



**PARIS**

---

**APEXA INVESTMENT PRODUCT RECOMMENDATION SERVICE**



# AGENDA

---

- What is a neural network
- How do we build recommendation system
- Proposition
- PARIS
- Q&A

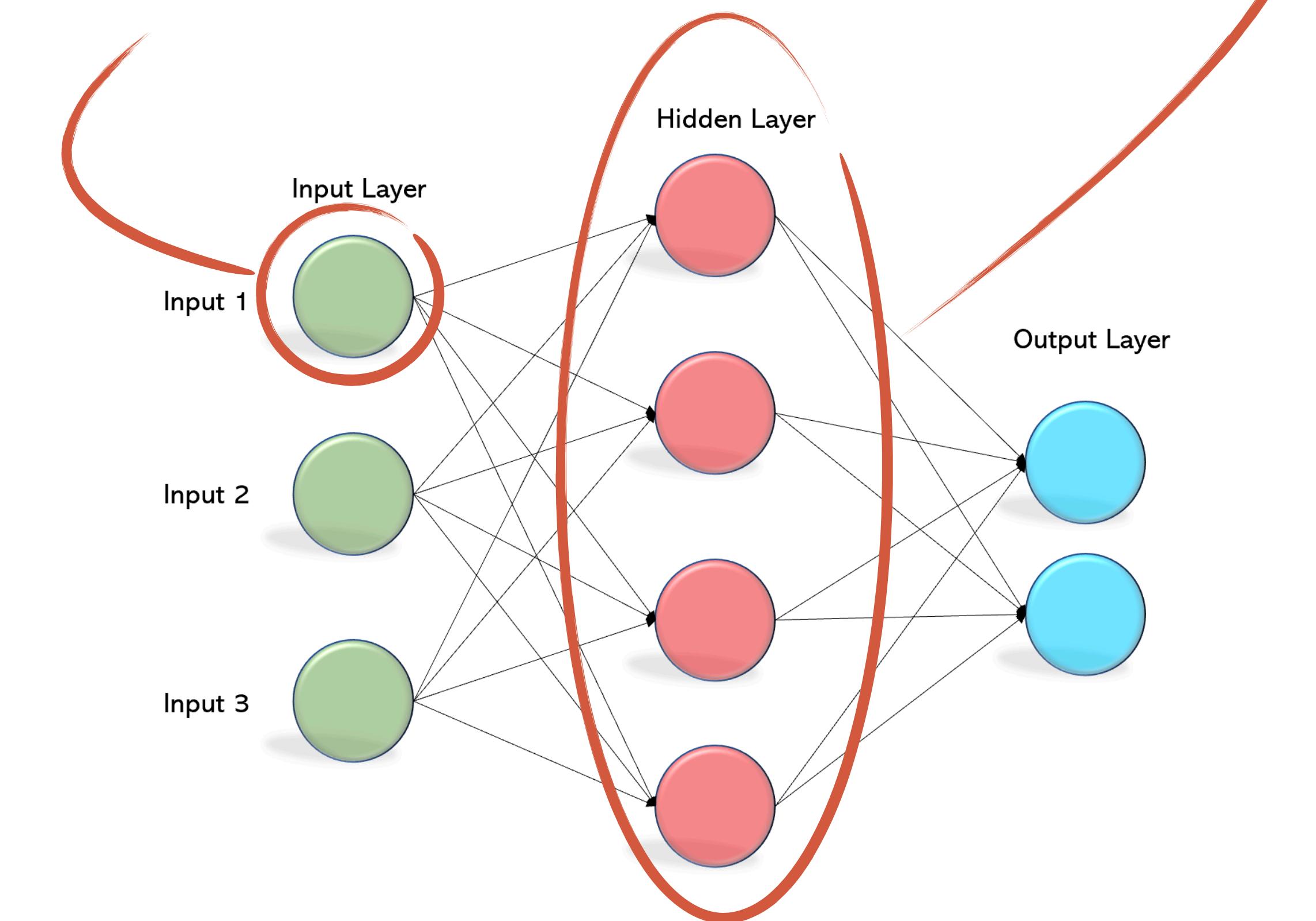
# **WHAT IS A NEURAL NETWORK**



It is a method of finding the relationship between  
a collection of inputs and its corresponding  
outputs



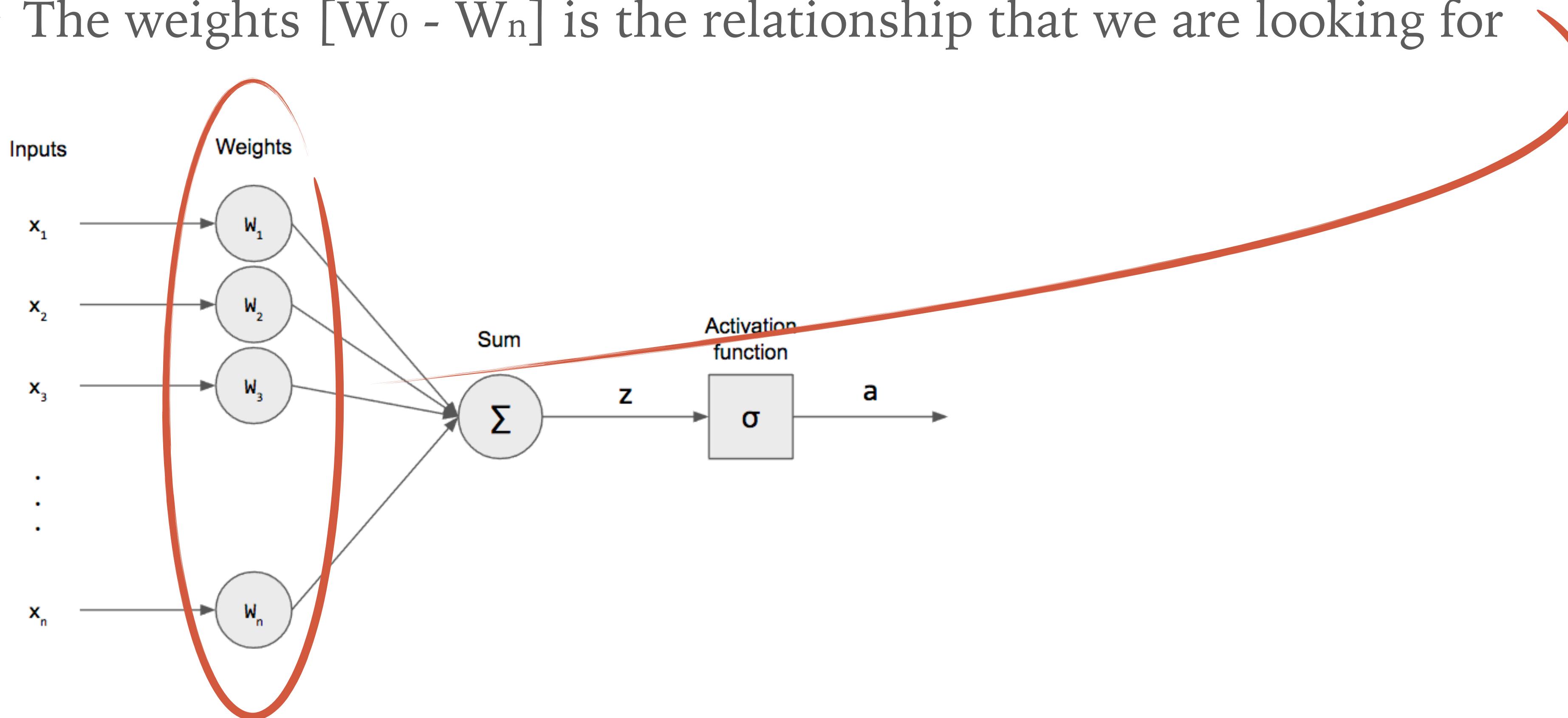
- It is made up of 1 or more layer
- Each layer is made up of 1 or more neuron (perceptron)



