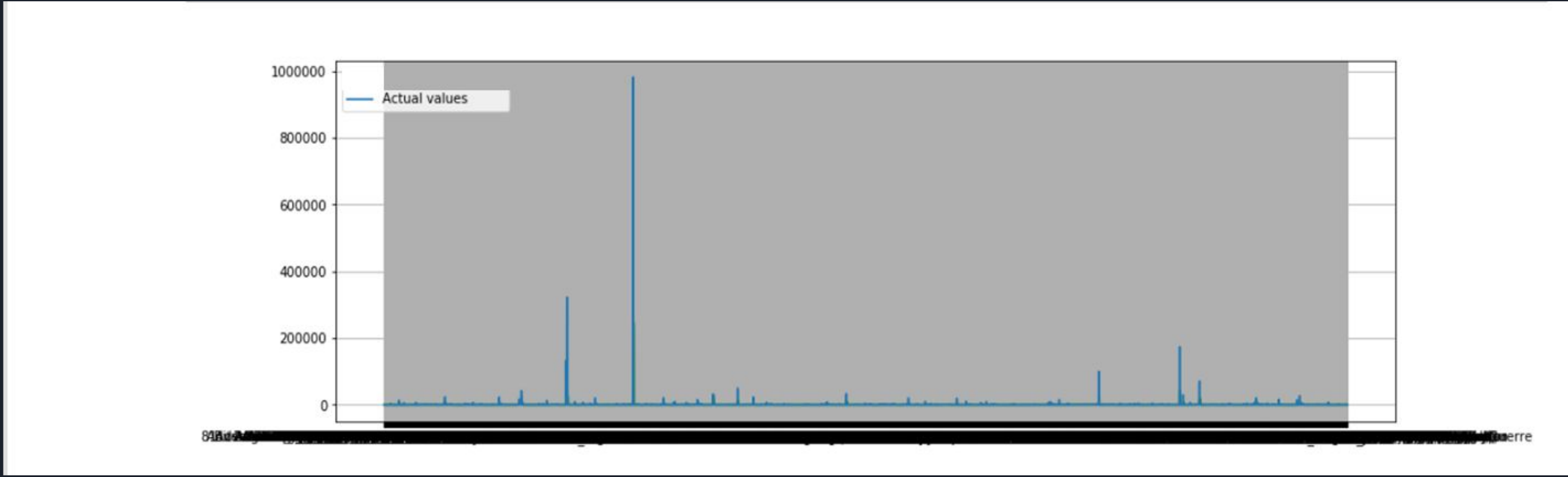
Since the dataset is bound to have some outliers and anomalies we applied anomaly detection method using the local-outlier factor and detecting the anomalies.

We are using The Local Outlier Factor for find out the anomalies in this step.

