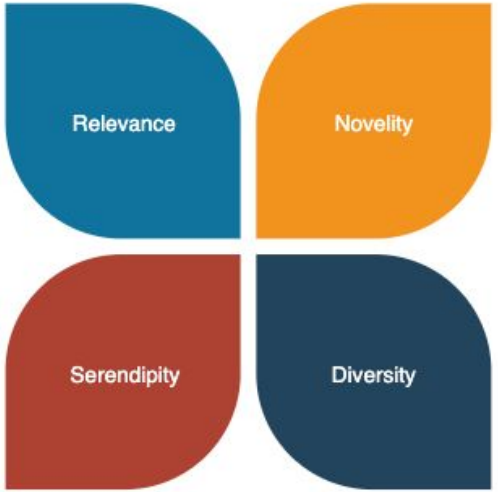
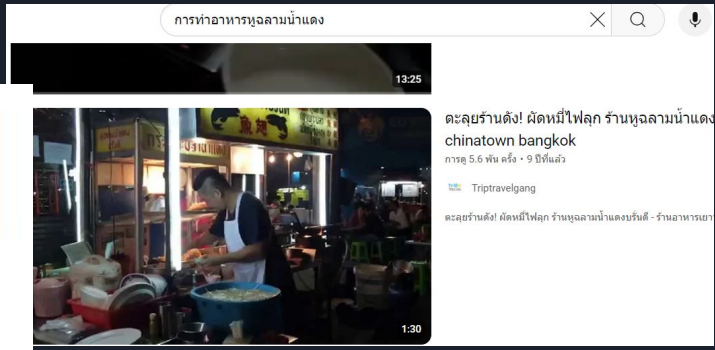
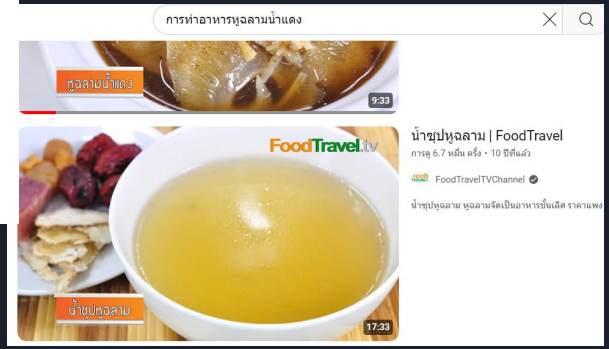
Recommended System (RecSys)

The Remarkable World of Recommender Systems







reference: https://towardsdatascience.com/the-remarkable-world-of-recommender-systems-bff4b9cbe6a7

Exploratory Data Analysis (EDA)

Book Recommended System Datasets & EDA

About books_df files:

Books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset.

Moreover, some content-based information is given (Book-Title, Book-Author, Year-Of-Publication, Publisher), **obtained from Amazon Web Services.**

Note that in case of several authors, only the first is provided. URLs linking to cover images are also given, appearing in three different flavours (Image-URL-S, Image-URL-M, Image-URL-L), i.e., small, medium, large. These URLs point to the Amazon web site.

	ISBN	Book-Title	Book-Autho
0	0195153448	Classical Mythology	Mark P. O. M
1	0002005018	Clara Callan	Richard Bru
2	0060973129	Decision in Normandy	Carlo D'Este
3	0374157065	Flu: The Story of the Great Influenza Pandemic of 1918 and the Search for the Virus	Gina Bari Ko
4	0393045218	The Mummies of Urumchi	E. J. W. Barb
5	0399135782	The Kitchen God's Wife	Amy Tan
6	0425176428	What If?: The World's Foremost Military Historians Imagine What Might Have Been	Robert Cow
7	0671870432	PLEADING GUILTY	Scott Turow

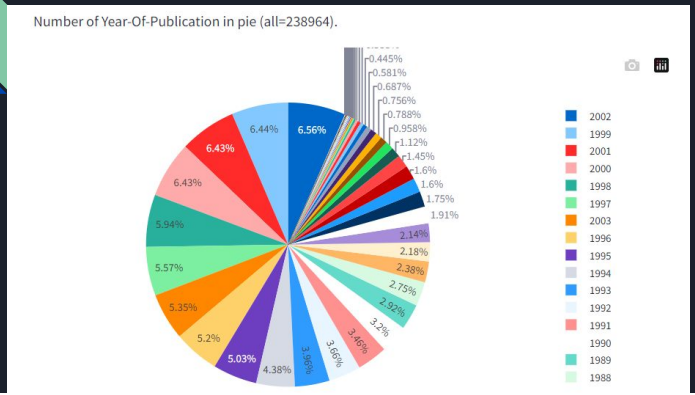
https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset

