Recommender System

Prepared by:
Sumaira Javed Malik
(DataScientist Analyst)

Outline

- TapmadTv & the Recommendation Problem
- Anatomy of TapmadTv Personalization
- Data & Models
- And…

a) Consumer (Data) Science
b) Or Software Architectures

Motivation: why care about recommender systems?

Because money

For companies like Amazon, Netflix, and Spotify, recommender systems drive significant engagement and revenue

Because value

Recommender systems provide a scalable way of personalising content for users in scenarios with many items

Defining Recommender Systems

- A recommender system is a subclass of information filtering system that seeks to predict the "rating" or "preference" that a user would give to an item.

- Recommender systems have become increasingly popular in recent years, and are utilized in a variety of areas including movies, music, news, books, research articles, search queries, social tags, and products in general.

Examples

NETFLIX

Video-on-demand provider in North America and UK

- Matches 23 million customers with a huge inventory of movies according to their tastes
- 60 -70% of views result from the recommendations[9]

amazon.com

Gold standard of e-commerce. Pioneer in using recommendations

- Sits on a huge volume of collective information of its customers
- Customers can view what people with similar tastes viewed or purchased
- Customers can ask the recommendations engine to ignore selected purchases

LinkedIn
facebook

Social and professional networking sites

- Sits on a huge volume of collective information of its customers
- Customers can view what people with similar tastes viewed or purchased
- Customers can ask the recommendations engine to ignore selected purchases

PANDORA

Music station. Offers music suggestions based on ratings

- Sits on a huge volume of collective information of its customers
- Customers can view what people with similar tastes viewed or purchased
- Customers can ask the recommendations engine to ignore selected subscriptions[3]

TapmadTv & the Recommendation Problem

Our project is about creating a recommendation engine which will recommend movies and TV shows that would have a high chance of being enjoyed by the users.

Requirements

- ✓ some information about the available items such as the genre ("content")
- ✓ some sort of user profile describing what the user likes (the preferences)

Cont…

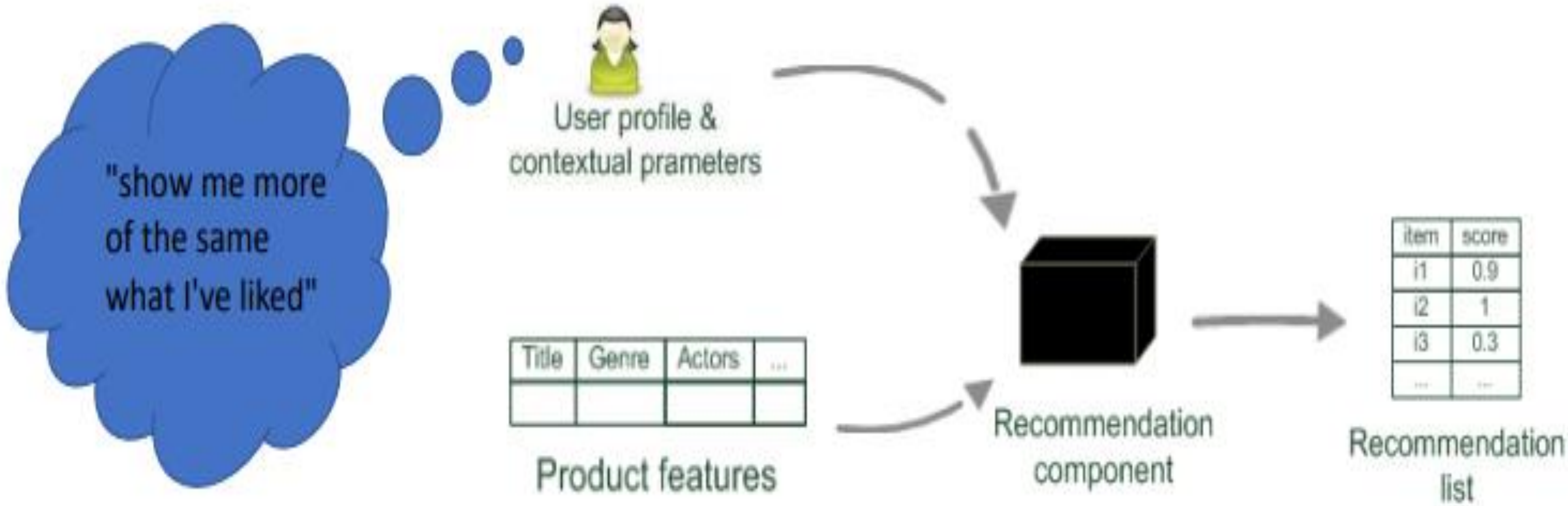
“Similarity” is computed from item attributes, e.g.,

- User profiling like user_id, gender, age, zip
- Similarity of movies by actors, director, genre
- Ratings in which user_id, movie_id, rating, timestamp
- Item files movie id, tittle, release date, video release date, IMDB URL, unknown, Action, Adventure, Animation, Comedy, Crime, Documentary, Dramma, Fantasy, Horror, Musical, Mystery, Thriller, War.