# NEURON (PERCEPTRON)

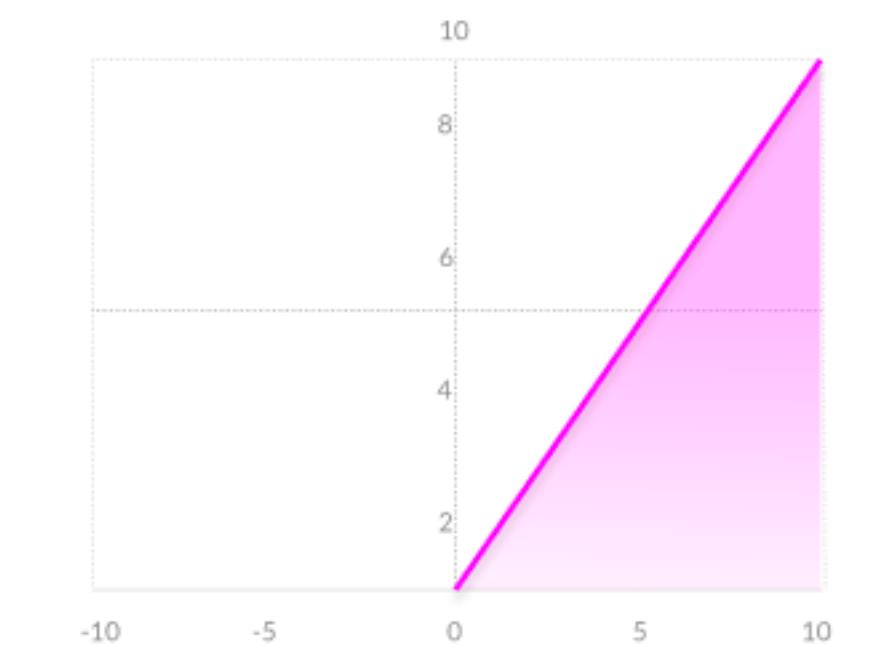
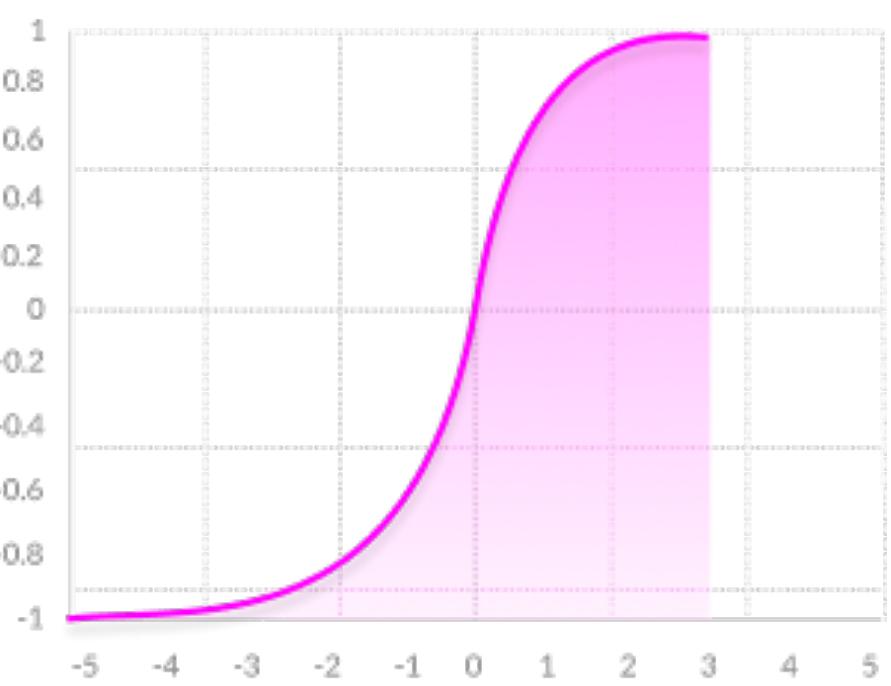
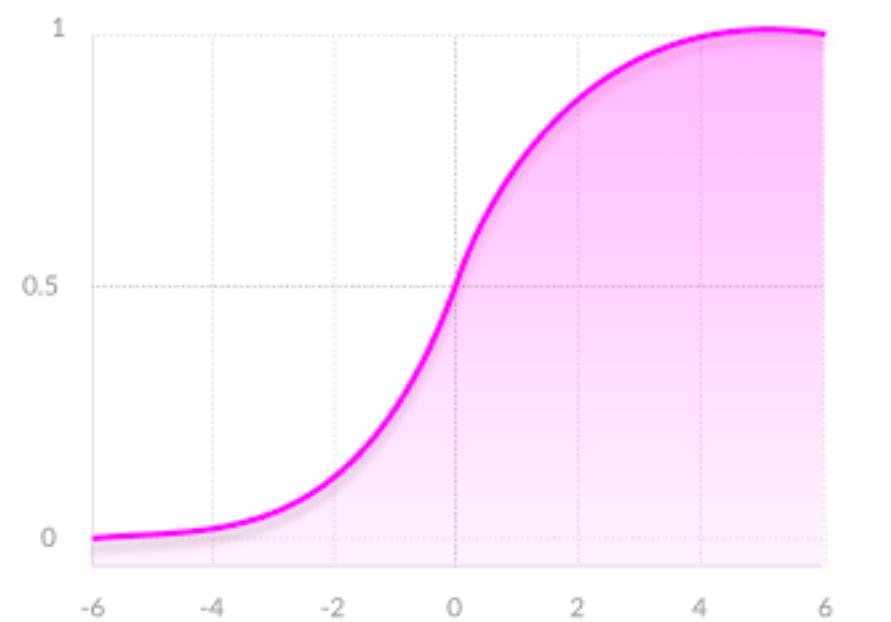
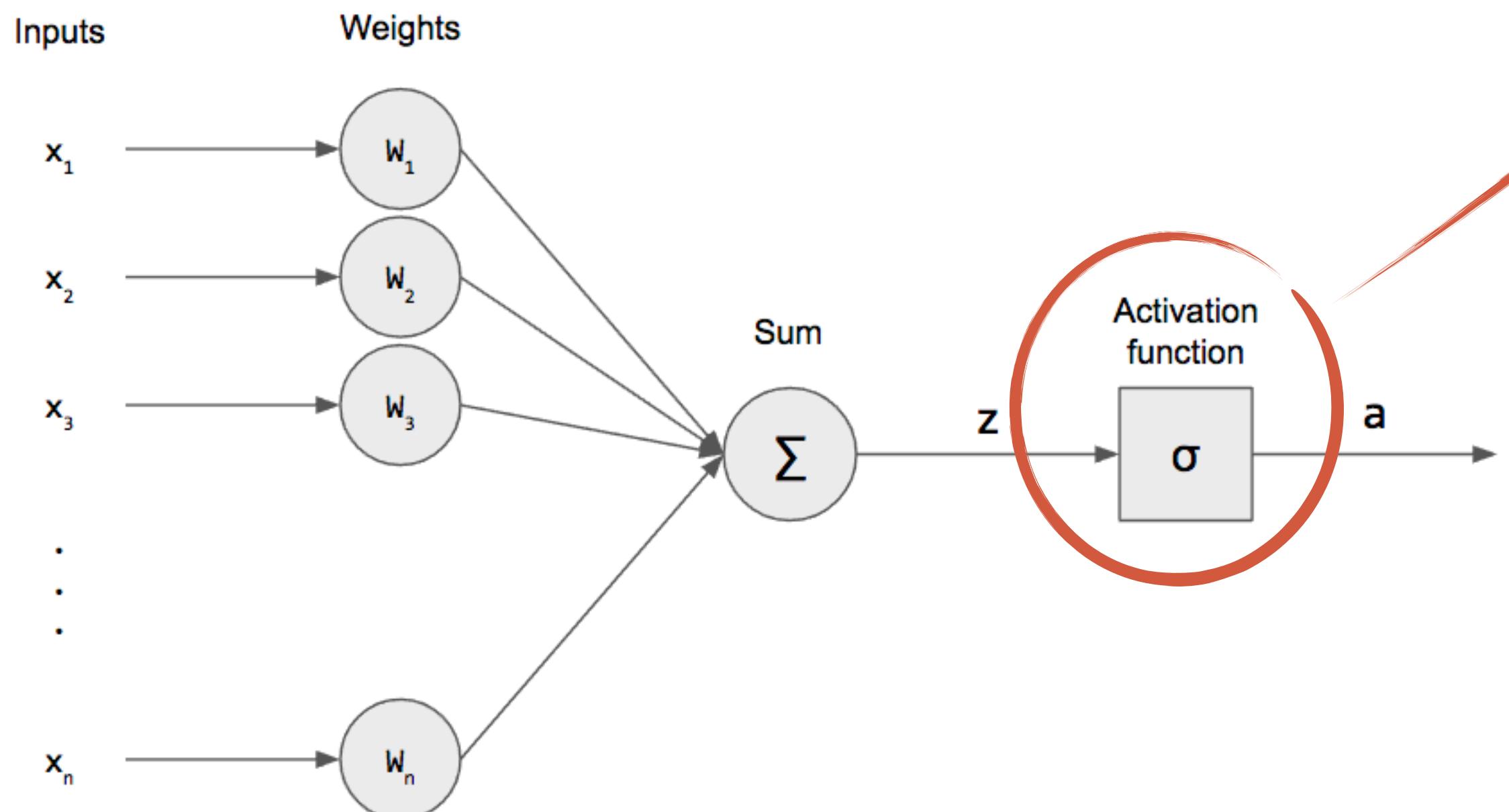
---

- It is the most basic unit in a neural network
- The weights [ $W_0 - W_n$ ] is the relationship that we are looking for



# STEP FUNCTION AKA ACTIVATION FUNCTION

- A function to limit the output's data range
- Add non-linearity into a neural network



# HOW TO INTERPRET NEURAL NETWORK RESULT (LOSS / ACCURACY)

---

- Loss is a measurement of the difference between the actual and predicted values
- Accuracy is a measurement of how many predictions the model has got right



We train the neural network to workout 1 set of weights that work across the entire dataset (seen or unseen) and we are using the weight for predictions