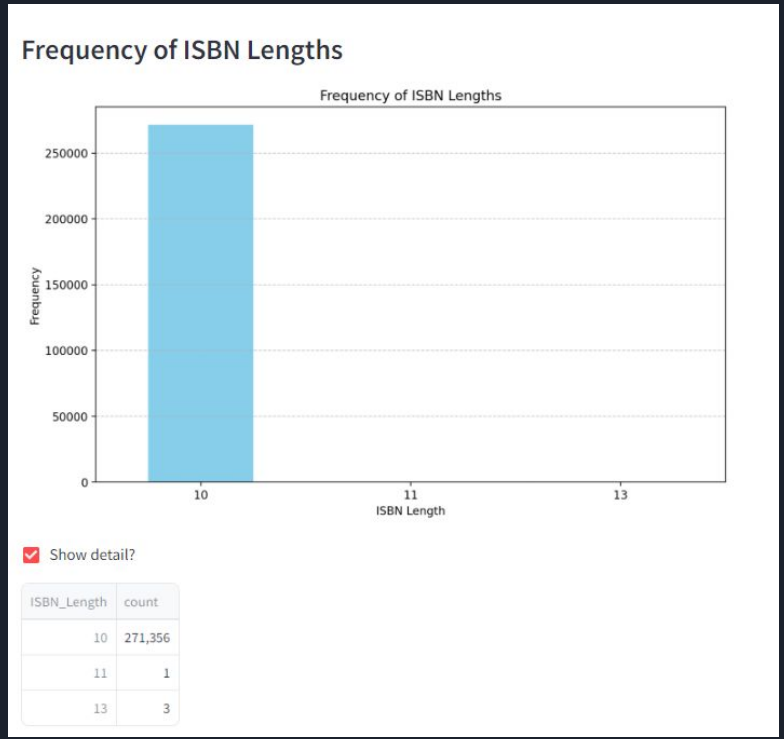
Book Recommended System Datasets & EDA

ISNB เลขมาตรฐานสากลประจำหนังสือ (International Standard Book Number)

https://www.nupress.grad.nu.ac.th/international-standard-book-number/

Book Recommended System Datasets & EDA





https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset

Book Recommended System Datasets & EDA

About users_df files:

Contains the users.

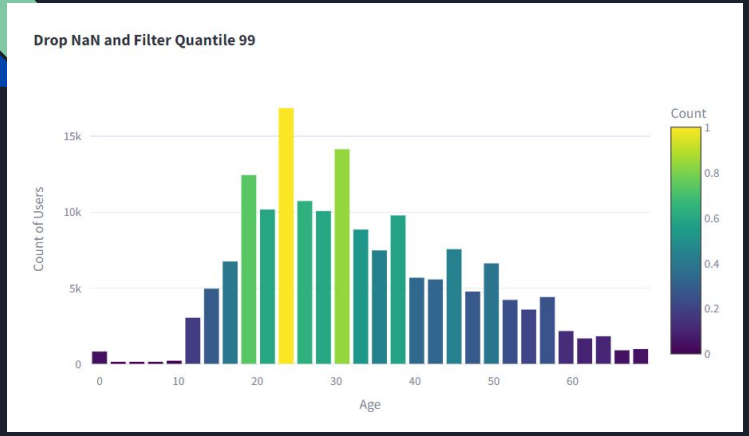
Note that user IDs (User-ID) have been anonymized and map to integers.

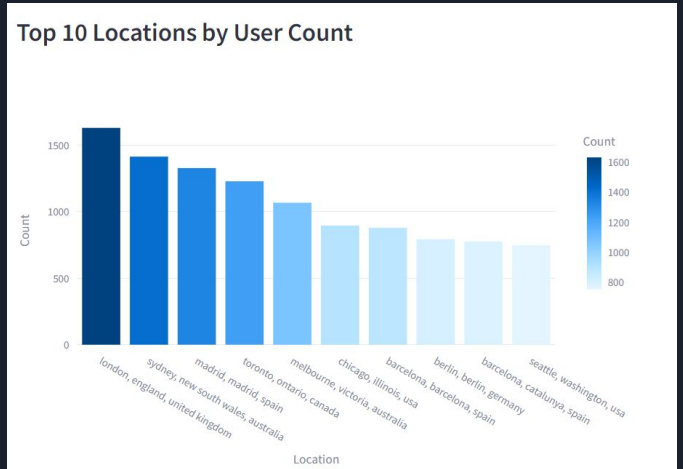
Demographic data is provided (Location, Age) if available. Otherwise, these fields contain NULL-values.

	User-ID	Location	Age
0	1	nyc, new york, usa	None
1	2	stockton, california, usa	18
2	3	moscow, yukon territory, russia	None
3	4	porto, v.n.gaia, portugal	17
4	5	farnborough, hants, united kingdom	None
5	6	santa monica, california, usa	61
6	7	washington, dc, usa	None
7	8	timmins, ontario, canada	None
8	9	germantown, tennessee, usa	None
9	10	albacete, wisconsin, spain	26

https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset

Book Recommended System Datasets & EDA





note: sample because data is to big cann’t run

Book Recommended System Datasets & EDA

About ratings_df files:

Contains the book rating information.

