BUILDING CINEPHILE

an Online Database AI-enabled System

1 System Requirements:

1.1 Primary Groups of Users:

Cinephile, an IMDb new competitor, is entering the digital entertainment market. It is targeting three users' groups: group 1- fans seeking entertainment, group 2- entertainment industry professionals, group 3- developers sending queries to Cinephile APIs in their applications. Requirements for systems design are surely influenced by user needs. For the sake of simplicity and RS-focused approach, I will focus *primarily* on designing a system that fulfil group 1's user need.

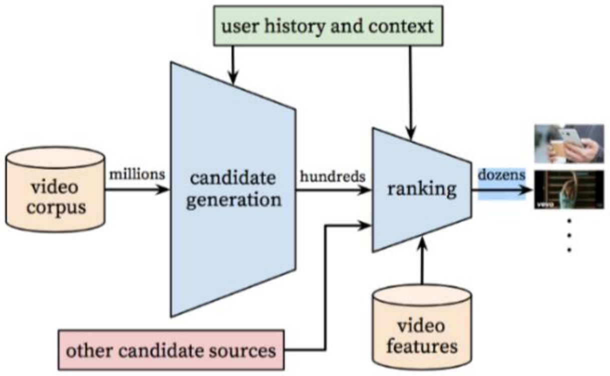
1.2 List of Requirements:

Requirement #	SYS-1	**Applicable to User Group**	**Group 1**	X	**Req. Type**	Functions & Performance
			Group 2	X		
			Group 3	-		
Description	System should recommend its users a selection of contents filtered from database based on their profile and context.					
Sub-system Req.	SUB-1-1	Service performance should generate recommendations within a time frame of 500 milliseconds for each request (max. response time)				
	SUB-1-2	Max. re-train time for the model is two hours.				
	SUB-1-3	Service shall be maintained during re-training of the model.				
	SUB-1-4	System shall produce 25 recommendations when users load the home page				
Requirement #	SYS-2	**Applicable to User Group**	**Group 1**	X	**Req. Type**	Network
			Group 2	X		
			Group 3	-		
Description	System should be capable of processing heavy traffic					
Sub-system Req.	SUB-2-1	Traffic routing to manage resources				
	SUB-2-2	System should be able to handle unpredictable load variations				
	SUB-2-3	System resources should account for 3% annual traffic growth				
Requirement #	SYS-3	**Applicable to User Group**	**Group 1**	X	**Req. Type**	Monitoring
			Group 2	X		
			Group 3	X		
Description	System should maintain its operational metrics in check					
Sub-system Req.	SUB-3-1	System shall track and log user actions, including visit time, clicks and IP address.				
	SUB-3-2	System must continuously analyse data to provide direction for changes in strategy, product, or services				
Requirement #	SYS-4	**Applicable to User Group**	**Group 1**	X	**Req. Type**	Capacity & Network
			Group 2	X		
			Group 3	X		
Description	Services should be able to run across multiple locations (regions or availability zones) at the same time.					
	SUB-4-1	Services shall have a high availability				

Sub-system Req.	SUB-4-2	System should offer flexibility to adopt with changing business conditions (e.g., capability to expand operations to a new region or offer new, compelling features more quickly than competitors)				
	SUB-4-3	Data integrity and consistency are guaranteed for user requests across the locations				
Requirement #	SYS-5	**Applicable to User Group**	**Group 1**	X	**Req. Type**	Reliability & Availability
			Group 2	X		
			Group 3	X		
Description	Systems need to stay up and running, but if they fail, they must recover quickly and gracefully with no human in the loop.					
Sub-system Req.	SUB-5-1	System failures shall be logged and analysed regularly				
Requirement #	SYS-6	**Applicable to User Group**	**Group 1**	X	**Req. Type**	Storage
			Group 2	X		
			Group 3	X		
Description	System shall handle SQL and NoSQL data types.					
Sub-system Req.	SUB-6-1	System shall run a data validation exercise against a data schema				
	SUB-6-2	Database system should handle heavy write and read actions.				
	SUB-6-3	Data must be stored consistently				
	SUB-6-4	Data integrity shall be respected				
	SUB-6-5	Data shall be protected against disasters, data corruptions, data lose.				
	SUB-6-6	Databases servers shall be capable of managing large queries				
Requirement #	SYS-7	**Applicable to User Group**	**Group 1**	X	**Req. Type**	Privacy
			Group 2	X		
			Group 3	X		
Description	The privacy of the users should be guaranteed in the system					
Sub-system Req.	SUB-7-1	Sensitive Personal Identifiable Information (PII) should be identified				
	SUB-7-2	PII should be encrypted				
Requirement #	SYS-8	**Applicable to User Group**	**Group 1**	X	**Req. Type**	Interface
			Group 2	X		
			Group 3	X		
Description	System shall have an intuitive and modern UI/UX					
Sub-system Req.	SUB-8-1	Interface must allow users to search and retrieve information from website content.				
	SUB-8-2	Interface must allow users to rate items (movie and shows) on the scale of 1-5				
	SUB-8-3	Interface must allow users to add items to their watchlist				
	SUB-8-4	Interface must support different devices (e.g., screen sizes)				
	SUB-8-5	Interface shall have a home page that display multiple rows of recommended selections with each row containing 5 items on one row.				