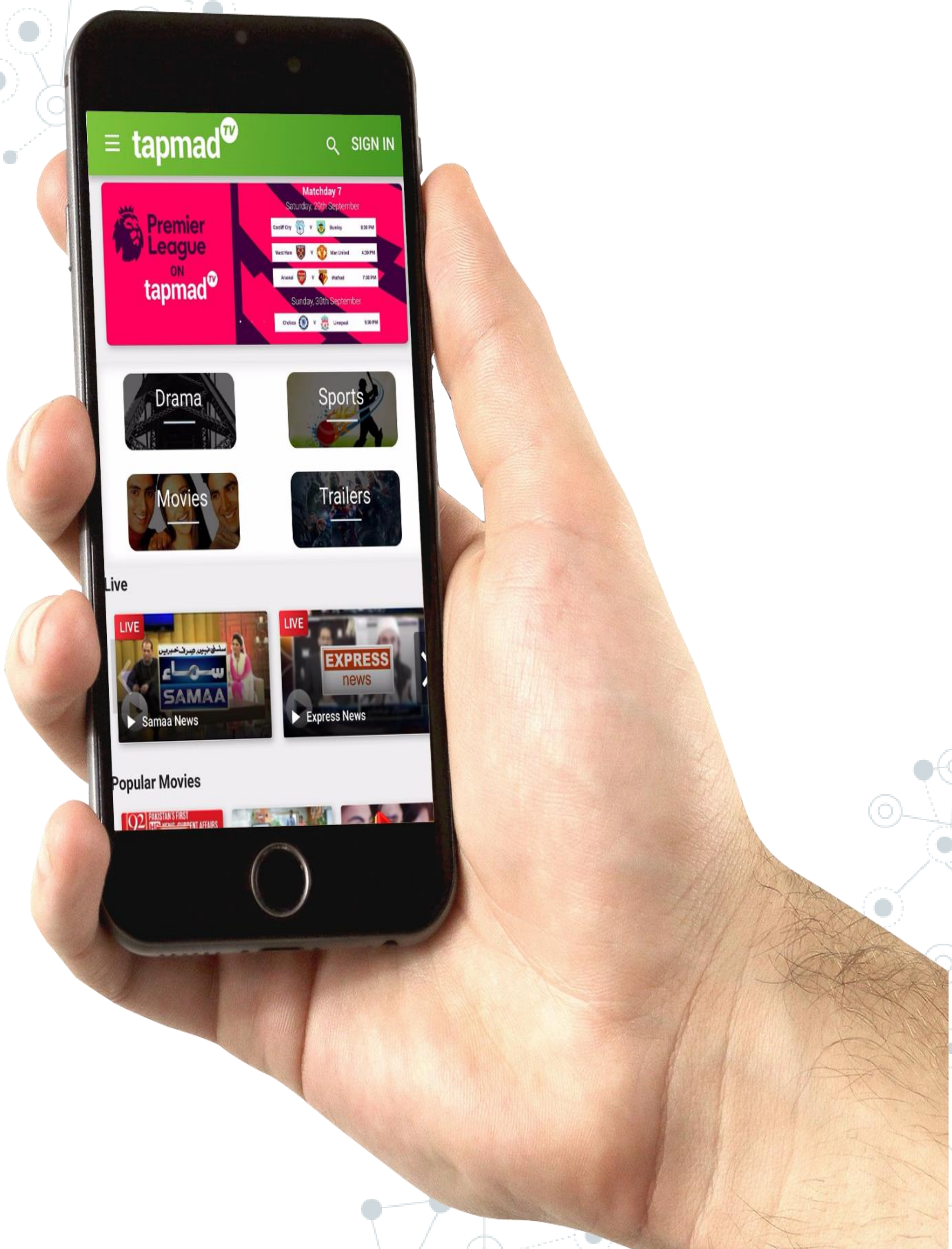
The Task:

✓ learn user preferences

✓ locate/recommend items that are "similar" to the user preferences



- ❑ Start presenting the users journey
- ❑ How a user comes in and make account(or not).
- ❑ What is the first Screen that he would see.
- ❑ How can he select option for the first time and then how would we like to show more items