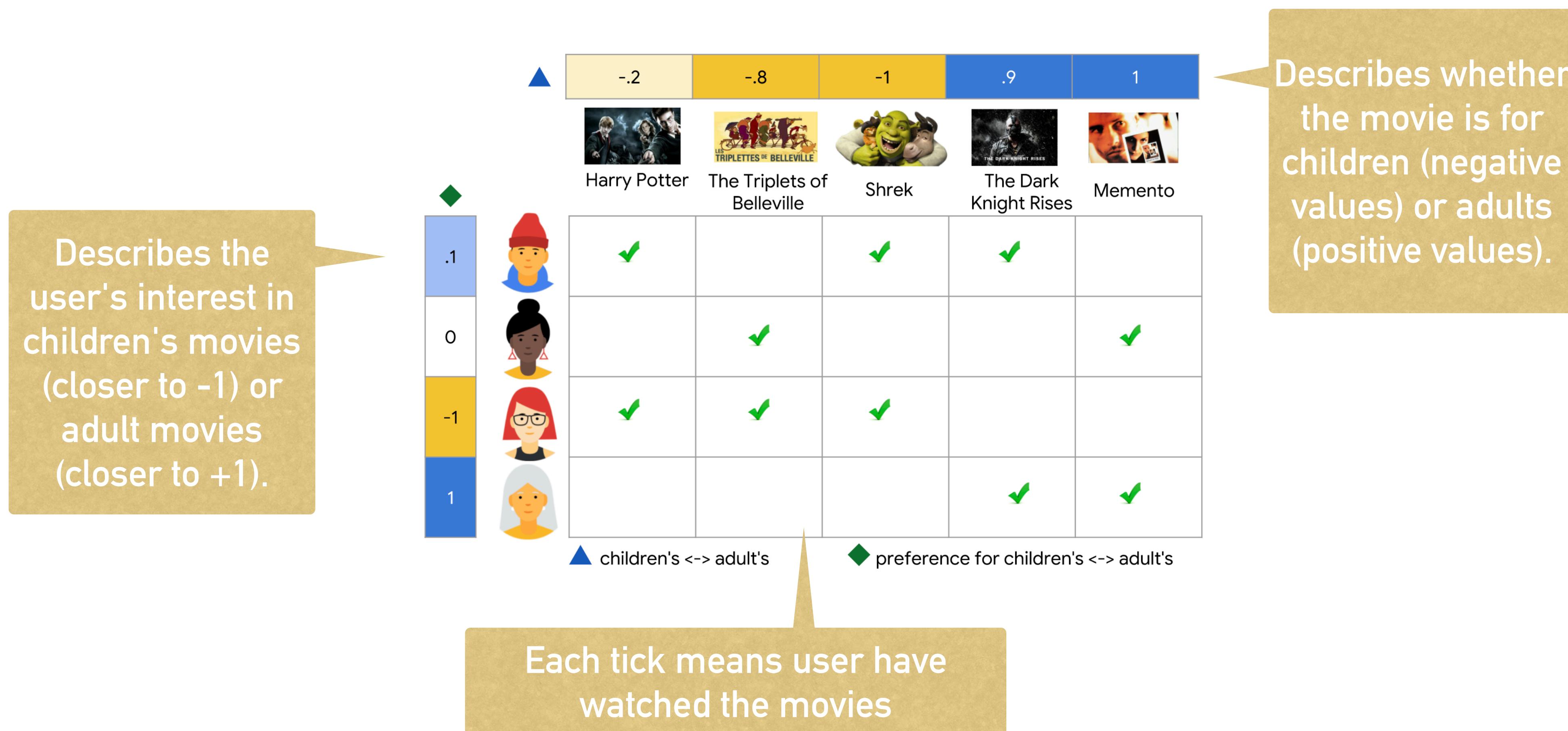
# HOW DO WE BUILD RECOMMENDATION SYSTEM (FROM PAST TO PRESENT)