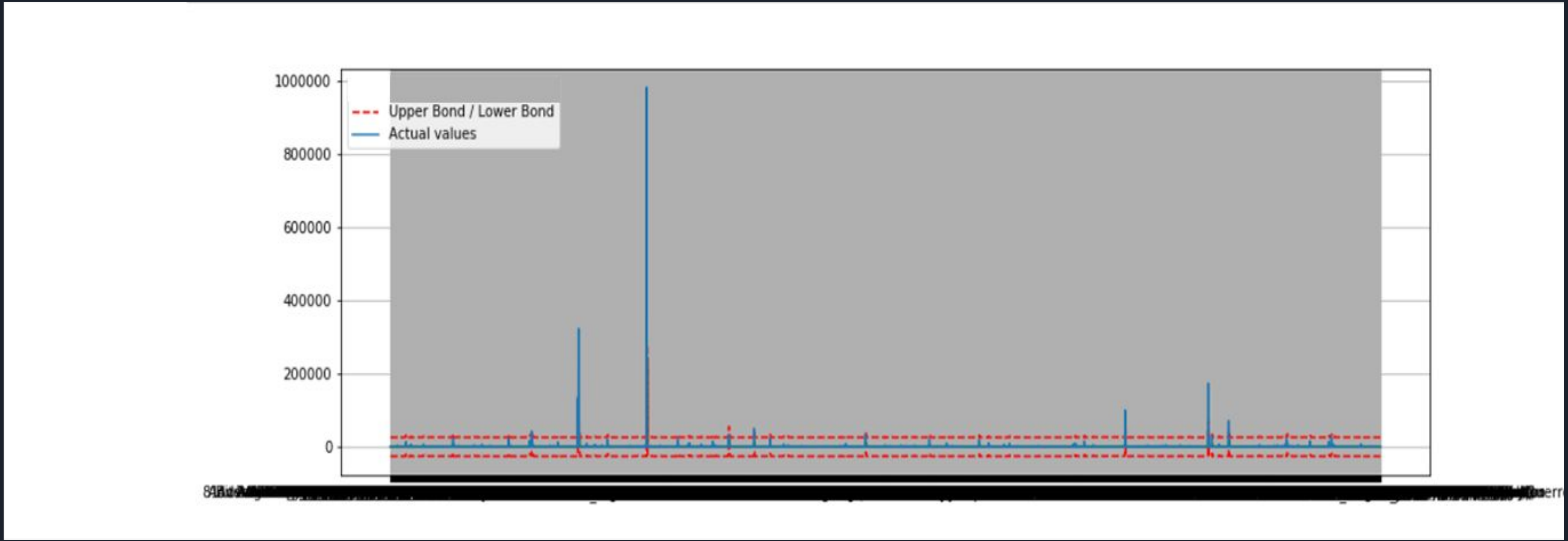
Reference : Check here

Here we are using Hours played as the classifying variable.

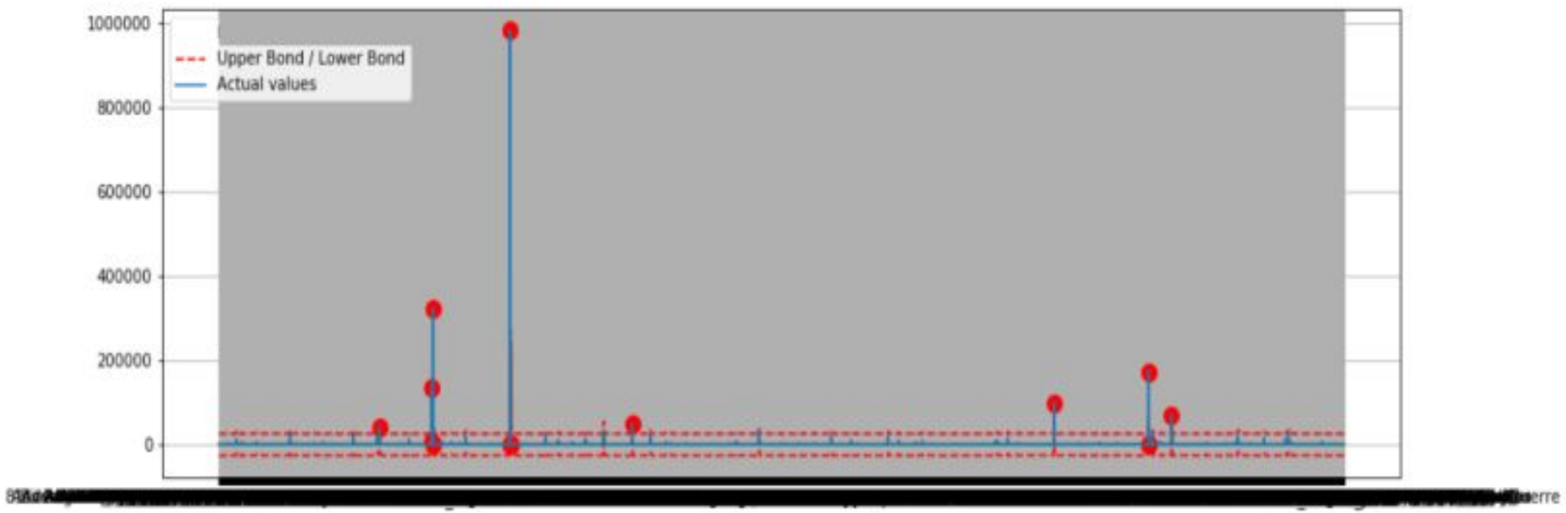
Anomaly detection



Anomaly detection



Anomaly detection



Recommendation engines

- Information filtering system
- User focused based

Recommendation engines-example

Facebook - “People you may know”