Anatomy of TapmadTv Personalization

Everything is a Recommendation

Example

Everything is personalized

Note:
Recommendations are per household, not individual user

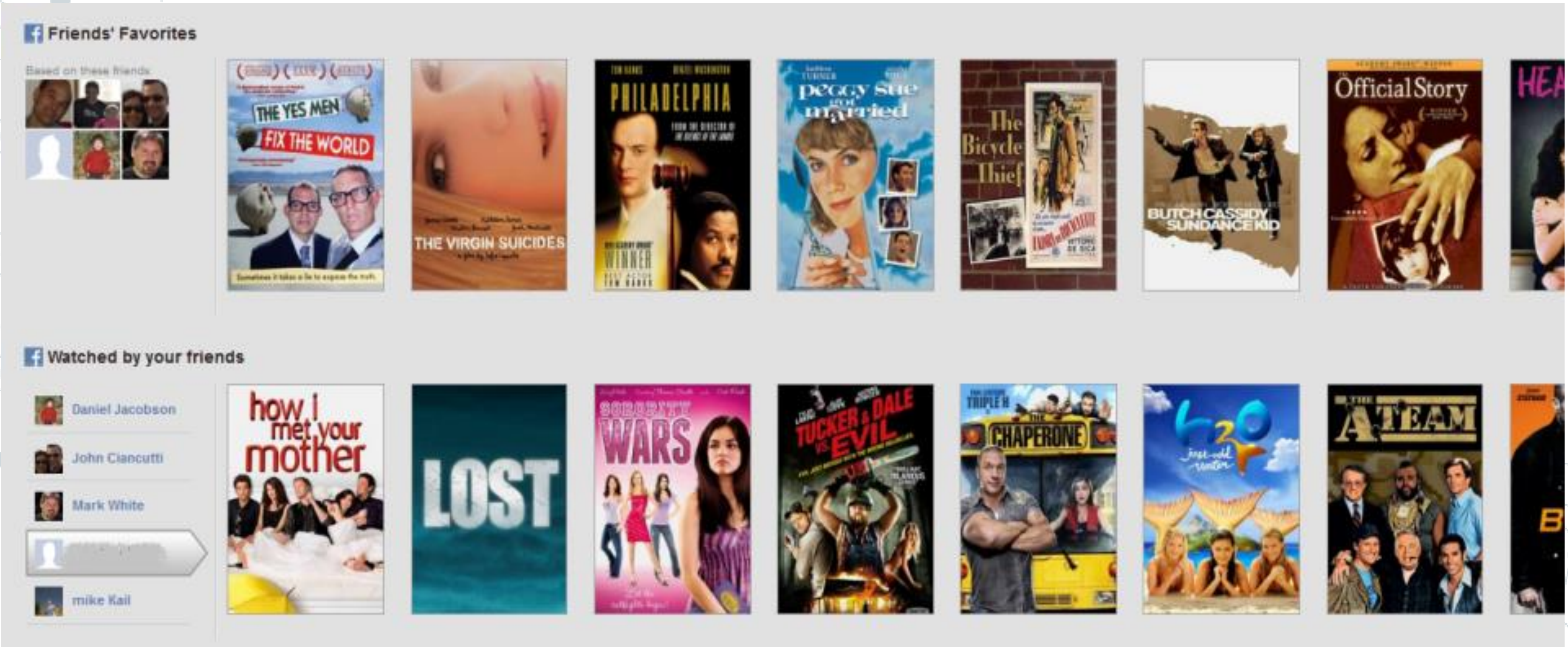
Example

Top 10

Personalization awareness

Diversity

Social Recommendations



Genres

NETFLIX

Genre rows

- Personalized genre rows focus on user interest

  - Also provide context and “evidence”
  - Important for member satisfaction – moving personalized rows to top on device increased retention

- How are they generated?

  - Implicit: data, such as “user viewed an item”, “user finished reading the article” or “user ordered a product”.
  - Explicit feedback is intentionally provided by users in form of clicking the “like”/”dislike” buttons, rating an item by number of stars, etc.