# MATRIX FACTORISATION (MF)

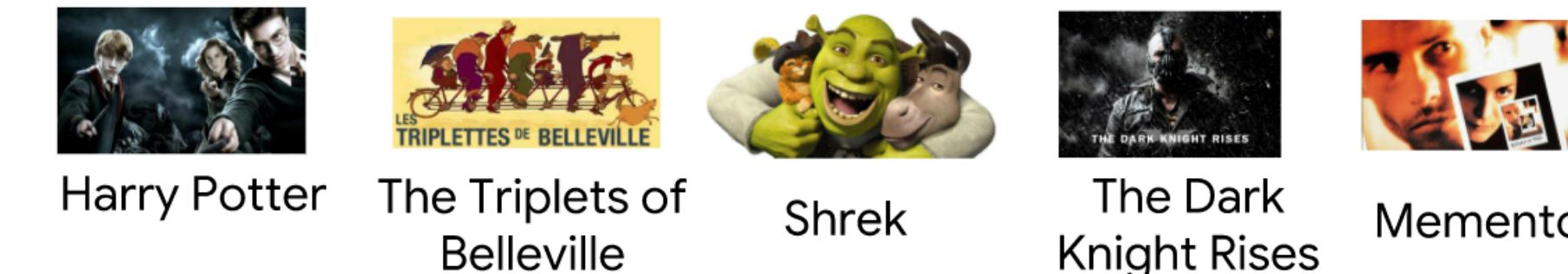
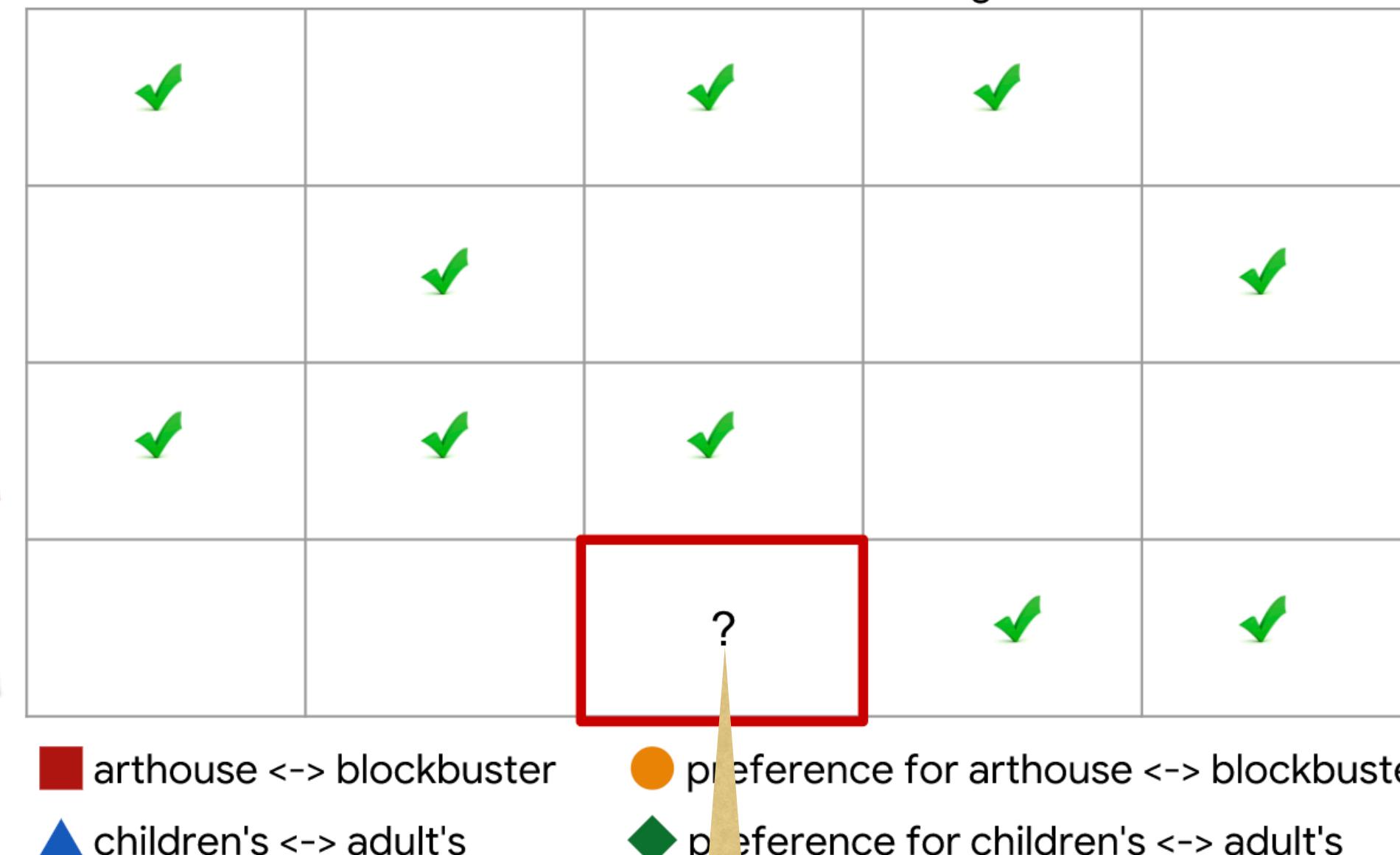
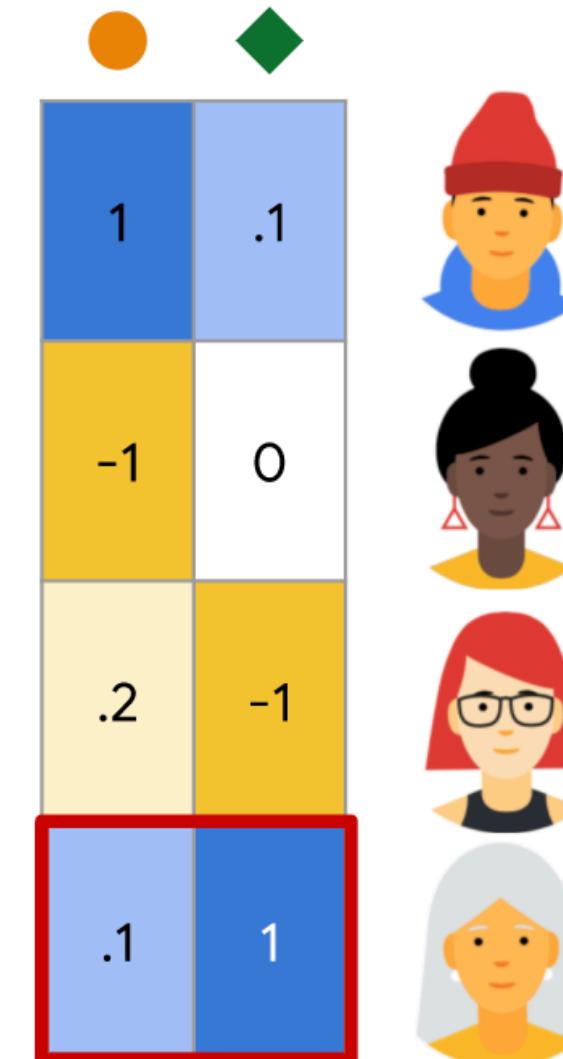
# A VARIANT OF COLLABORATIVE FILTERING (CF)

# WHAT IS A MATRIX FACTORIZATION (MF)

- A dot product of 2 matrix to approximate the level of interest between the customers and the products



Describes the user's interest in Arthouse (closer to -1) or Blockbuster (closer to +1)



Describes whether the movie is Arthouse (negative values) or Blockbuster (positive values)

We can use dot product between 2 matrix to find the missing values



Dot Product of  $[0.1, 1]$  and  $[1, -1]$  give us  $-0.9$

# LIMITATION

We can see very clearly that row  $U_4$  is similar to row  $U_1$ ,  $U_3$  and then row  $U_2$