Youtube - “Recommended videos”

Netflix - “Other movies you may enjoy”

Pinterest - “Recommend Images”

Linkedin - “Jobs you may be interested in ”

Amazon - “Customers who bought this item also bought ….”

Today’s focus

Collaborative Recommender

- Collaborative Recommender with ALS
- Collaborative Recommender with EM and EVD

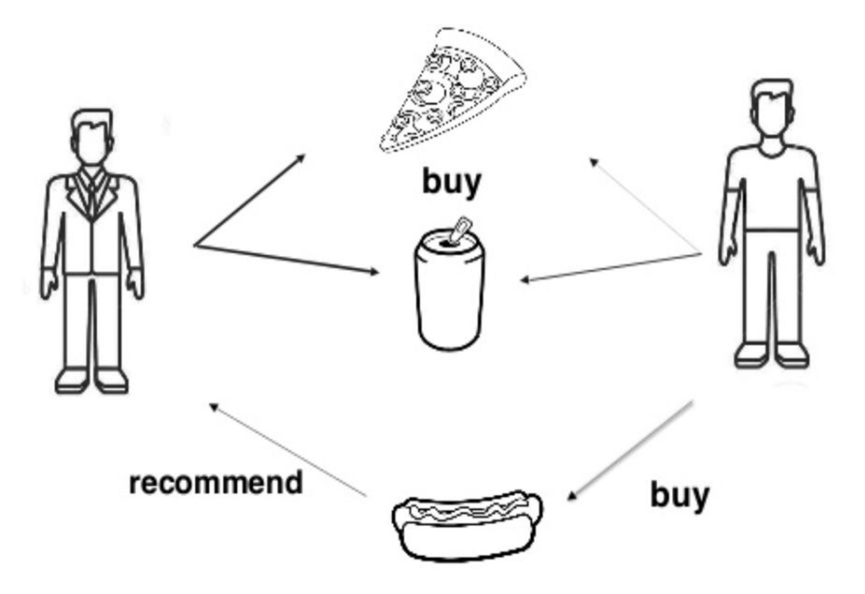
Collaborative Recommendation

Collaborative filtering is a technique that can filter out items that a user might like on the basis of reactions by similar users.

We generally consider three types of scenarios:

- User-User
- Item-Item
- User-Item

How does it work..??



Collaborative filtering does not require information about the items or the users in order to provide the recommendations. We only require user interactions with the items and a method to express these interactions in some kind of rating.