1.3 System Components

Component	Recommender Module	Related Req.	SYS-	1
			SUB-	1-1, 1-2, 1-3, 1-4
Description	Recommender model components provide both personalized and contextualized recommendations then let users make the choice. 1. Personalised RS (AI module): this contains two elements, RS1 and RS2. Regarding RS1, the items embedding data will come through a “funnel” where candidate items are retrieved then ranked before presenting only a few to the user in an architecture similar to YouTube’s			

<table>
<tr><td></td><td colspan="4">RS (Covington, 2016). Taking events from a user's history as input, the candidate generation network significantly decreases the amount of items and makes a group of the most relevant ones to the user from a large database (collaborative filtering). Next is the candidates lists go through a ranking network, which can assign a prediction rating to each item according to a desired objective function that uses items and user behaviour data. Items with the highest rating predictions are presented to the user, ranked by their predicted score.


user history and context
video corpus
millions
candidate generation
hundreds
ranking
dozens
other candidate sources
video features

Figure 1 YouTube recommender system architecture (Covington, 2016)
Also, this module will return provide another list that is titled 'Because you watch this ….'. This list is an output of a content-based recommender, RS2. Personalised RS is located on a web server and run-on large spark clusters.
2. Contextualized RS: provide popular items in geographical and global context.</td></tr>
<tr><td rowspan="2"><b>Component</b></td><td rowspan="2">Monitoring</td><td rowspan="2">Related Req.</td><td>SYS-</td><td>3, 5</td></tr>
<tr><td>SUB-</td><td>3-1, 3-2, 5-1</td></tr>
<tr><td><b>Description</b></td><td colspan="4">A monitor component to recover and analyse user events, but also for general system health monitoring. User logs are used for building the AI module quality metrics but also to derive implicit feedback about user behaviour. This will be become an input to personalised RS. Moreover, logs will be channelled to a monitoring platform where Cinephile staff monitor live KPI. Platform helps staff to organise resources and to plan strategically to mitigate foreseeable operational problems in an autonomous manner.</td></tr>
<tr><td rowspan="2"><b>Component</b></td><td rowspan="2">Database</td><td rowspan="2">Related Req.</td><td>SYS-</td><td>6, 7</td></tr>
<tr><td>SUB-</td><td>6-1, 6-2, 6-3, 6-4, 6-5, 6-6, 7-1, 7-2</td></tr>
<tr><td><b>Description</b></td><td colspan="4">Since a Cinephile's recommendation engine mainly runs on data, data mining and storage are of primary concern. Cinephile have 3 different data layers: user profile data, items data, behaviour data. 1. user profile data may be users' age, sex, location, etc.. 2. item data is metadata information about the items such as titles, genre, actors, release year, category, average rating, etc... 3. behaviour data (historical data) refers to the interaction between users and items for each log on-site activity, e.g., clicks, searches, page, and item views, item rating, items added to watch list, comments and frequency of visits. Contextual information shall also be logged, e.g., device used, current location, referral URL. All PIIs data are anonymized and encrypted.</td></tr>
<tr><td rowspan="2"><b>Component</b></td><td rowspan="2">User Interface</td><td rowspan="2">Related Req.</td><td>SYS-</td><td>8</td></tr>
<tr><td>SUB-</td><td>8-1, 8-2, 8-3, 8-4, 8-5</td></tr>
<tr><td><b>Description</b></td><td colspan="4">UI that allows user to interact with content through a web browser (client).</td></tr>
<tr><td rowspan="2"><b>Component</b></td><td rowspan="2">Load balancer</td><td rowspan="2">Related Req.</td><td>SYS-</td><td>2</td></tr>
<tr><td>SUB-</td><td>2-1, 2-2</td></tr>
<tr><td><b>Description</b></td><td colspan="4">This distributes traffic across a number of servers</td></tr>
<tr><td rowspan="2"><b>Component</b></td><td rowspan="2">NON-COMPONENT
General</td><td rowspan="2">Related Req.</td><td>SYS-</td><td>4, 5, 6</td></tr>
<tr><td>SUB-</td><td>4-1, 4-2, 4-3, 6-2, 6-3, 6-4, 6-5, 2-3</td></tr>
<tr><td><b>Description</b></td><td colspan="4">see sec 4</td></tr>
</table>

2 System Goals

Cinephile seeks to grow their business on a global scale, that is, becoming the largest shows and movies database with a fully global reach. They develop and use their Recommender System (RS) because they believe that it is core to their business for a number of reasons. Their recommender system helps them win moments of truth: when a member starts a session and Cinephile helps them find an engaging material within seconds, augmenting user experience and preventing the abandonment of Cinephile service for an alternative entertainment option. This often immediately converts into corporate value