Ratings (Book-Rating) are either

explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or

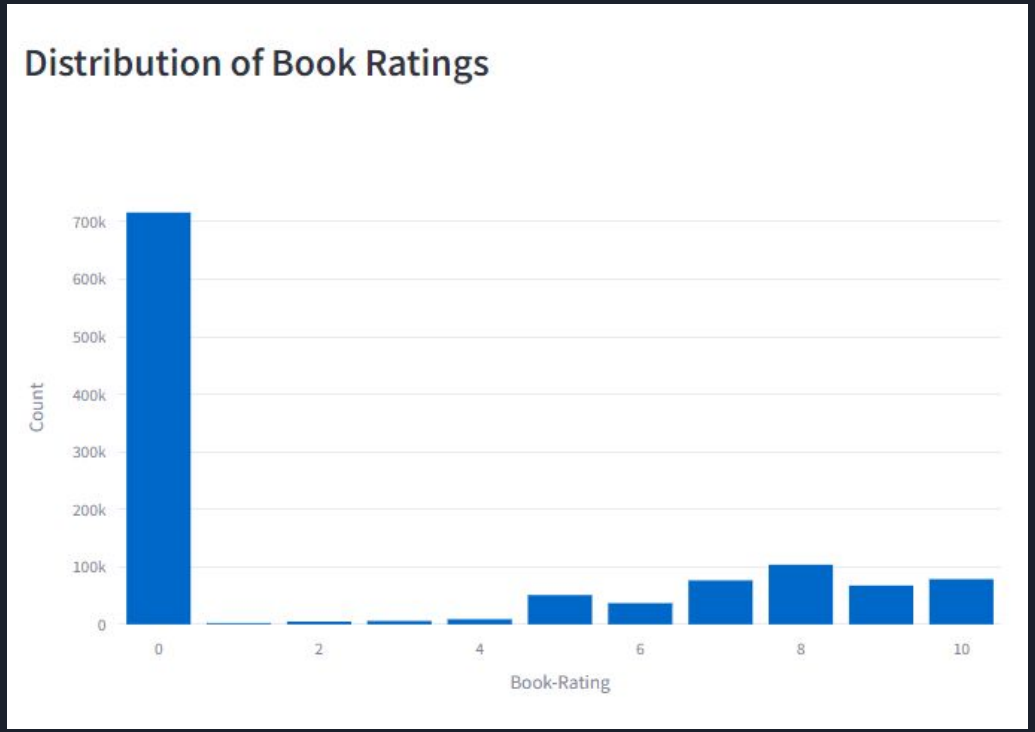
implicit, expressed by 0.

	User-ID	ISBN	Book-Rating
0	276725	034545104X	0
1	276726	0155061224	5
2	276727	0446520802	0
3	276729	052165615X	3
4	276729	0521795028	6
5	276733	2080674722	0
6	276736	3257224281	8
7	276737	0600570967	6
8	276744	038550120X	7
9	276745	342310538	10