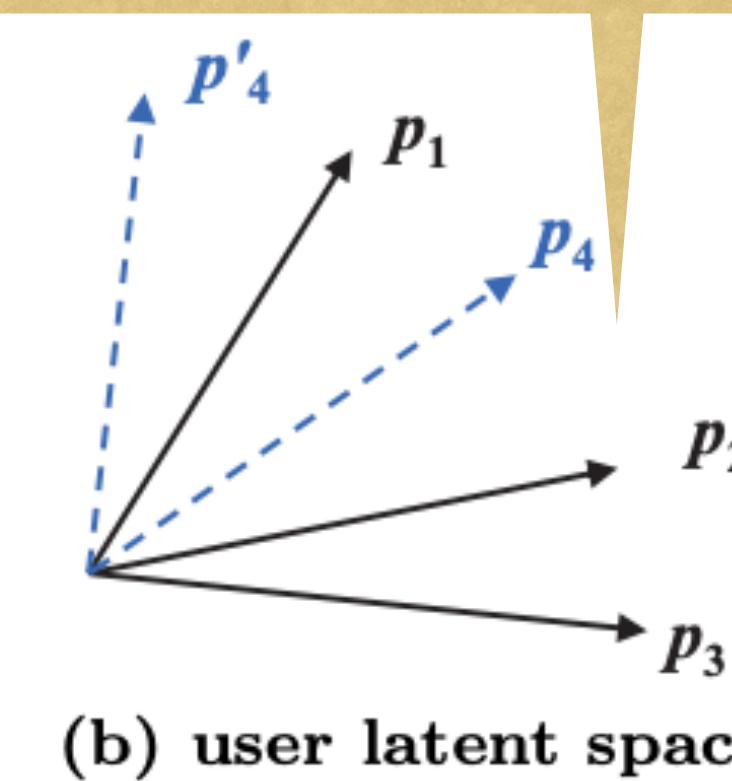
|       | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ |
|-------|-------|-------|-------|-------|-------|
| $u_1$ | 1     | 1     | 1     | 0     | 1     |
| $u_2$ | 0     | 1     | 1     | 0     | 0     |
| $u_3$ | 0     | 1     | 1     | 1     | 0     |
| $u_4$ | 1     | 0     | 1     | 1     | 1     |

(a) user-item matrix

Transform

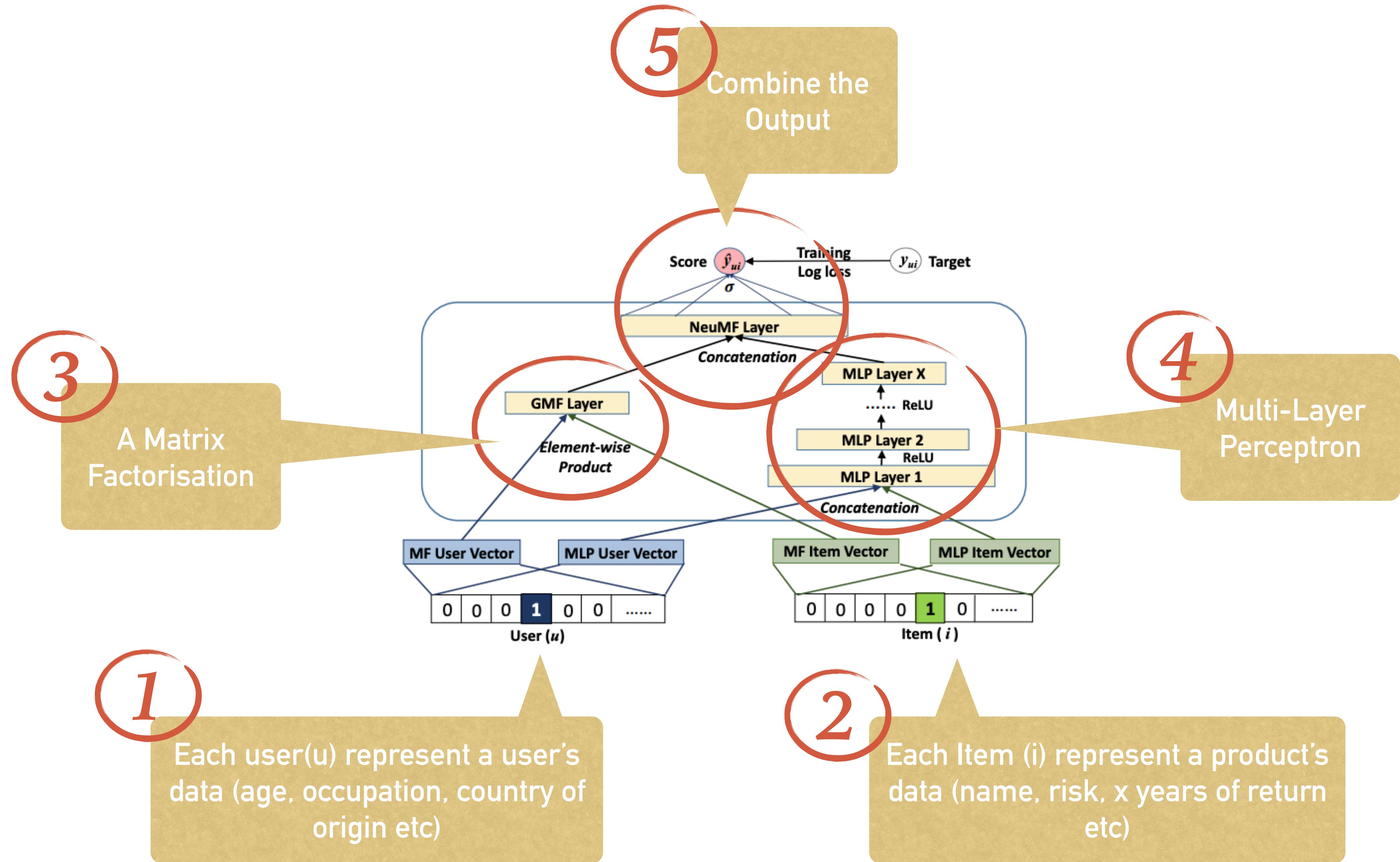
However if we transform each row to another plane, we see a different result

We can see that after transformation,  $P_4$  is similar to  $P_1$ ,  $P_2$  then  $P_3$  which contradict to the 1st conclusion, incurring a large ranking loss