2.1 Organisational objectives

- Help fans explore the world of movies, shows and artists, - Help fans find the best movie or show to watch next. - Empower fans to share their entertainment knowledge and opinions with the world's largest community of fans. - Go-to source for entertainment

2.2 Leading Indicators

Engagement Metrics: the great thing about digital analytics is that they measure engagement in detail. Such metrices are mostly either time-based or driven by user actions. Time on site, visit length, engaged time, repeat engagement are all indicators of how much customer are using the product. Cinephile is collecting:

- Overall average engagement time: Website users count as "engaged" if they tick one of the following boxes: they lingered on the website longer than ten seconds, viewed more than one page, completed an action. And based on that, its formula would be:

  Overall average engagement time = engaged sessions / total sessions
- User engagement average time**:** this number shows how long each individual lead is sticking around per user via page or screen. Formula is:

  User engagement average time = user engagement durations / number of triggered events.
- New vs. returning visitor traffic: this is expressed by number of registered users and active participants. this indicator portrays the seasonal context. In case of interest in traffic volume, comparison should be made between returning traffic to the same month last year. If people tend to come back to Cinephile website, it means they found that its product is valuable the first time around that gave them a reason to return.

Pleasure metrics (satisfaction and sentiment metrics, among others): These metrics give the ability to control and increase customer loyalty as it measures how customer feel about the product. When acted upon, they can be employed to build brand awareness and to create positive company image.

- Customer Satisfaction Score (CSAT) is a survey-based metric that measure whether and to what extent product or service meets/exceeds customer expectations. Formula is:

  CSAT = number of satisfied customers / numbers of all customers in analysed period
- Customer Effort Score (CES) is one of the key performance indicators of a successful customer-brand relationship is the effort needed to finalize transactions or understand the instructions provided by the customer service rep. CES is collected through a simple questionnaire.
- Net Promoter Score comes down to checking how likely a customer would recommend the product or service to family and friends. In other words, customer loyalty is measured that way. Respondents answer the NPS question on an eleven-point scale (0-10). Depending on their score, they go to the group of detractors (0-6), neutrals (7-8), or promoters (9-10). NPS score is a difference between the percentage of promoters and critics.

2.3 User Outcomes

Users' outcomes quantify the interaction between group 1 users and the live product. Metrics takes their input from user logs (behaviour data).

- The click-through rate (CTR) is a metric that measures how many people click on the recommendations. The basic notion is that if more people click on the recommended items, the recommendations are more relevant to them.

- Adoption and Conversion: while the CTR can measure user attention or interest, it can't tell whether users liked the recommended news article they clicked. Cinephile can tally user's clicks on suggestions only if they spend above certain amount of time browsing or had completed an action (e.g., signing up, sharing a page, writing comments, see Figure 5). This is called the completion rate.
- Novelty: A novel recommendation is one that the users did not know about. Measuring novelty is to count the number of popular items that have been recommended. This metric is based on the assumption that highly rated and popular items are likely to be known to users and therefore not novel. Another good measure for novelty might be to look more generally at how well a RS made the user aware of previously unknown items that subsequently turn out to be useful in context.
- Serendipity: similar to novelty, but concerns only with unexpected recommendation that prove to be relevant to customer (correctness). This is a user-centric metric that focus on improving user utility.

On the other hand, group 2 users' outcome will include the total revenue generated from paid premium accounts and promotional contents. Seasonal analysis of API requests is an indicator of group 3 outcome.

2.4 Model Properties

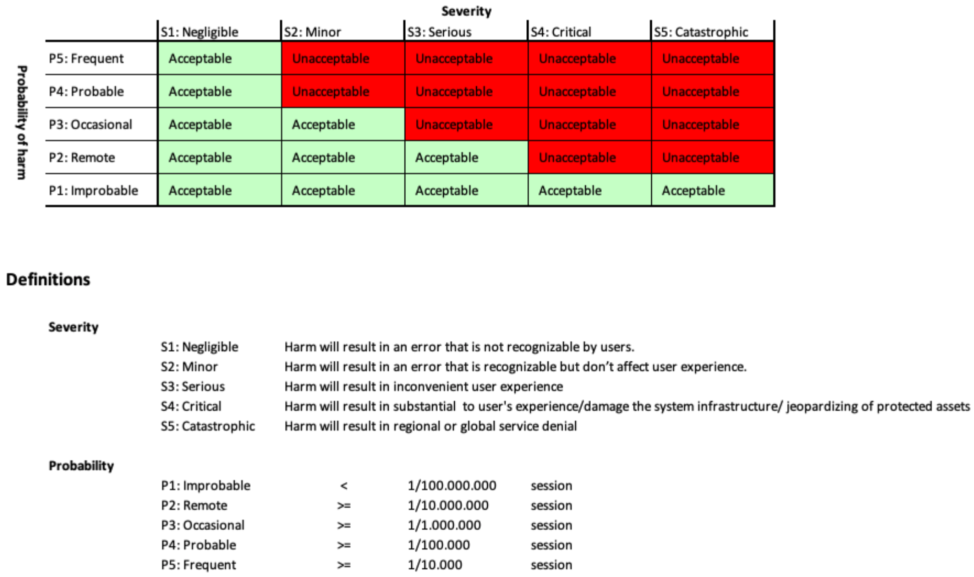
A recommender system impacts users, e.g., in terms of their decision making and choice processes. Among metrics used to measure model properties are:

- Rating prediction metrics: RMSE and MEA. These are measured by comparing the predicted rating with user explicit or implicit ratings.
- Relevance metrics: using classification metrics i.e. precision and recall. It is useful to monitor whether RS is providing items that users may like to use.
- Ranking metrics: there are a number of useful metrics, e.g., Mean Reciprocal Rank (MRR) and Discounted Cumulative Gain (DSG) that can be used to determine how relevant the rank, or ordering of items in the list.
- Catalogue coverage: this gives the insight into how many of the items in collection are being recommended to all users who get recommendations.

The quality metrics of the RS is intricately linked with the overall revenue model of the Cinephile site (Jannach and Adomavicius, 2017). Dias et al. (2008) emphasises the value of RS to e-Businesses and in a study, they conducted showed the extra revenue generated when deploying RS. A number of real-world tests of recommender systems have found that having a RS increases user activity, (Freyne, Jacovi, Guy and Geyer, 2009). When RS is successfully implemented and continuously monitored, model properties will be maintained and enhanced. A well-functioning RS concurs with an increase of adoption and conversion. This would enhance the user experience and lead to an increase in customer engagement and uplift in sentiment. Higher levels of user engagement are thought to contribute to increased levels of user retention often immediately converts into business value. Generally, a successful implementation of the RS or changes to running system that enhance model metrics would ripple up and have a positive effect on higher-level organisational goals.

3 Risks

RS's risk management starts proactively during product development. It concerns with identifying risk potentials. Figure 2 describes Cinephile's risk acceptability matrix, where ascending levels of probability and severity are defined and product specific. Severity can be estimated qualitatively before system deployment then monitored and measured quantitively using telemetry data. Goals of Cinephile project team is to mitigate all risks by one or combination of the following techniques: - avoid the risk by categorically removing the hazard this is regularly done through design. - reduce the severity of the harm or the probability of the hazard occurrence - accept the residual risk. These mitigations apply to all risk items even if they are in the acceptable area. However, all risks that have unacceptable levels MUST be brought back to acceptable territory. Risk mitigation techniques may introduce new entries to the requirements repository.



	Severity				
Probability of harm	S1: Negligible	S2: Minor	S3: Serious	S4: Critical	S5: Catastrophic
P5: Frequent	Acceptable	Unacceptable	Unacceptable	Unacceptable	Unacceptable
P4: Probable	Acceptable	Unacceptable	Unacceptable	Unacceptable	Unacceptable
P3: Occasional	Acceptable	Acceptable	Unacceptable	Unacceptable	Unacceptable
P2: Remote	Acceptable	Acceptable	Acceptable	Unacceptable	Unacceptable
P1: Improbable	Acceptable	Acceptable	Acceptable	Acceptable	Acceptable

Definitions

Severity

S1: Negligible	Harm will result in an error that is not recognizable by users.
S2: Minor	Harm will result in an error that is recognizable but don't affect user experience.
S3: Serious	Harm will result in inconvenient user experience
S4: Critical	Harm will result in substantial to user's experience/damage the system infrastructure/ jeopardizing of protected assets
S5: Catastrophic	Harm will result in regional or global service denial

Probability

P1: Improbable	<	1/100.000.000	session
P2: Remote	>=	1/10.000.000	session
P3: Occasional	>=	1/1.000.000	session
P4: Probable	>=	1/100.000	session
P5: Frequent	>=	1/10.000	session

Figure 2 Risk Acceptability Matrix

Mitigation techniques can also be costly (e.g., redundancy/ backup hardware, using a more secure and expensive servers, backup data in different locations, etc..) but they can be justified if their absence would lead to unfulfilled system goals (cost/benefit ratio). The effectivity of the control measures is verified and checked that it is not introducing new risks/error through regression testing.

Risks management continues after system deployment and concerns with monitoring the success of the mitigation for online systems and detection of new risks.

Table 1 lists few risk items and analyse them then propose mitigation measures and evaluate the update of severity and probability accordingly. One risk is related to the event of recommending users a list of items that they are not interested in (false positive). This could be a result of several hazards (cold start, data distribution shift, i.e. covariate shift, concept shift). Disliked recommended items have a negative effect on the user experience (harm). From a wider perspective, scholars have studied how a bad recommendation can even hurt user trust (Chau, et al., 2013). Psychological studies have described a negativity bias in human perception, whereby bad impressions may sometimes outweigh good ones in our overall assessment of an experience (Baumeister, Bratslavsky, Finkenauer and Vohs, 2001) and (Yin, Bond, and Zhang, 2010).