Describe Summary statistics of the ratings

	Book-Rating
count	1,149,780
mean	2.867
std	3.8542
min	0
25%	0
50%	0
75%	7
max	10

In ratings_df all have row=1149780 column=3

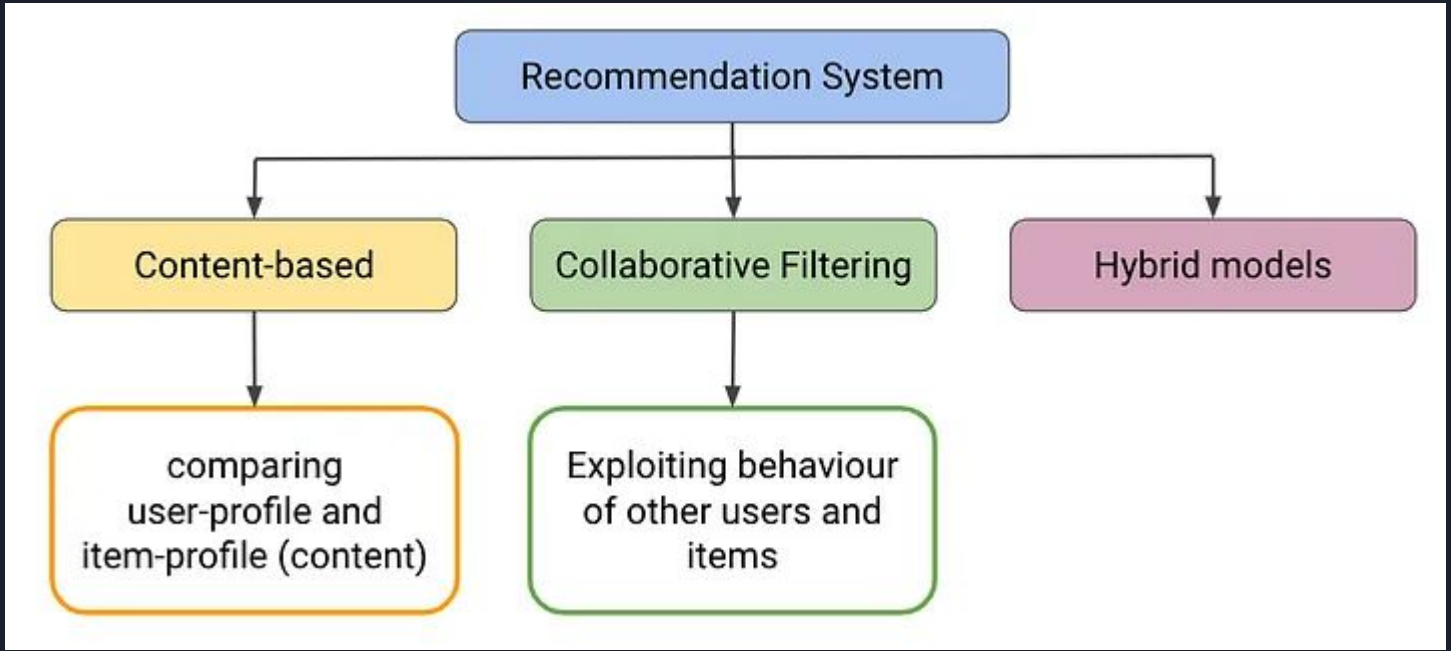


https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset

Type RecSys

All Type in RecSys

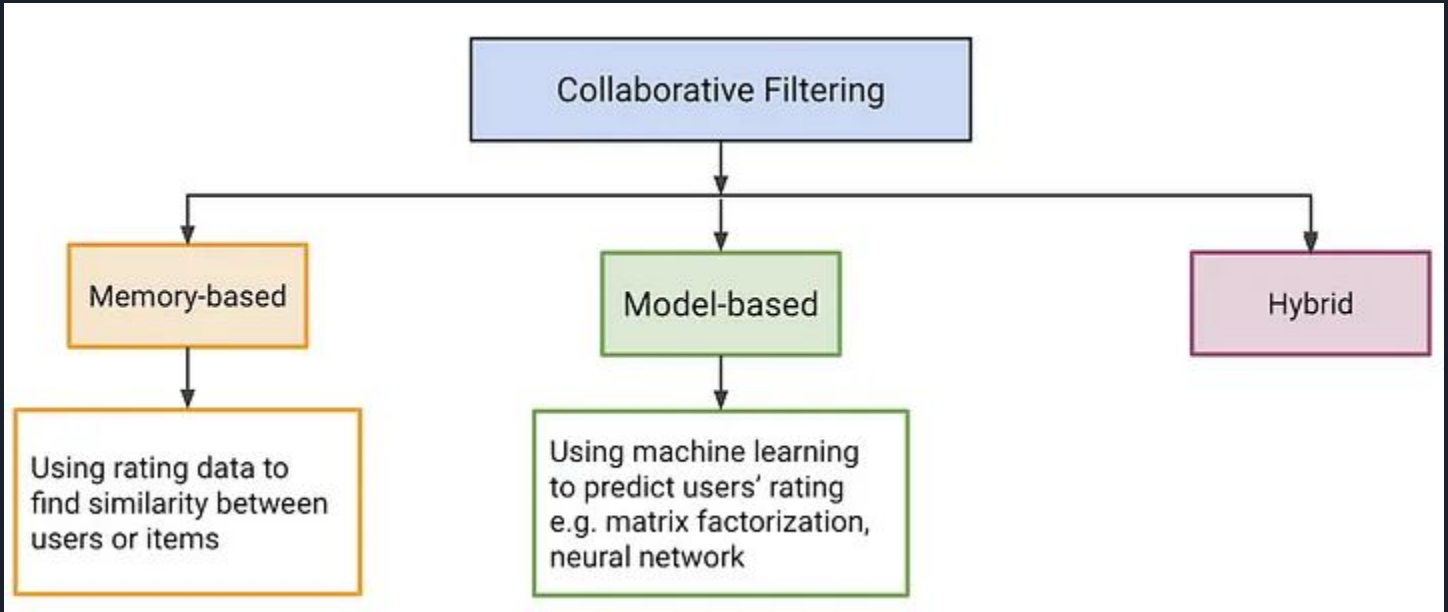
(It might be more than this that but I don't know yet.)



https://medium.com/@sinart.t/recommendation-system-%E0%B9%81%E0%B8%9A%E0%B8%9A%E0%B8%AA%E0%B8%A3%E0%B8%B8%E0%B8%9B%E0%B9%80%E0%B8%AD%E0%B8%B2%E0%B9%80%E0%B8%AD%E0%B8%87-ce6246f49754

All Type in RecSys

(It might be more than this that but I don't know yet.)