Example
Ranking

Key algorithm, sorts titles in most contexts

Ranking

- Ranking = Scoring + Sorting + Filtering bags of movies for presentation to a user
- Goal: Find the best possible ordering of a set of videos for a user within a specific context in real-time Objective: maximize consumption
- Aspirations: Played & “enjoyed” titles have best score
- Akin to CTR forecast for ads/search results

Cont…

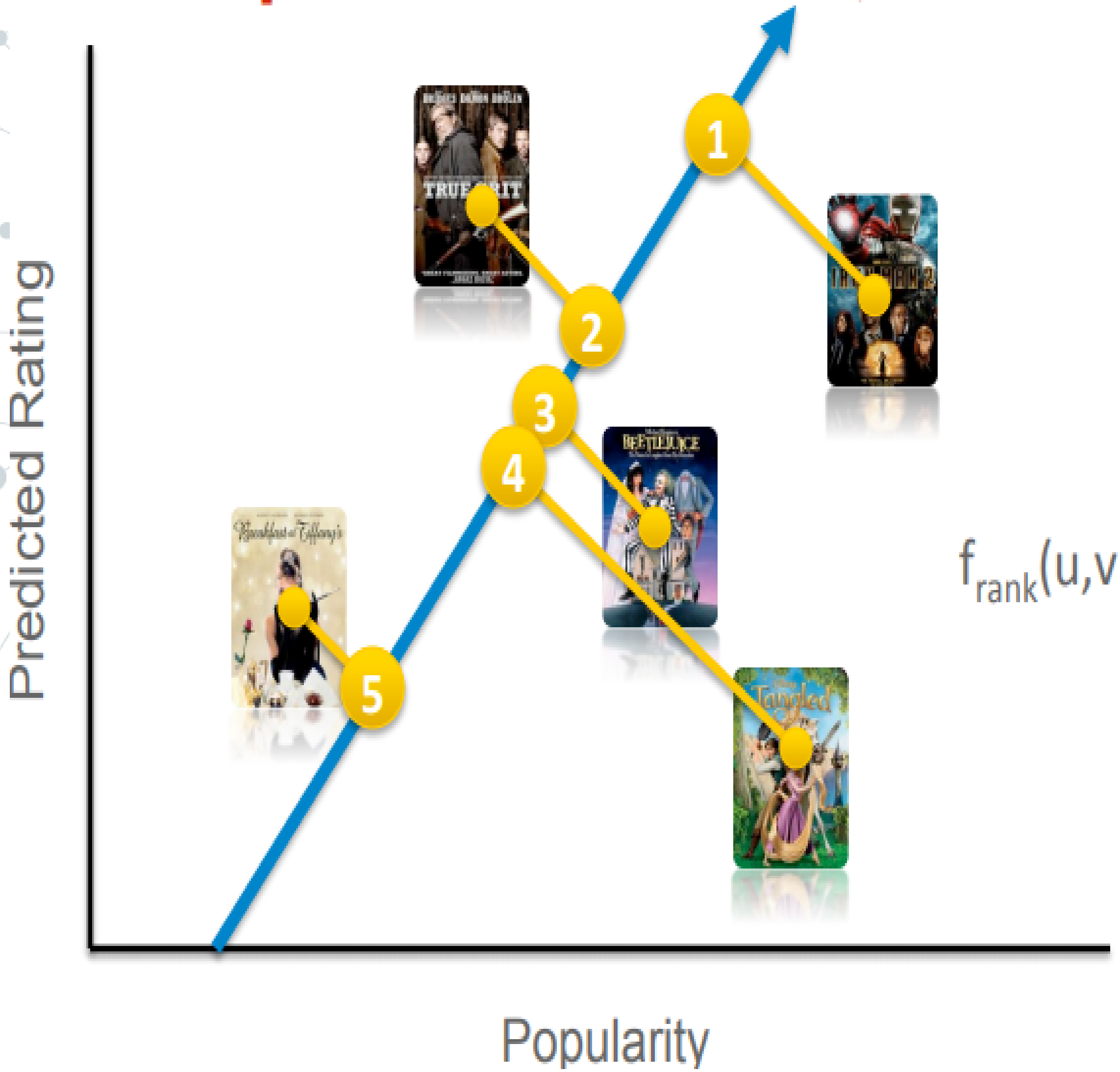
- Popularity is the obvious baseline

- Ratings prediction is a clear secondary data input that allows for personalization

- We have added many other features (and tried many more that have not proved useful)