4 Deployment Strategies

Deployment starts already at design phase. Software components of the systems are designed and fit into microservices, which when combined provide all of the functions required to run the overall system service in Cinephile. Microservices provide system with: - agility to scale up on part of the system where loads are high, -resilience against a failure in one microservice (increased availability), - easy-to-update and maintainability (update and redeploy microservices individually). These microservices will be put into containers then orchestrated with tools like Docker. On a separate, but related note, system architecture needs to be flexible to support the need of AI module to be plugged into or unplugged from the whole system in order to take the benefit of latest progress in algorithms. Such modularity concept will be generally followed in order to facilitate the update of the system to improve technology and fulfil new goals.

To make deployment process as reliable and efficient as possible, Cinephile will create an automated workflows for deployment while monitoring and maintaining the entire pipeline and reporting back quality metrics and user outcomes , with support for delayed ground truth. Deployment will have a version-control on every release, including training data, with a capability of automated rollback between versions. Besides unit and integration testing, the ML pipeline in AI module ought to have data- and model validation tests during CI. Moreover, as model will be retrained with fresh data, CD has to deliver and to deploy an entire pipeline in order to automatically retrain the model. Lastly, data drift as well as training-serving skew must be monitored as it will lead eventually to a degradation in model performance. For testing during operation, Cinephile will test a hypothesis using variant implementations using A/B testing then use canary releases to

Risk Analysis								Risk Control		Risk Evaluation after Risk Control		
ID.	*Hazard*	*Foreseeable sequence of events*	*Hazardous situation*	*Harm*	*Severity*	*Probability of Harm*	*Acceptability*	*Risk control measures*	*Verification*	*Severity*	*Probability of Harm*	*Acceptability*
1	User cold start	1. RS made incorrect prediction (false positive) 2. User adopts system prediction 3. User watch recommended content	User dislike recommended content	• User mistrust in service	Serious	Occasional	Unacceptable	• Upon account creation, ask users a series of questions that could increase system familiarity with user preferences. • Leveraging data already available e.g. in user social media accounts	A/B test	Minor	Improbable	Acceptable
2	Data drift	1. RS made incorrect prediction (false positive) 2. User adopts system prediction 1. User watch recommended content	User dislike recommended content	• Disengaged customer • Dissatisfied customer • User mistrust in service • Lost customer	Serious	Probable	Unacceptable	• Continuous monitoring to detect data drift • Collect fresh data along with their labels and periodically re-train the model	A/B test	Serious	Remote	Acceptable
3	High loads on network	1. Users log in to their session 2. web page takes long time to load in its entirety *or* no recommendation loads.	User gets impatient and disappointed	• Dissatisfied customer • Lost customer	Serious	Remote	Acceptable	• Servers with elastic scaling of resources especially in peak hours • Servers located in user geographic vicinity. e.g., cloud based elastic CDN could be the exact solution to use.	Blue/green	Serious	Improbable	Acceptable
4	A system vulnerability	1. Bad actors find a system vulnerability 2. Bad actors exploit the system vulnerabilities	Bad actors intrude into systems and gain unauthorized access through vulnerability exploitation.	User data leaks in the absence of explicit consent	Catastrophic	Probable	Unacceptable	• Architecture: Tighten network security • Algorithmic: anonymization, encryption • Conduct a simulated cyber-attack to proactively identifies vulnerabilities • Follow continuously the most recent attack to similar website and close any similar security gaps.	Simulated attacks	Serious	Remote	Acceptable

Table 1 RS Risk Assessment

detect problems and regressions. Deployment includes configuring and managing of control plane, nodes, and instances (part of cost of supervising the system's work). It will show compliance with regulatory requirements of every regulatory agency where it provides service.

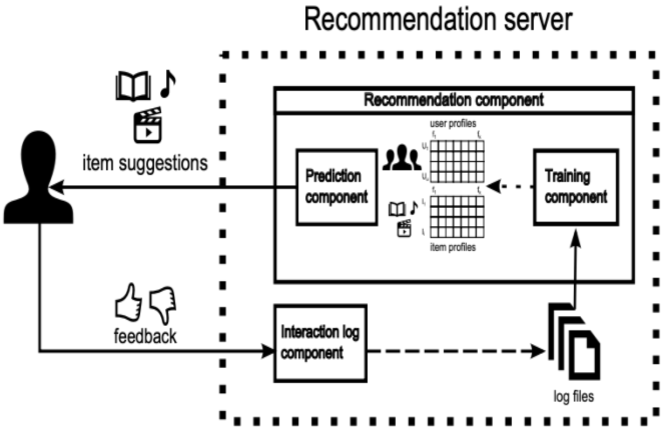


Figure 3 Traditional Recommender System (Moreno, Castro and Riveill, 2014)

CINEPHILE, as for most of the other recommender systems found on the web, are server-based and centralized. It gathers and stores as much information as possible about their users and items in the servers. This is supported by the current availability of inexpensive storage and the scale of computational power available. CINEPHILE benefits from servers' performance to run computationally intensive algorithms to retrain models on the server that scale up to the size of the collected data (batch processing) and use the trained and deployed models to adequately answer to a large number of user requests.
When user logs into their session, a request with contextual information arrives to the servers' side where user embedding (extracted from server-side user log) gets prepared and inputted into a pretrained AI component. The component generates the ranked recommendation lists (25 items) and return them to users. Users interact with the system and new events are logged into user log and used for next sessions.