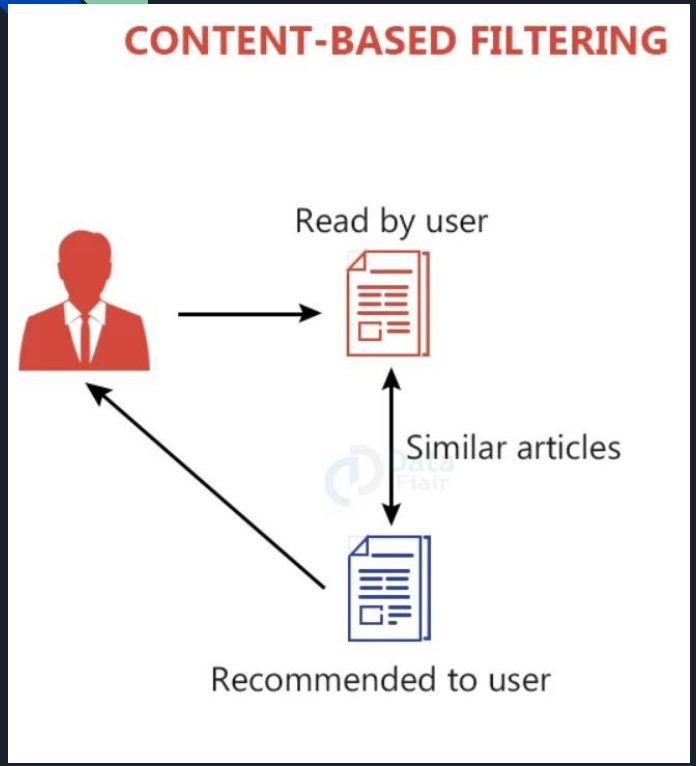


https://medium.com/@sinart.t/recommendation-system-%E0%B9%81%E0%B8%9A%E0%B8%9A%E0%B8%AA%E0%B8%A3%E0%B8%B8%E0%B8%9B%E0%B9%80%E0%B8%AD%E0%B8%B2%E0%B9%80%E0%B8%AD%E0%B8%87-ce6246f49754

Content base filtering

Prepare data to each type RecSys

Data to put into Content base



Dataframe preprocess from original data to use

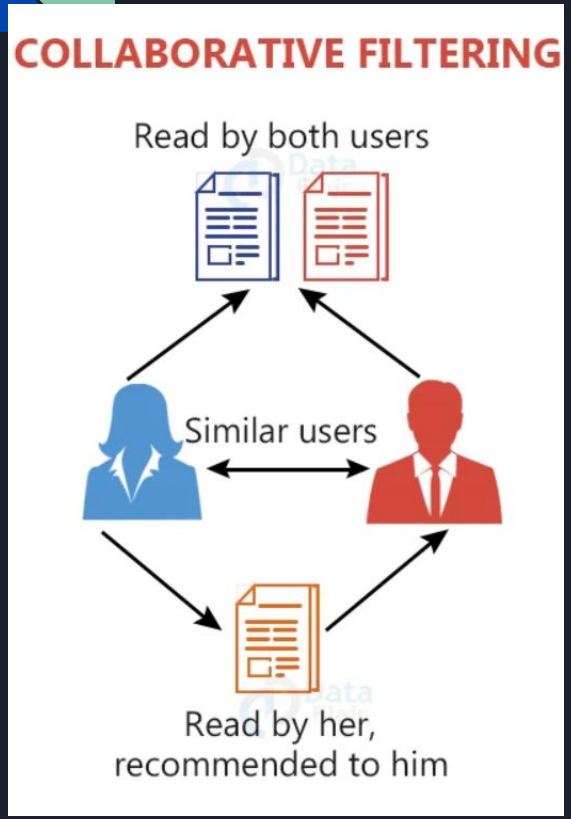
	Book-Title	Book-Author	Publisher	Image-URL-L
1	Clara Callan	Richard Bruce Wright	HarperFlamingo Canada	http://images.amazon.com/images/P,
3	Clara Callan	Richard Bruce Wright	HarperFlamingo Canada	http://images.amazon.com/images/P,
5	Clara Callan	Richard Bruce Wright	HarperFlamingo Canada	http://images.amazon.com/images/P,
8	Clara Callan	Richard Bruce Wright	HarperFlamingo Canada	http://images.amazon.com/images/P,
9	Clara Callan	Richard Bruce Wright	HarperFlamingo Canada	http://images.amazon.com/images/P,

Dataframe to use for content-base filtering

		User-ID	Book-Rating	all_features
0	.ZZZZZZZ.jpg	3,329	8	The Testament John Grisham Dell
1	.ZZZZZZZ.jpg	277,042	2	Wild Animus Rich Shapero Too Far
2	.ZZZZZZZ.jpg	1,376	8	Timeline MICHAEL CRICHTON Ballantine Books
3	.ZZZZZZZ.jpg	276,953	10	To Kill a Mockingbird Harper Lee Little Brown & Company
4	.ZZZZZZZ.jpg	277,922	6	The Street Lawyer JOHN GRISHAM Dell
5	.ZZZZZZZ.jpg	278,137	8	The Joy Luck Club Amy Tan Prentice Hall (K-12)