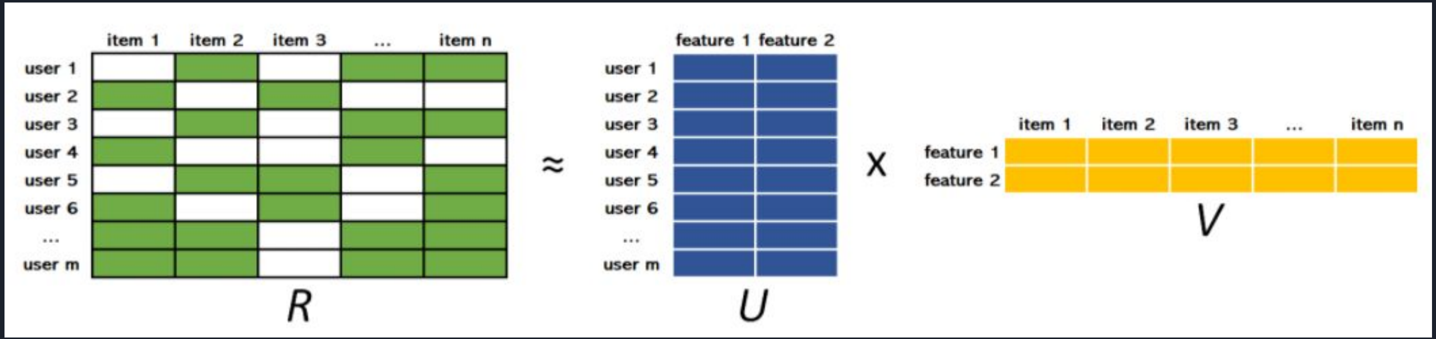
Collaborative Recommender with ALS

The Alternating Least square(ALS) is the model used to fit the data and also generate recommendations.

We have directly used the implementation of the ALS in the implicit library as it is implemented in Cython, making it faster and also allows parallelization of code among the threads as opposed to manual implementation.

ALS uses matrix factorization, here we break “user vs all items ” as “user vs some feature ” and “item vs some feature”.

Collaborative Recommender with ALS



The goal here is to compute the weights of U and V such that **R = UxV.**

ALS alternatively optimises U and V such that the product approximately equals to R.

Collaborative Recommender with EM and SVD

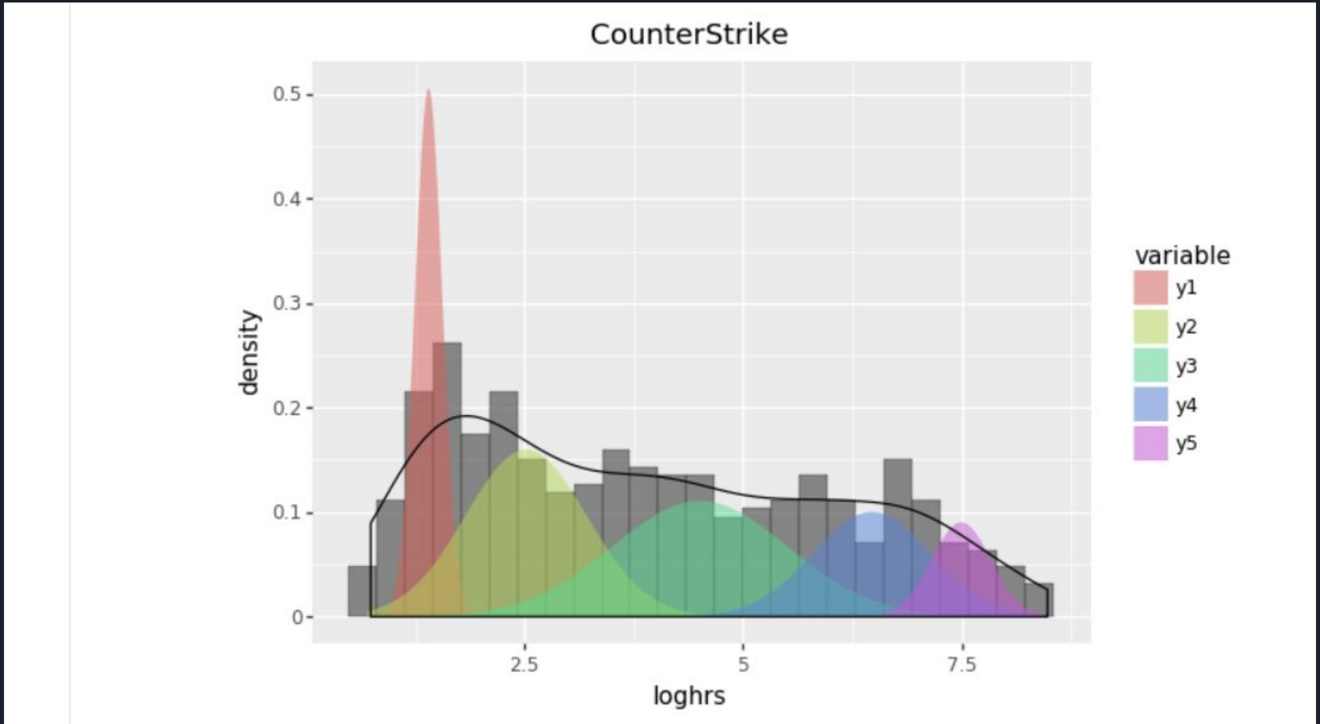
The Expectation-Maximization (EM) algorithm is an approach for maximum likelihood estimation in the presence of latent variables. It is an appropriate approach to use to estimate the parameters of a given data distribution.