- What about the weights?
  - Based on A/B testing
  - Machine-learned

Example: Two features, linear model



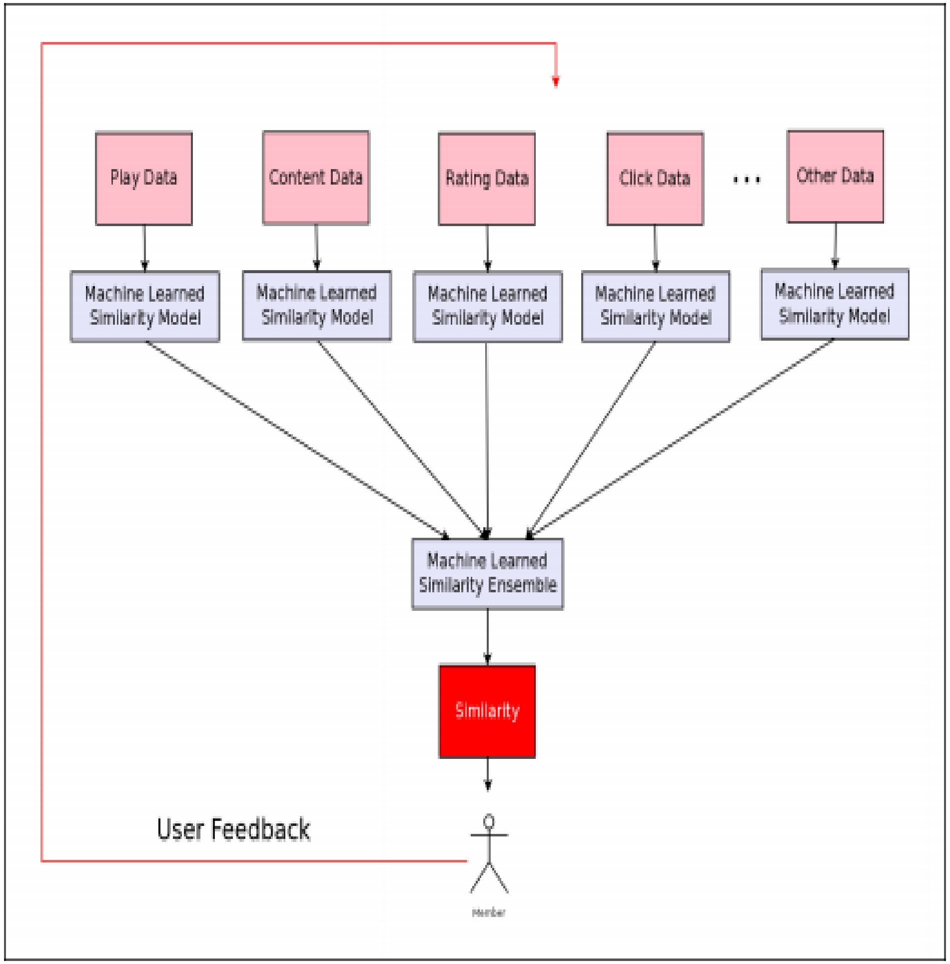
Linear Model:

$$f_{rank}(u,v) = w_1\, p(v) + w_2\, r(u,v) + b$$

Final Ranking

NETFLIX

Similars



- Different similarities computed from different sources: metadata, ratings, viewing data…
- Similarities can be treated as data/features
- Machine Learned models improve our concept of "similarity"

NETFLIX

Data & Models

- All sorts of feedback from the user can help generate better recommendations

- Need to design systems that capture and take advantage of all this data

- The right model is as important as the right data
- It is important to come up with new theoretical models, but also need to think about application to a domain, and practical issues

- Rating prediction models are only part of the solution to recommendation (think about ranking, similarity…)

Consumer (Data) Science

Consumer Science

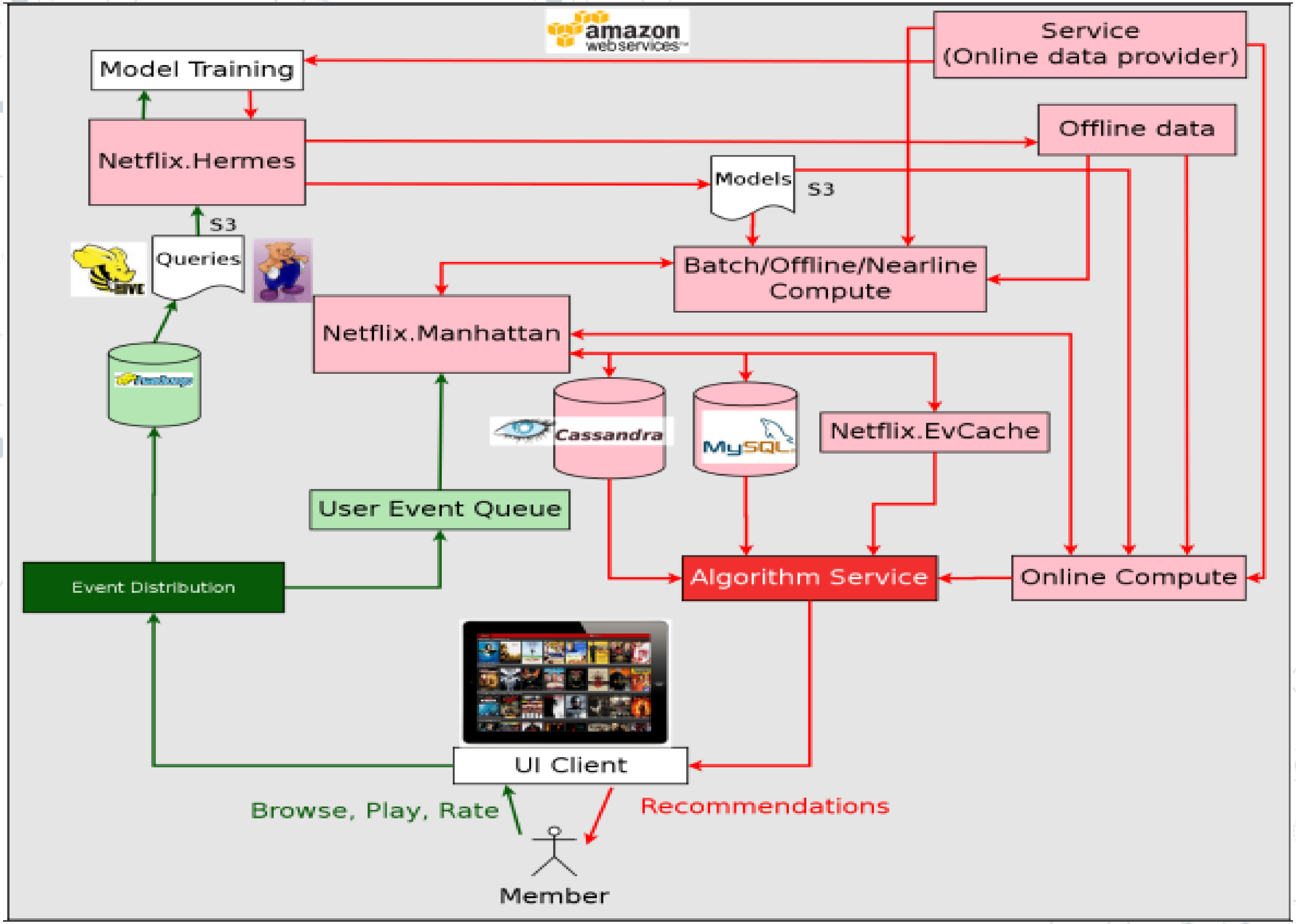
- Main goal is to effectively innovate for customers
- Innovation goals