Collaborative filtering

Data to put into Collaborative filtering



book_user_rating head df

		Book-Author	User-ID	Book-Rating	unique_id_book
0		John Grisham	207,782	0	0
1	tion)	Toni Morrison	129,373	0	1
2	tion)	Toni Morrison	101,851	0	1
3	tion)	Toni Morrison	105,214	10	1
4	v: Why Not Learn from the Mistakes of Others? : You	J. R. Parrish	67,997	0	2

then pivot columns unique_id_book and index is User-ID using values = Book-Rating (fill NA with 0)

note: first column is User-ID and row of top column is unique_id_book

User-ID	0	1	2	3	4	5	6	7	8	9	10	11	12
139	0	0	0	0	0	0	0	0	0	0	0	0	0
254	0	0	0	0	0	0	0	0	0	0	0	0	0
388	0	0	0	0	0	0	0	0	0	0	0	0	0
602	0	0	0	0	0	0	0	0	0	0	0	0	0
625	0	0	0	0	0	0	0	0	0	0	0	0	0

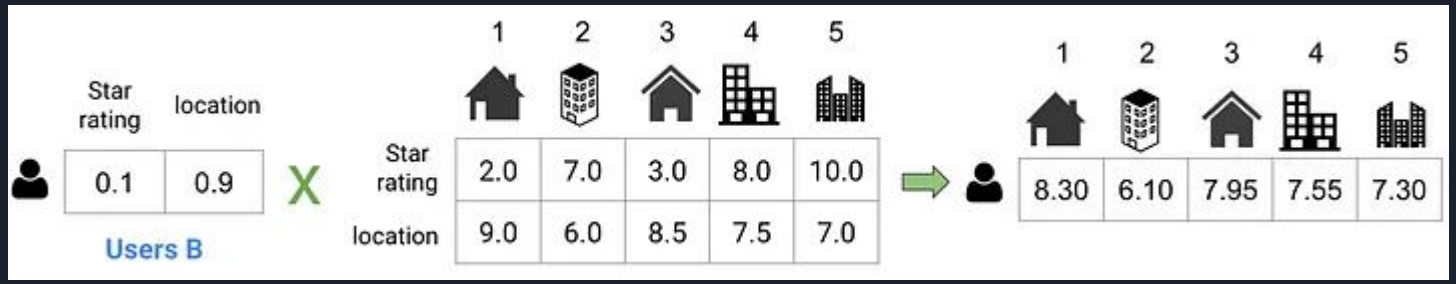
matrix rating

CF often uses Matrix Factorization (MF) techniques

$$24 = 12(\text{user}) \times 2(\text{item})$$

$$\text{matrix(rating)} = \text{matrix(u)} \times \text{matrix(i)}$$

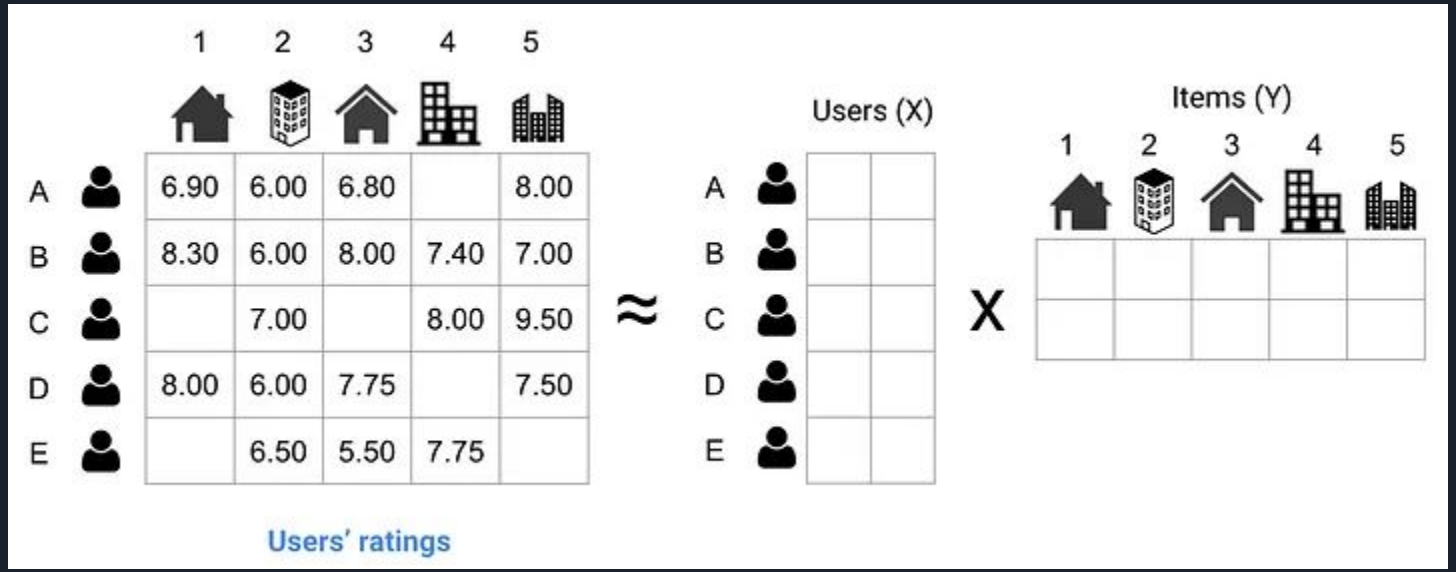
CF often uses Matrix Factorization (MF) techniques