Without the bustle of infrastructure and in-house server maintenance, Cinephile can deploy its services using pay-for-use cloud solutions within minutes rather than months. This reduces operational complexity and cost when compared to physical servers. Cloud solutions promises auto scalability. It is only required to calculate how many servers the website requires and request them from any Platform as a Service (PaaS) provider. It guarantees both structural scalability and load scalability though the use of efficient scheduling algorithms and parallelism. PaaS provider will be responsible for ensuring high availability and fault tolerance of their offerings. This reduces engineering complexity and time spent. One more advantage of this deployment strategy is to shorten time-to-market which permits development team to focus on essential value-stream boosting activities, and also, facilitate CI/CD. Cloud provider will also take care of sharding, replicating the database in order to ensure data integrity, availability, and consistency when data get loaded by clients globally. Privacy of users' data is well protected when stored in the cloud, but recently, this aspect continues to receive increasing public pressure to heighten the protection level. Although clouds providers claim to provide high level of privacy, scandals keep coming out to public indicating breaches that put millions of records in jeopardy.

Moreno, Castro and Riveill, (2014) propose an alternative architecture. It mainly consists of moving user data gathering, processing and storage to client-side. Although the decentralized model presented here doesn't share the whole profile of the user with the server, the proposed system still lets know the recommendation server which items the user has interacted. Anonymization networks could be used to hide the identity of the user, with the cost of having a negative impact in the scalability of the system. Then

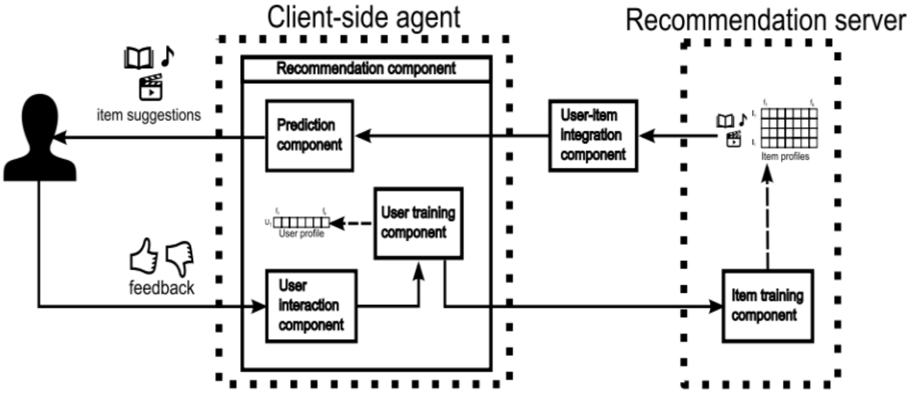


Figure 4 Proposed Architecture by (Moreno, Castro and Riveill, 2014)

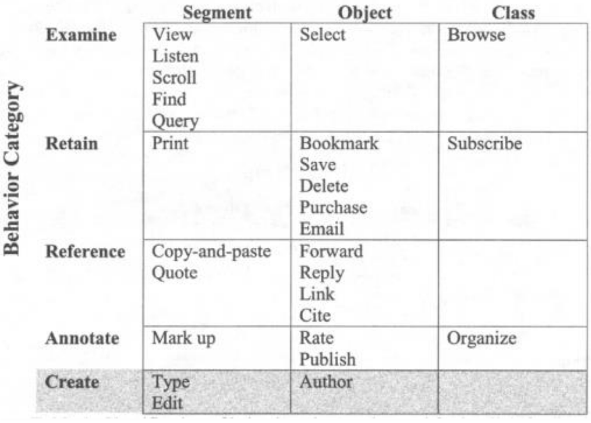
this alternative does not suit Cinephile needs. *(Newell, 2013)* designed a client-side recommender system where all tasks except model training are implemented on the client system. Since both of these components are static files, which can be cached, the approach is easily scalable to a large number of users. The recommender system is also easier to manage and deploy as the complexity and processing requirements of the server are reduced. The responsiveness becomes independent of the load on the server and the round-trip delay to the server is eliminated. However, the client-side approach is only practical when there is a limited number of recommendable items since the model size generally scales with the number of items. hence this approach is not possible for large database like CINEPHILE's.

5 Data

According to (Huyen, 2022), failure of ML systems originates either from Software system failures (dependency failures, deployment failure, hardware failures, downtime or crashing, issues related to distributed systems) or from ML-specific failures (data collection and processing problems, poor hyperparameters, changes in the training pipeline not correctly replicated in the inference pipeline and vice versa, data distribution shifts that cause a model's performance to deteriorate over time, edge cases, and popularity bias.). Cinephile shall maintain the health of the AI module by monitoring of model's *accuracy-related metrics, predictions, features, and raw inputs*. While it is hard to monitor the upstream part of the pipeline, downstream process steps are easier to monitor and closer to user outcomes and business values. Firstly, I describe data collection then briefly go through monitoring of different stages.