-“If you want to increase your success rate, double your failure rate.” (Thomas Watson, Sr., founder of IBM)

- The only real failure is the failure to innovate
- Fail cheaply
- Know why you failed/succeeded

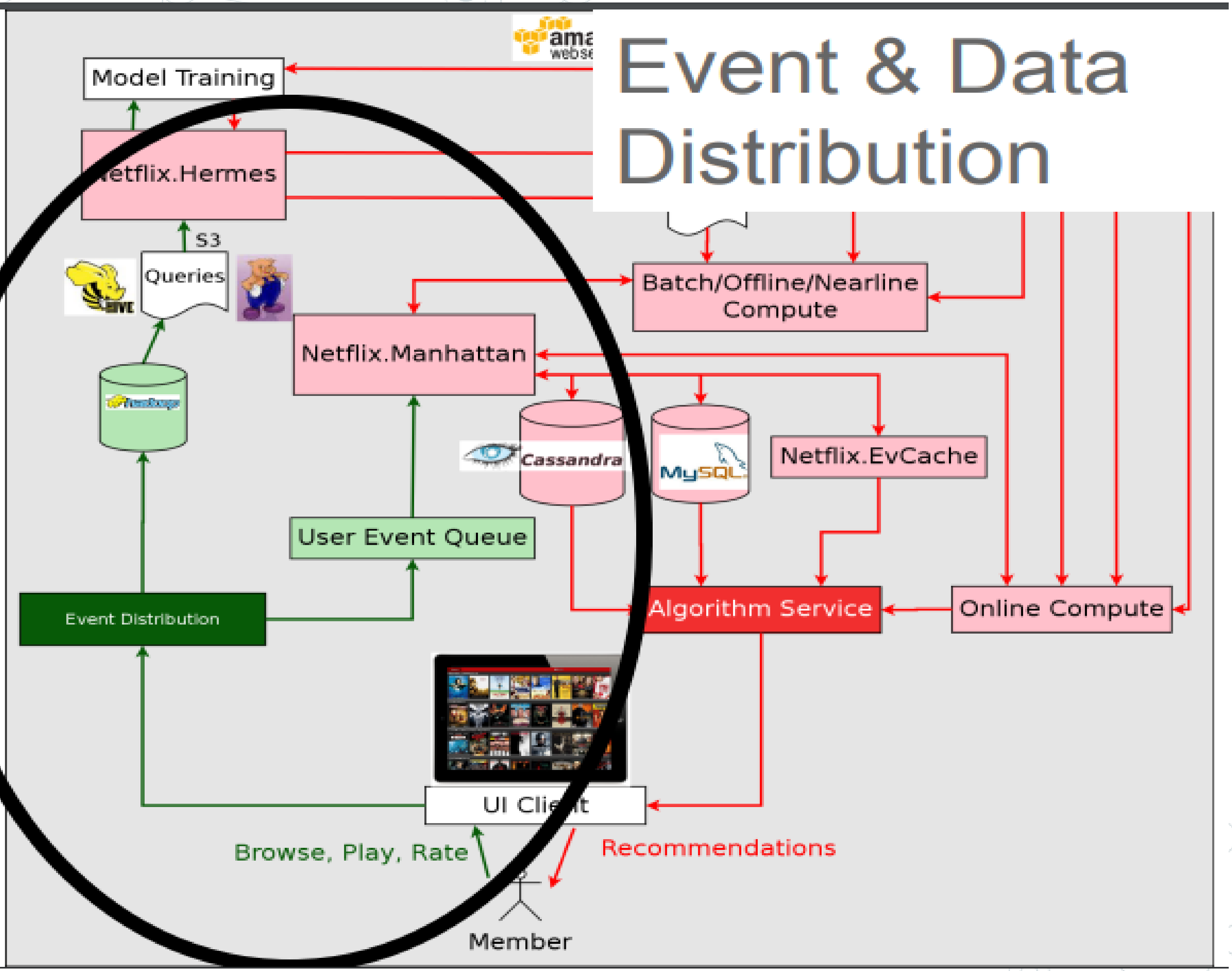
Architectures

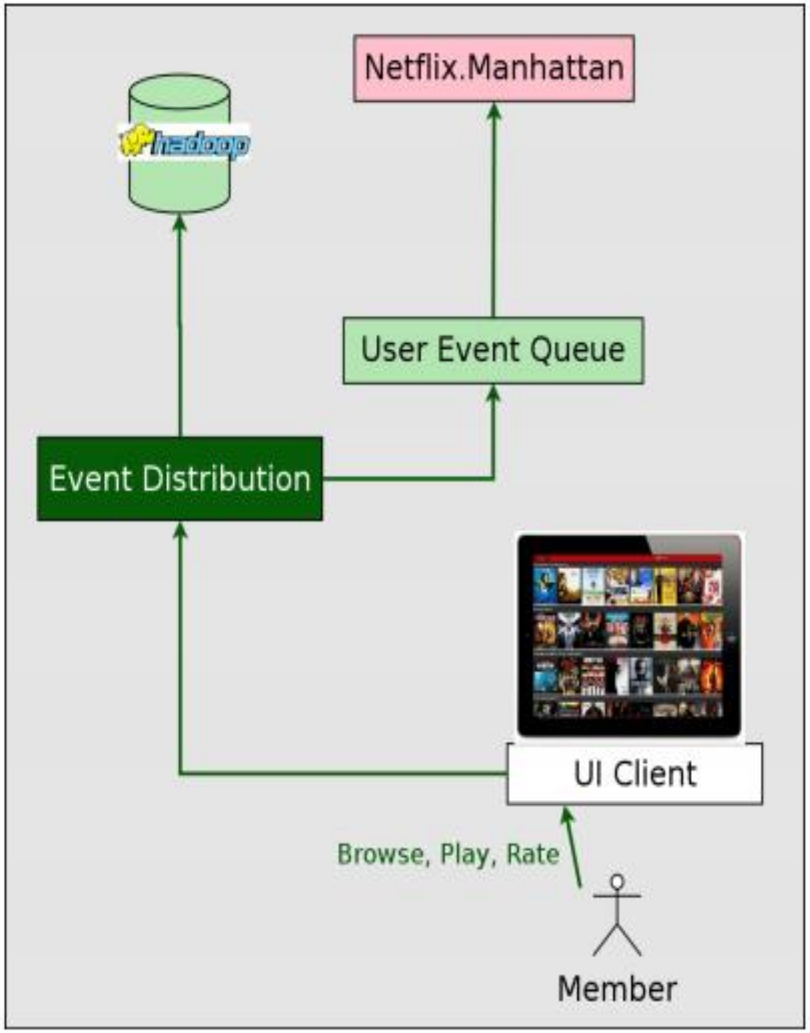
NETFLIX



amazon web services
Service (Online data provider)
Model Training
Offline data
Netflix.Hermes
Models
S3
S3
Queries
Batch/Offline/Nearline Compute
Netflix.Manhattan
Cassandra
MySQL
Netflix.EvCache
User Event Queue
Event Distribution
Algorithm Service
Online Compute
UI Client
Browse, Play, Rate
Recommendations
Member