จากตัวอย่างนี้ จะเห็นว่าโรงแรมที่จะได้คะแนนจากผู้ใช้เยอะ แปลได้ว่าโรงแรมมี features ที่ไปในทางเดียวกันกับ user weights

ในความเป็นจริง มันก็ไม่ได้หา feature ได้ชัดเจนตรงไปตรงมาขนาดนี้ ดังนั้นสำหรับ model-based ที่ใช้ จึงอยู่ในรูปของการพยายามใช้โมเดลในการหา user weight (หรือ user vector หรือ user embedding) และ item weight (หรือ item vector หรือ item embedding) นี้ โดยก็เพื่อให้เราสามารถประเมินความชอบ (เช่น rating score, relevance score) ของผู้ใช้ได้ใกล้เคียงกับความเป็นจริงที่สุด

Matrix Factorization



note: ที่ทำ ใช้ SVD model ในการแยก matrix ออกมา ซึ่งผมยังเข้าใจไม่ละเอียด เลยอธิบายยังไม่ครบทุกส่วนนะครับ

Neural Collaborative filtering

Data to put into NCF

	User-ID	ISBN	Book-Rating	↑ user_id	item_id	user_idx	item_idx
0	276,725	034545104X	0	104,433	57,188	0	0
1	276,726	0155061224	5	104,434	29,750	1	1
2	276,727	0446520802	0	104,435	107,392	2	2
3	276,729	052165615X	3	104,436	127,253	3	3
4	276,729	0521795028	6	104,436	127,287	3	4
5	276,733	2080674722	0	104,437	283,332	4	5
6	276,736	3257224281	8	104,438	290,525	5	6
7	276,737	0600570967	6	104,439	148,937	6	7
8	276,744	038550120X	7	104,440	83,283	7	8
9	276,745	342310538	10	104,441	292,989	8	9

Model NCF | Generalize Matrix Factorization (GMF)

```python
class GMF(nn.Module):
    def __init__(self, num_users, num_items, embedding_size):
        super(GMF, self).__init__()
        self.relu = nn.ReLU()
        self.user_embedding = nn.Embedding(num_users, embedding_size)
        self.item_embedding = nn.Embedding(num_items, embedding_size)
        self.fc = nn.Linear(embedding_size, 32)
        self.output_layer = nn.Linear(32, 1)
        self.dropout = nn.Dropout(0.2)

    def forward(self, user_ids, item_ids):
        user_embed = self.user_embedding(user_ids)
        item_embed = self.item_embedding(item_ids)
        element_product = user_embed * item_embed
        x = self.fc(element_product)
        x = self.relu(x)
        x = self.dropout(x)
        output = self.output_layer(x)
        output = torch.sigmoid(output)  # Ensure output is between 0 and 1
        return output.view(-1)
```