In order to come up with a rating system , we decided to use the distributions of hours played for each game with the EM algorithm.

We are using the 5-star rating system.

And we filter-out games which are played less than two hours.

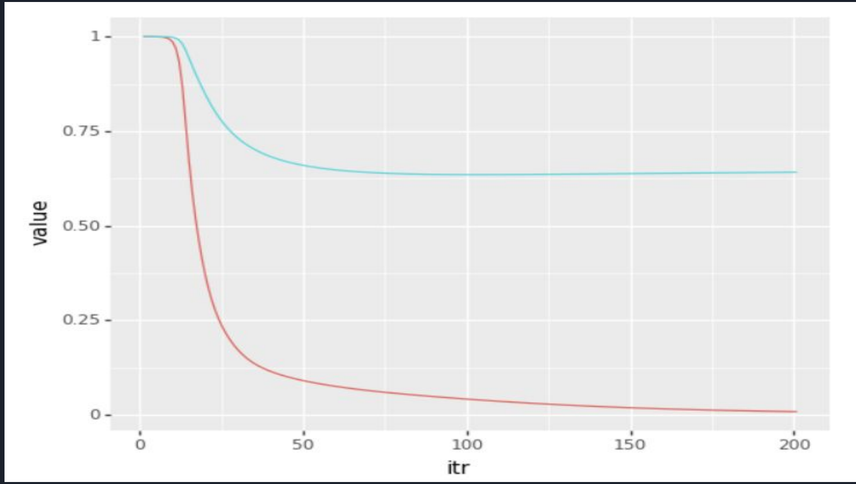
Collaborative Recommender with EM and SVD



Collaborative Recommender with EM and SVD

Now we apply SVD algorithm to factorize the user-item matrix into singular vectors and singular values.

We kept the learning rate at 0.001 and number of iterations as 200 and tracked the RMSE.