5.1 Data collection:

The behaviour data contains data that portraits explicitly and implicitly users feedback about the product. Explicit data is information that is provided intentionally, i.e., input from the users such as movie ratings, comments, search for an item, rank a collection of items from favorite to least favorite, choose the better prediction, create a list of items that the user likes. Implicit data is information that is not provided intentionally but gathered from available data streams like search history, clicks, watch list history, user viewing time, user's social network or search activity, etc, see Figure 5. This data is aggregated and used for feature engineering and for system monitoring , e.g., implicit ratings derived from user behaviour (e.g., sentiment analysis of comments). For example, when the user accesses to news or read a movie plot (click through rate), according to the time it takes for reading (completion rate), the system could automatically infer whether the content is on their interest. Even if the feedback can't be used to infer user preferences directly, it can be used to detect changes in ML model's performance. If over time, the click through rate remains the same but the completion rate drops, it might mean that AI module performance is getting worse. These techniques take advantage of user behaviour to understand user interests and preferences.



Behavior Category	Segment	Object	Class
Examine	View Listen Scroll Find Query	Select	Browse
Retain	Print	Bookmark Save Delete Purchase Email	Subscribe
Reference	Copy-and-paste Quote	Forward Reply Link Cite	
Annotate	Mark up	Rate Publish	Organize
Create	Type Edit	Author	

Figure 5 Behaviours classification used for implicit feedback (Kelly, 2003)

A recommendation is presumed to be bad if there's a lack of positive feedback. After a certain time window or certain number of sessions logins, if there is no event (examine, retain, or reference items), the item is presumed to be irrelevant (false positive). Choosing the right window length requires thorough consideration, as it involves trade-off between speed and accuracy. Similar assumptions needed to flag false negative cases when users examine (> 3mins), retain or reference items that is not part of user recommendation pool then user is assumed to prefer these items, hence, these items should be tagged as a false negative and used to improve future prediction but also to improve the classification component. Processing of logs can happen in batch processes. In this scenario, system collects a large amount of logs, then periodically query over them looking for specific events using

SQL or process them using a batch process like in a Spark or Hadoop. This makes the processing of logs efficient because system can leverage distributed and mapreduce processes. Hadoop uses HDFS to split files into large blocks and distributes them across nodes in a cluster, which means the dataset will be processed faster and more efficiently. The total volume of stored telemetry data for all users can be roughly estimated. I assume that: number of system users is 2 million, number of items is billion items and log memory size per item per user is 100 bytes. Also, I assume that user will interact on average with 0.01% of the database on their lifetime, which is 100 thousand items. Then, estimated required storage will be 100 byte * 100 thousand item * 2 million user = 20 TB. Transmitted telemetry data is small per user, but at a specific point in time, if we assume that a system would have 200 thousand users concurrently online at peak hours, then at that point, accumulated volume of transmitted data will be 200 thousand * 100 bytes = 20 Mb.

5.2 Monitoring:

There is one important assumption underlying all personalized recommender system, which is this: users who have similar preferences in the past are likely to have similar preferences in the future. But what if the distribution of inputs or the relations between independent variables and targets have changed? Monitoring system needs to be in place to ensure that system operates efficiently and fulfil its business goals. To monitor the AI module, 4 areas are under the spotlight: 1- *accuracy-related metrics:* system use combination of explicit feedback and the translating of explicit feedback into user preferences to calculate AI model’s accuracy. This will help monitor model’s performance to determine what's relevant or irrelevant for users to see. 2- *Monitoring predictions:* rating prediction is a regression task which makes it easy to visualize, and its summary statistics are straightforward to compute and interpret. System shall monitor for shift in prediction distribution (a signal to an underlying change in input distribution). 3- *Monitoring features*: system needs to track changes in features, both the features that a model uses as inputs and the intermediate transformations from raw inputs into final features. This activity consists of comparing these features against a predefined schema, i.e., feature validation. If schema is violated, a shift in distribution might be suspected. Subsequently, a A/B test (and t-test) may be conducted to examine the alternative hypothesis. 4- *raw inputs*: like monitoring activities on features, inputs are controlled against a schema. Table 2 gives a detailed description of the quality metrics, data necessary to calculate them and assumptions needs to get these data if they weren't explicit or output of the system.

Recommendation systems, like all ML models, tend to degrade in performance over time – often failing silently. ML observability is the practice of obtaining a deep understanding into model’s performance across all stages of the model development cycle: as it is being built, once it is deployed, and long into its life in production.

6 Bibliography

Baumeister, R.F., Bratslavsky, E., Finkenauer, C. and Vohs, K.D., 2001. Bad is stronger than good. *Review of general psychology, 5(4)*, pp.323-370.

Chau, P.Y., Ho, S.Y., Ho, K.K. and Yao, Y., 2013. Examining the effects of malfunctioning personalized services on online users' distrust and behaviors. (56, pp.180-191.).

Covington, P., Adams, J. and Sargin, E., 2016. Deep neural networks for youtube recommendations. *10th ACM conference on recommender systems*, (pp. 191-198).