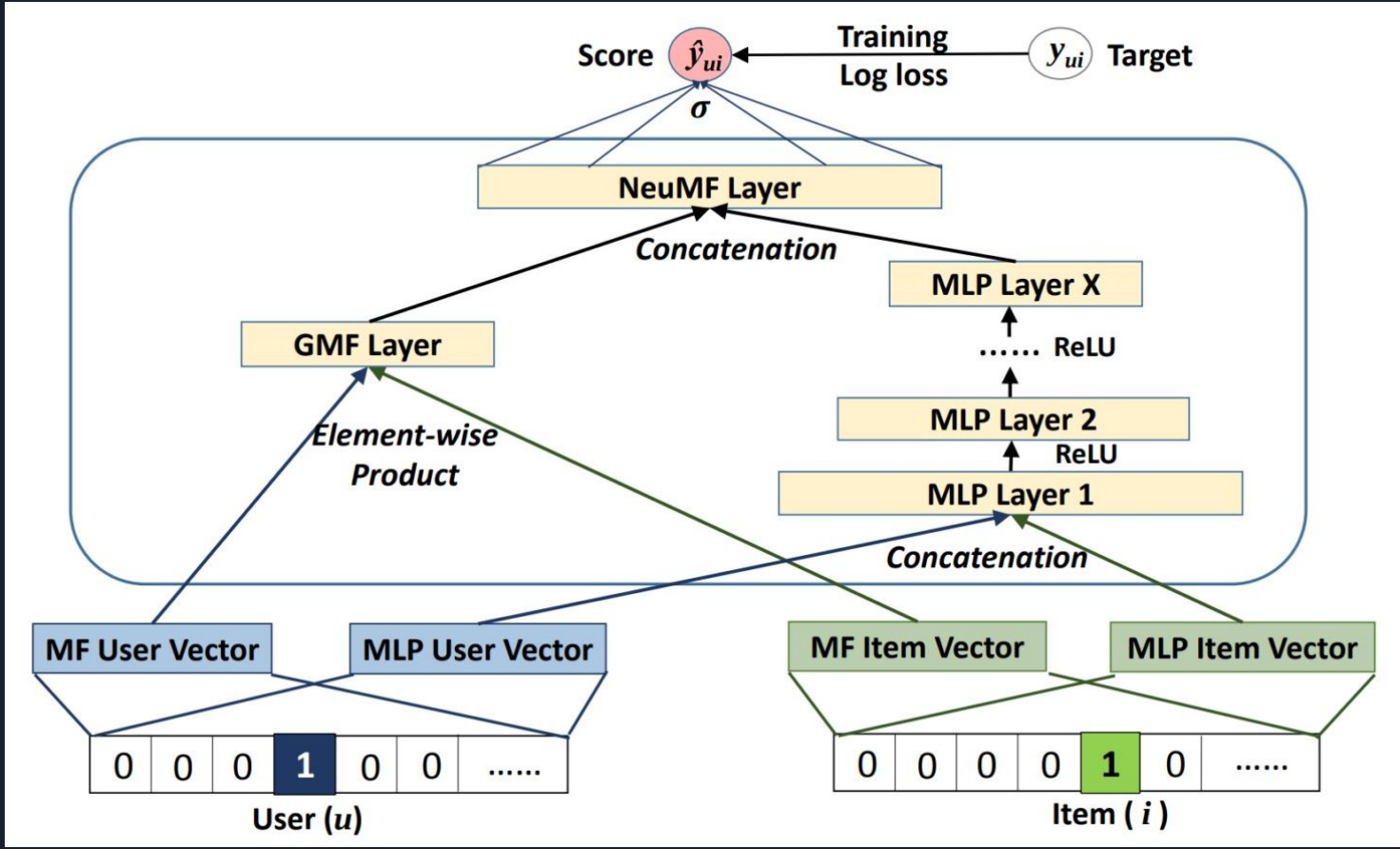
Model NCF | Multi Layer Perceptron (MLP)

```python
class MLP(nn.Module):
    def __init__(self, num_users, num_items, embedding_size, hidden_layers=[64, 32]):
        super(MLP, self).__init__()
        self.user_embedding = nn.Embedding(num_users, embedding_size)
        self.item_embedding = nn.Embedding(num_items, embedding_size)
        layers = []
        input_size = embedding_size * 2
        for hidden_size in hidden_layers:
            layers.append(nn.Linear(input_size, hidden_size))
            layers.append(nn.ReLU())
            layers.append(nn.Dropout(0.2))
            input_size = hidden_size
        layers.append(nn.Linear(hidden_layers[-1], 1))
        self.layers = nn.Sequential(*layers)

    def forward(self, user_ids, item_ids):
        user_embed = self.user_embedding(user_ids)
        item_embed = self.item_embedding(item_ids)
        concat_embed = torch.cat((user_embed, item_embed), dim=1)
        output = self.layers(concat_embed)
        output = torch.sigmoid(output)  # Ensure output is between 0 and 1
        return output.view(-1)
```

Model NCF

```
class NCF(nn.Module):
    def __init__(self, gmf_model, mlp_model):
        super(NCF, self).__init__()
        self.gmf = gmf_model
        self.mlp = mlp_model

    def forward(self, user_ids, item_ids):
        gmf_output = self.gmf(user_ids, item_ids)
        mlp_output = self.mlp(user_ids, item_ids)
        combined_output = (gmf_output + mlp_output) / 2
        return combined_output
```

Model NCF | Architectur



Streamlit

main

EDA Book datasets

Book recommendations content base

Book recommendations collaborative filtering

Book recommendations ncf

Rating visualization DR

Select User ID

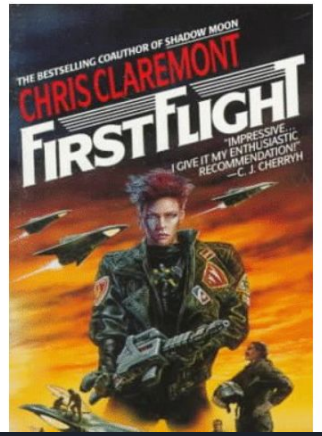
276755

Insert a number to show books

5

Book name: Firstflight

Author: Chris Claremont



ML Prediction: Clustering group of user