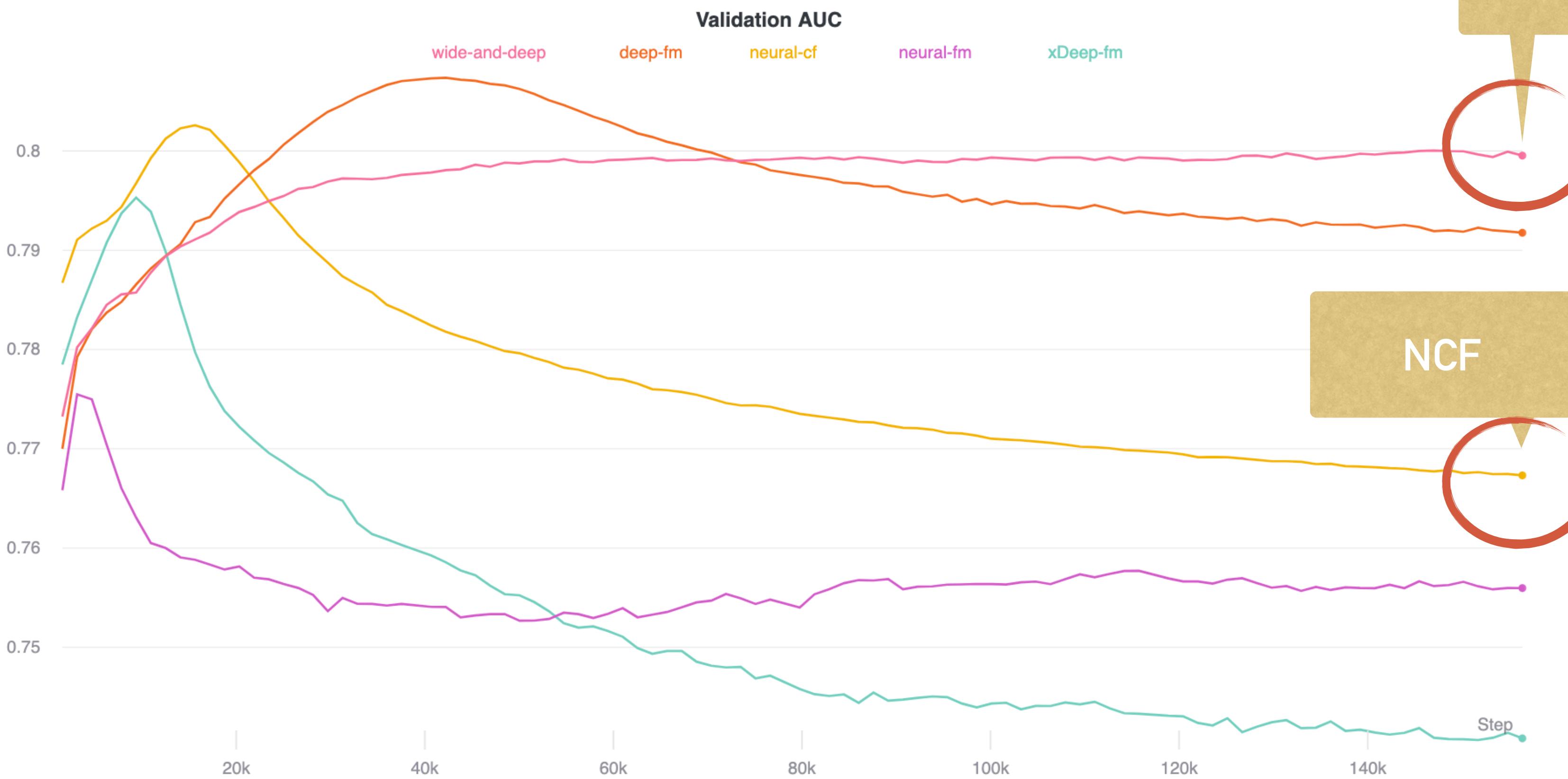
# **USE OF NEURAL NETWORK IN COLLABORATIVE FILTERING (NCF)**

# **NEURAL MATRIX FACTORISATION (NEUMF) A VARIANT OF NCF**



# WHY NCF?

Wide-and-deep, developed by  
GOOGLE and launched on PlayStore  
(2006)



| Model         | Test AUC | Valid AUC | Runtime  |
|---------------|----------|-----------|----------|
| wide-and-deep | 0.7991   | 0.7995    | 1h12m15s |
| deep-fm       | 0.7915   | 0.7918    | 1h06m50s |
| xDeep-fm      | 0.7429   | 0.7408    | 2h15m17s |
| neural-fm     | 0.7589   | 0.7560    | 1h36m0s  |
| neural-cf     | 0.7668   | 0.7673    | 54m15s   |

Minimal runtime with  
outstanding accuracy

|            |   |
|------------|---|
| Mask R-CNN | Mask R-CNN  |
| ShapeMask  | ShapeMask: Learning to Segment Novel Objects by Refining Shape Priors       |
| SpineNet   | SpineNet: Learning Scale-Permuted Backbone for Recognition and Localization |

## On the Official TensorFlow

### Natural Language Processing

| Model  | Reference (Paper)  |
|--|--|
| ALBERT (A Lite BERT)   | ALBERT: A Lite BERT for Self-supervised Learning of Language Representations     |
| BERT (Bidirectional Encoder Representations from Transformers) | BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding |
| NHNet (News Headline generation model)                         | Generating Representative Headlines for News Stories                             |
| Transformer  | Attention Is All You Need  |
| XLNet  | XLNet: Generalized Autoregressive Pretraining for Language Understanding         |

### Recommendation

| Model | Reference (Paper)              |
|-------|--------------------------------|
| NCF   | Neural Collaborative Filtering |

NCF is the only recommended model for Recommendation system

### How to get started with the official models

- The models in the master branch are developed using TensorFlow 2, and they target the TensorFlow [nightly binaries](#) built from the [master branch of TensorFlow](#).
- The stable versions targeting releases of TensorFlow are available as tagged branches or [downloadable releases](#).
- Model repository version numbers match the target TensorFlow release, such that [release v2.2.0](#) are compatible with [TensorFlow v2.2.0](#).

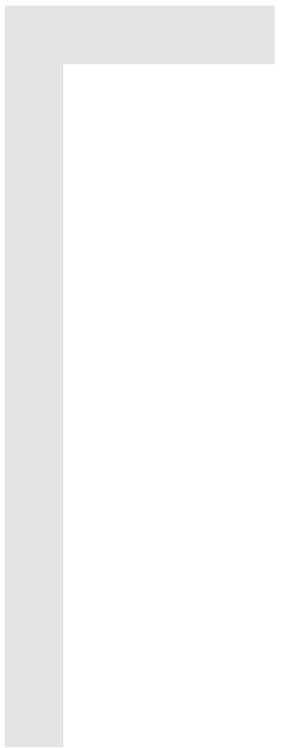
Please follow the below steps before running models in this repository.