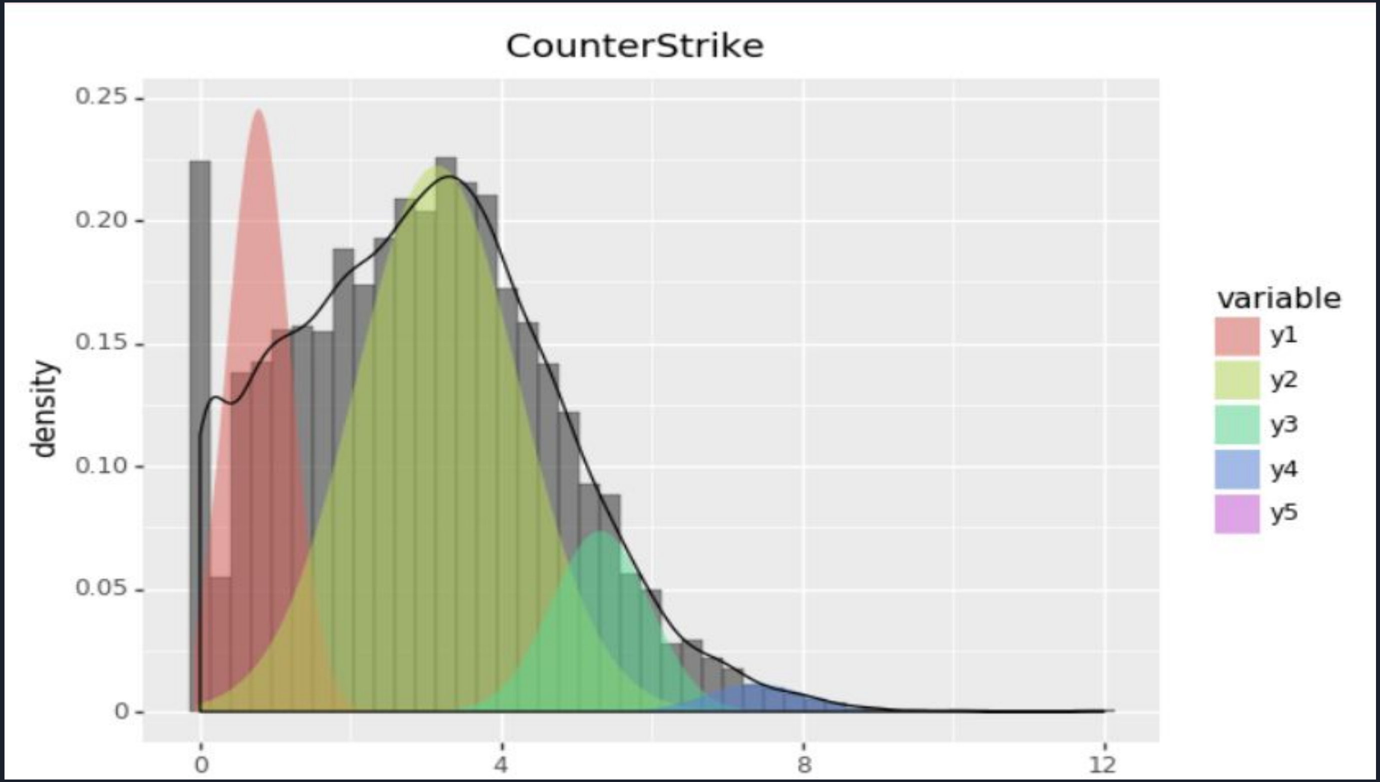


Collaborative Recommender with EM and SVD

Now we just apply EM-algorithm, post SVD .

Keeping the same 5-star rating which was described earlier.