Event & Data Distribution



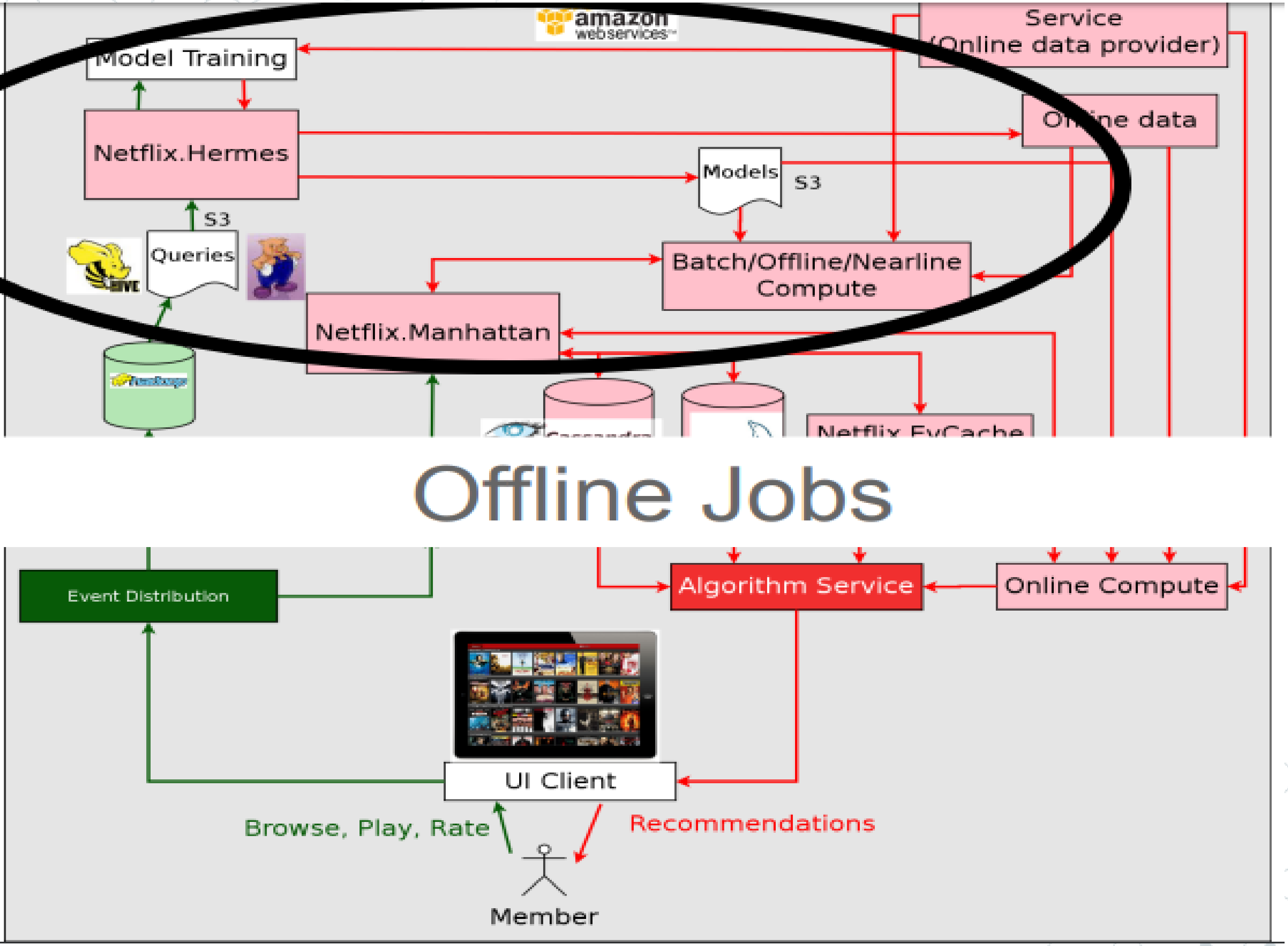


Event & Data Distribution

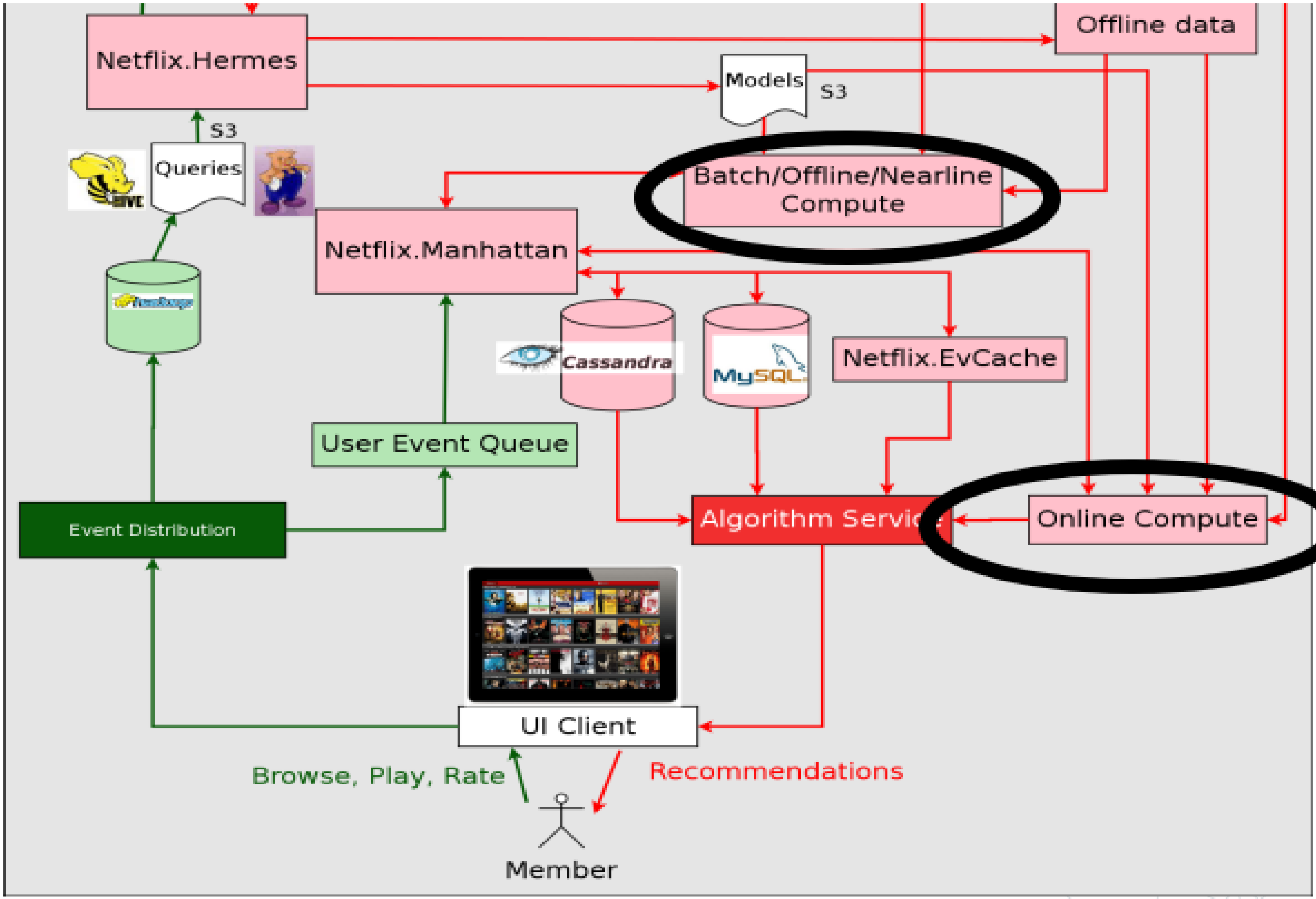
- UI devices should broadcast many different kinds of user events
  - Clicks
  - Presentations
  - Browsing events
  - …
- Events vs. data
  - Some events only need to be propagated and trigger an action (low latency, low information per event)
  - Others need to be processed and “turned into” data (higher latency, higher information quality).
  - And… there are many in between
- Real-time event flow managed through internal tool (Manhattan)
- Data flow mostly managed through Hadoop.