Dias, M.B., Locher, D., Li, M., El-Deredy, W. and Lisboa, P.J., 2008. The value of personalised recommender systems to e-business: a case study. *2008 ACM conference on Recommender systems* , (pp. 291-294).

Freyne, J., Jacovi, M., Guy, I. and Geyer, W., 2009. Increasing engagement through early recommender intervention. *Third ACM conference on Recommender systems*, (pp. 85-92).

Huyen, C., 2022, Feb 7. *Data Distribution Shifts and Monitoring* . Retrieved June 23, 2022, from https://huyenchip.com/2022/02/07/data-distribution-shifts-and-monitoring.html

Jannach, D. and Adomavicius, G., 2017. Price and profit awareness in recommender systems. *arXiv preprint*, arXiv:1707.08029.

Kelly, D. and Teevan, J., 2003. Implicit feedback for inferring user preference: a bibliography. *In Acm Sigir Forum (Vol. 37, No. 2)* (pp. 18-28). New York, NY, USA: ACM.

Moreno, A., Castro, H. and Riveill, M., 2014. Client-side hybrid rating prediction for recommendation. *International Conference on User Modeling, Adaptation, and Personalization* (pp. 369-380). Cham: Springer.

Newell, C. and Miller, L., 2013. Design and evaluation of a client-side recommender system. *7th ACM conference on Recommender Systems*, (pp. 473-474).

Yin, D., Bond, S. and Zhang, H., 2010. Are bad reviews always stronger than good? Asymmetric negativity bias in the formation of online consumer trust. *ICIS 2010 Proceedings. 193.*

Perspectives	**Indicators**	**Evaluation Content (Reason)**	**Formula**	**Telemetry data**	**Assumptions**
Machine learning	RMSE	Prediction Accuracy (Measure error of prediction)	$RMSE = \sqrt{\frac{1}{\lvert Q \rvert} \sum_{(u,i) \in Q} (r_{ui} - \hat{r}_{ui})^2}$	$\hat{r}_{ui}$ Predicated rating: given by AI module for each recommended item. **r_{ui}:** true rating **Q:** number of sampled queries	True rating is collected either: * *Explicitly:* in case user rate items * *Implicitly:* in case user don't give an explicit rating, system could give an implicit rating that follow rating vs. visit length distribution. e.g. score of 4 if customer stayed over 6 mins in rated page.
	MEA		$MAE = \frac{1}{\lvert Q \rvert} \sum_{(u,i) \in Q} \lvert r_{ui} - \hat{r}_{ui} \rvert$		
Information retrieval	Precision	Precision of recommendation (describes the proportion of items that users prefer, and the recall describes the proportion of the user favorite items that are not missed.)	$Precision = \frac{N_{rs}}{N_s}$	**N_{rs}**: Number of recommended items that users prefer. **N_s**: Number of the recommended items **N_r**: Number of items that user prefers	Items are considered to be preferred in case: * User has completed an explicit action to the item * User has completed an implicit action as defined in Figure 5 with a defined threshold (e.g. visiting time of > 2 min to an item page is an indicator that user prefers this item)
	Recall		$Recall = \frac{N_{rs}}{N_r}$		
	MRR	Ranking precision of recommended items	$\mathrm{MRR} = \frac{1}{\lvert Q \rvert} \sum_{i=1}^{\lvert Q \rvert} \frac{1}{\mathrm{rank}_i}$	**Q**: number of sampled queries. rank_i: refers to the rank position of the first relevant document for the i-th query.	relevant items are items of the recommendation list where: * User has completed an explicit action with a defined threshold. e.g. any true rating above 3.5 corresponds to a relevant item and any true rating below 3.5 is irrelevant. * User has completed an implicit action as defined in Figure 5 with a defined threshold (e.g. visiting time of > 2 min to an item page is an indicator that user finds this item relevant)

	nDCG		$\mathrm{nDCG_p} = \frac{DCG_p}{IDCG_p}$ $\mathrm{IDCG_p} = \sum_{i=1}^{\lvert REL_p \rvert} \frac{rel_i}{\log_2(i+1)}$ $\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}$	rel_i relevancy score, scale: 0 least relevant, 1 somewhat relevant, 2 most relevant	relevancy score will be given to items in recommendation list where: * User has completed an explicit action with a threshold. e.g. any true rating above 4.5 corresponds to a 2 in relevancy score, 3-4.4 rating to 1 relevancy score and below rating receive a 0 in relevancy score. * User has completed an implicit action as defined in Figure 5 with a defined threshold (e.g. visiting time of > 2 min to an item page corresponds to a 2 in relevancy score)
	Coverage	Catalogue coverage	$coverage = \frac{\bigcup_{u \in U} I(u)}{I}$	$I(u)$: is the number of items recommended for a user I: represents the total number of items	-
Business metrics		Click-Through Rates	CTR = $\frac{Nc}{Nr}$	Nc: number of clicks Nr: number of recommendations	-

Table 2 Quality Metrics and Data to-be collected