### Requirements

- The latest TensorFlow Model Garden release and TensorFlow 2
  - If you are on a version of TensorFlow earlier than 2.2, please upgrade your TensorFlow to [the latest TensorFlow 2](#).

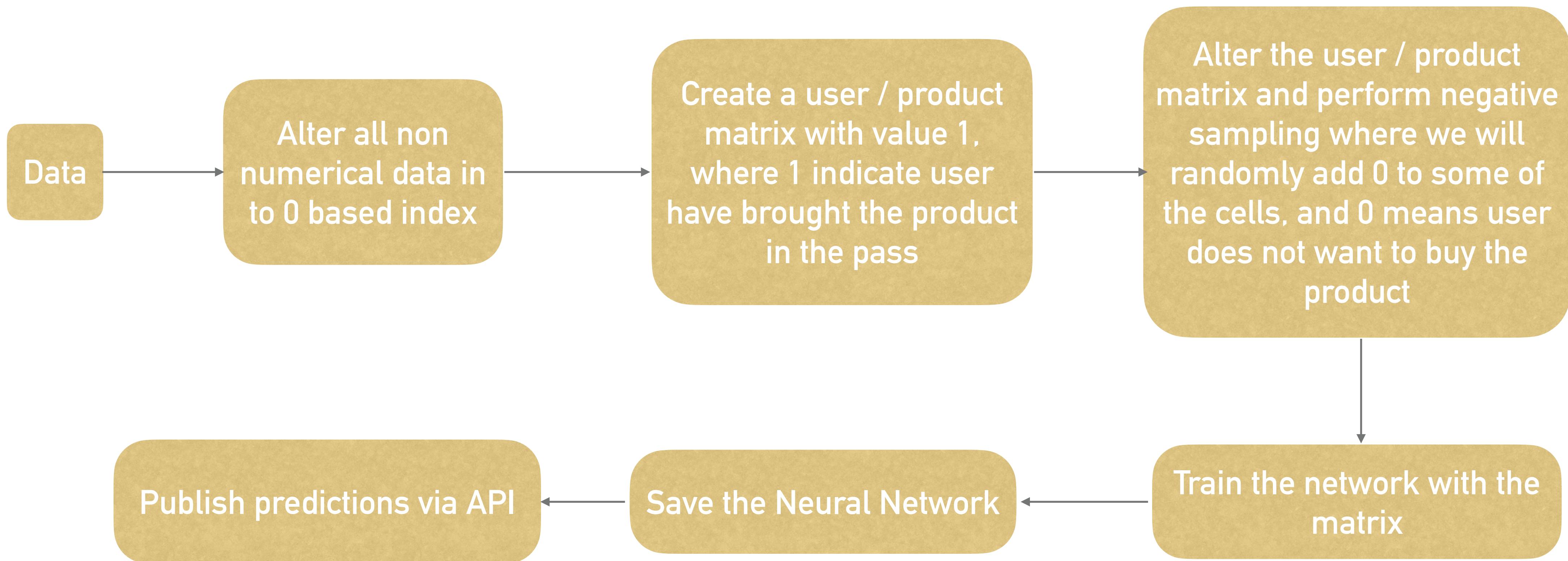
```
pip3 install tf-nightly
```

# PROPOSITION



PARIS aim to compare different portfolios and predict whether a customer should be interested in certain products or not

# BUILDING A RECOMMENDATION SYSTEM FROM SCRATCH



# PARIS

1

Insomnia – product recommendation

POST http://localhost:5000/recomm... Send

200 OK 56.1 ms 427 B 5 Days Ago

No Environment Cookies

JSON Auth Query Header 1

Preview Header Cookie Timeline

Filter

Recommendation System API

POST product recommendation

GET user recommendation

GET get data

```
1 {  
2   "user": "CUST00000134",  
3   "age": 20,  
4   "gender": "M",  
5   "maritalStatus": "SINGLE",  
6   "haveChild": "N",  
7   "education": "SECONDARY"  
8 }
```

\$..store.books[\*].author

Given a New / Existing user, we are able to compute a list of products that he/she is interested in

2

Insomnia – user recommendation

GET http://localhost:5000/recomm... Send

200 OK 1.16 s 44 KB Just Now

No Environment Cookies

JSON Auth Query Header 1

Preview Header 5 Cookie Timeline

Filter

Recommendation System API

POST product recommendation

GET user recommendation

GET get data

```
1 [  
2   {  
3     "product_name": "U62300",  
4     "3year_return": "11.37",  
5     "standard_deviation": "20.12",  
6     "dividend": "0.22",  
7     "asset_class": "Equity Developed Market"  
8   },  
9   {  
10    "probability (%)": 99.9965548515,  
11    "user": "CUST00000499"  
12  },  
13  {  
14    "probability (%)": 99.9965369701,  
15    "user": "CUST00000714"  
16  },  
17  {  
18    "probability (%)": 99.9961972237,  
19    "user": "CUST00000995"  
20  },  
21  {  
22    "probability (%)": 99.9959468842,  
23    "user": "CUST00000921"  
24  },  
25  {  
26    "probability (%)": 99.9957799911,  
27    "user": "CUST00000984"  
28  },  
29  {  
30    "probability (%)": 99.9955117702,  
31    "user": "CUST00000802"  
32  },  
33  {  
34    "probability (%)": 99.9950289726,  
35    "user": "CUST00000150"  
36  }]
```

\$..store.books[\*].author

Given a New / Existing product, we are able to compute a list of customers that they are interested in

# IMPROVEMENTS FOR THE FUTURE

---

- labeling data with 1 and 0 as what we want to predict yield a relatively high loss in training the neural network
- due to a limited dataset available, left us very little room for maneuver
- We require rich data set from various sources i.e.
  - How long did a user use the app, web, length of the meeting etc
  - How long did a user browse a product
- The model currently did not account for time series pattern, i.e.
  - If there is an economic breakdown, holding stocks might not be a good choice, in this case, the model should return gold, or bonds instead.
- The model is unable to handle sudden change in the market

# QUESTION