NETFLIX

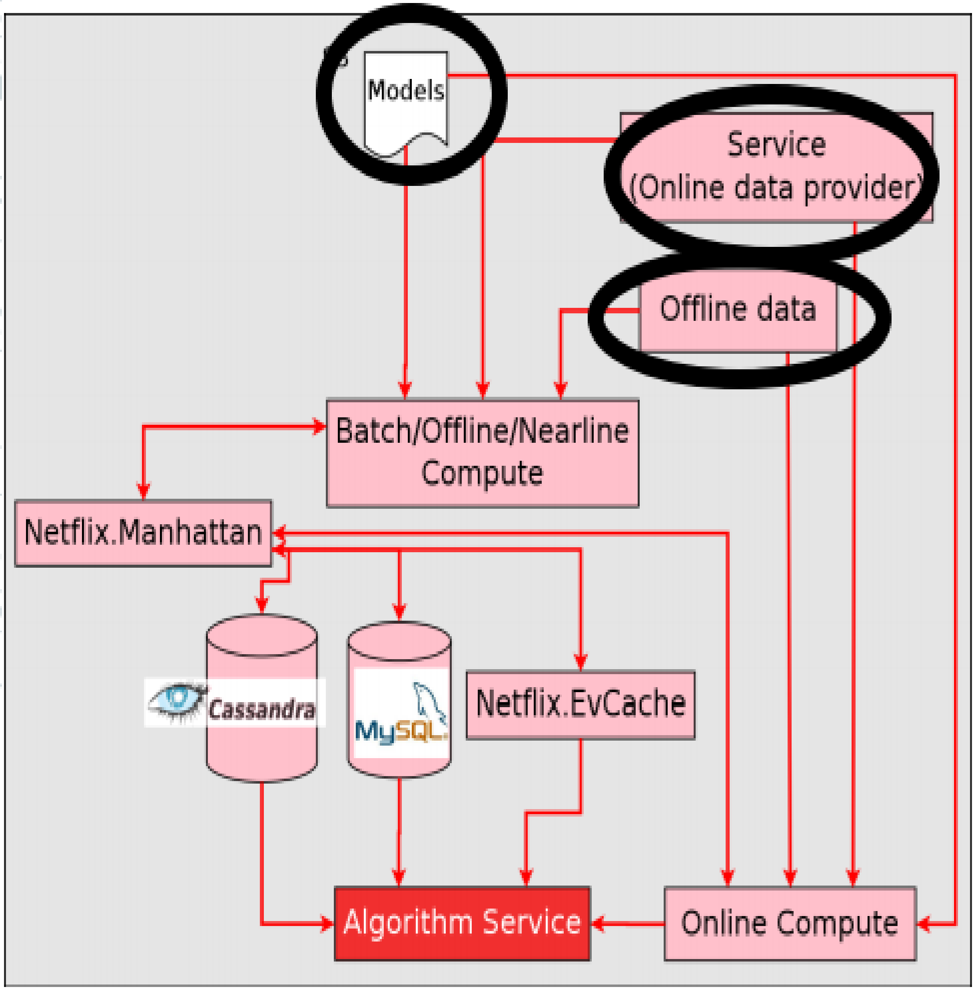
Offline Jobs



Computation



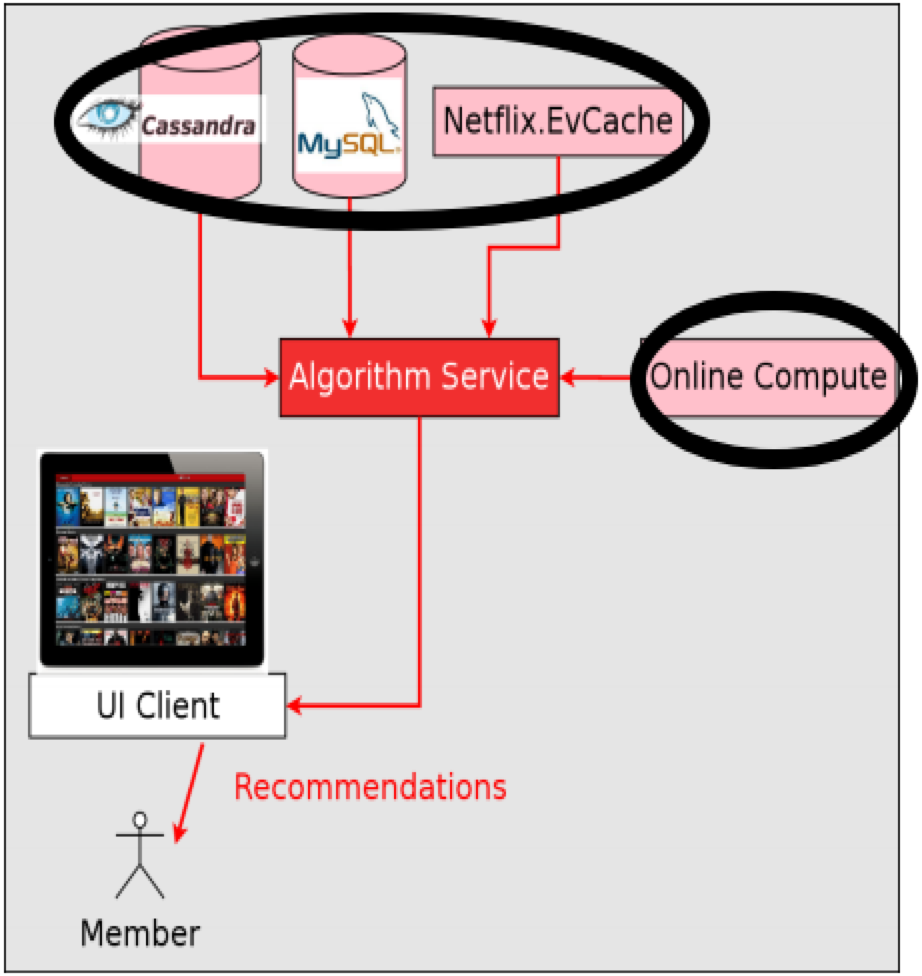
Signals & Models



- Both offline and online algorithms are based on three different inputs:
  - **Models:** previously trained from existing data
  - (Offline) **Data**: previously processed and stored information
  - **Signals:** fresh data obtained from live services
    - User-related data
    - Context data (session, date, time…)

NETFLIX

Results



- Recommendations can be serviced from:
  - Previously computed lists
  - Online algorithms
  - A combination of both
- The decision on where to service the recommendation from can respond to many factors including context.
- Also, important to think about the fallbacks (what if plan A fails)
- Previously computed lists/intermediate results can be stored in a variety of ways
  - Cache
  - Cassandra
  - Relational DB

Reference

- https://medium.com/netflix-techblog/system-architectures-for-personalization-and-recommendation-e081aa94b5d8

- https://www.slideshare.net/justinbasilico/deep-learning-for-recommender-systems-92331718

Thanks!

Any questions?