Collaborative Recommender with EM and SVD



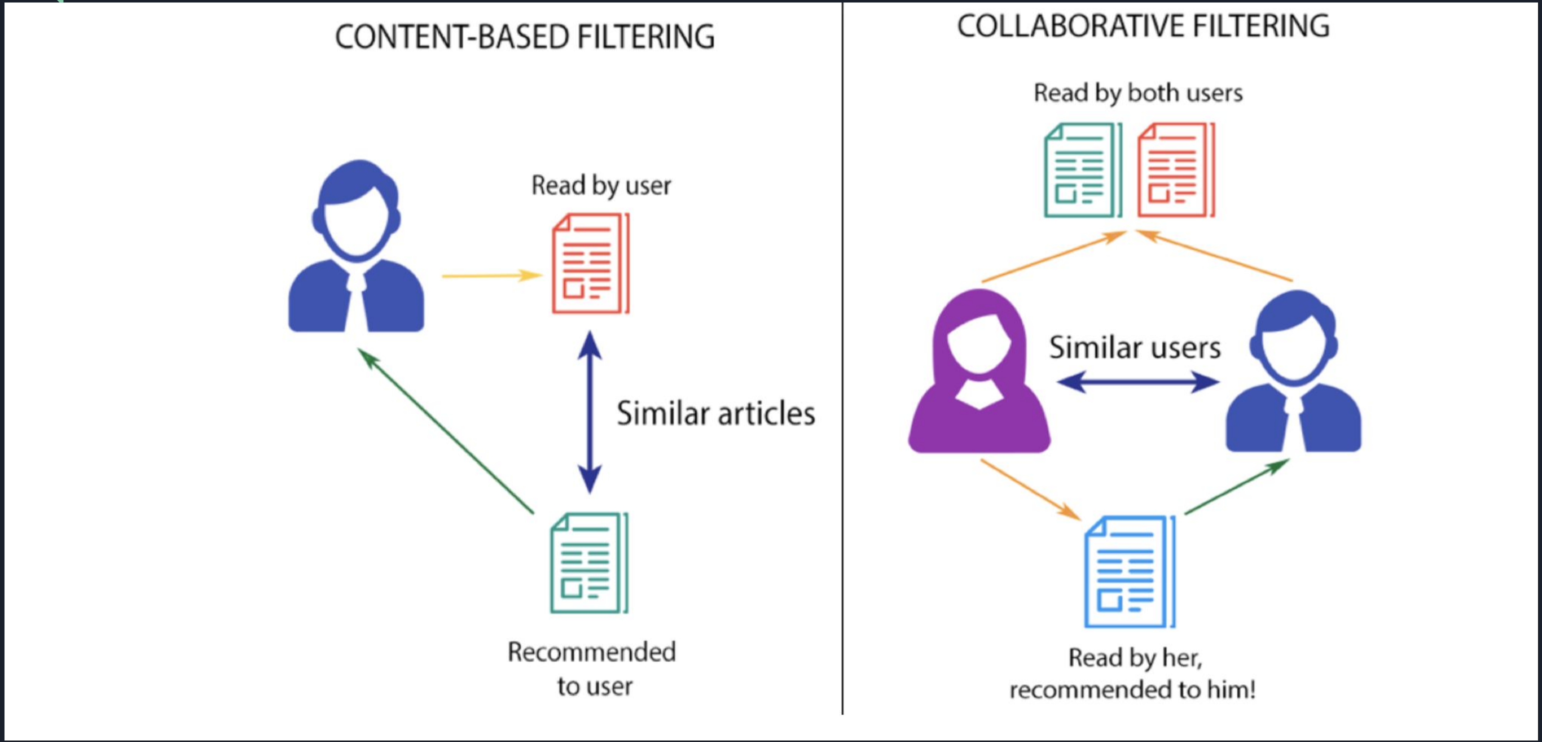
Recommendation

Now we just combine the results produced with ALS and EM-SVD together to get the recommendation.

```
top 20 recommended games for user 5250:
0 ) CitiesSkylines
1 ) FootballManager2015
2 ) GoatSimulator
3 ) Fallout3GameoftheYearEdition
4 ) AgeofEmpiresIIHDEdition
5 ) Terraria
6 ) TeamFortress2
7 ) MedievalIITotalWar
8 ) FootballManager2014
9 ) HalfLife2EpisodeTwo
10 ) StarTrekOnline
11 ) AmnesiaTheDarkDescent
12 ) FootballManager2012
13 ) HalfLife2
14 ) ChivalryMedievalWarfare
15 ) RedFactionGuerrillaSteamEdition
16 ) ScribblenautsUnlimited
17 ) CompanyofHeroes
18 ) Metro2033
19 ) NEOTOKYO
```

Future scope

Collaborative Filtering VS Content Based Filtering



Future scope

- Plans to enhance the predictions and the recommendation system by adding Content Based Filtering or content extraction to the model.
- Aim to implement a hybrid system to get perks of both Collaborative Filtering & Content Based Filtering.

Content Based Filtering

It basically Comprises of:

- Meta-data Extraction
- Clustering based upon the features
- Finding The Similarity/ Distance Between the objects based on the features

Difficulties faced

- Implementation of EM with SVD took some time.
- Recommendation for some of the users was not up to par when users only has very few interactions
- Initial product recommendation was difficult.
- Needed more interactions data like watch time, read time etc.

Conclusion

We made a collaborative recommendation system that works fine for most of the users .

Based on our experience with this project, we understand better how a collaborative filtering system works.

It indeed doesn’t use any information about the items, but relies entirely on the user-items interactions and matrix operations in order to produce recommendations.

We understand better to what point computing time is an important aspect to keep in mind.

Future Work

- In the future, we will dig into content based filtering and its performance on several evaluations.
- We will consider Recommendations both for binary ratings and continuous ratings.
- Optimizations for speeding up the process in real time.
- Try creating a hybrid model just for the learning process.