Recommender Systems

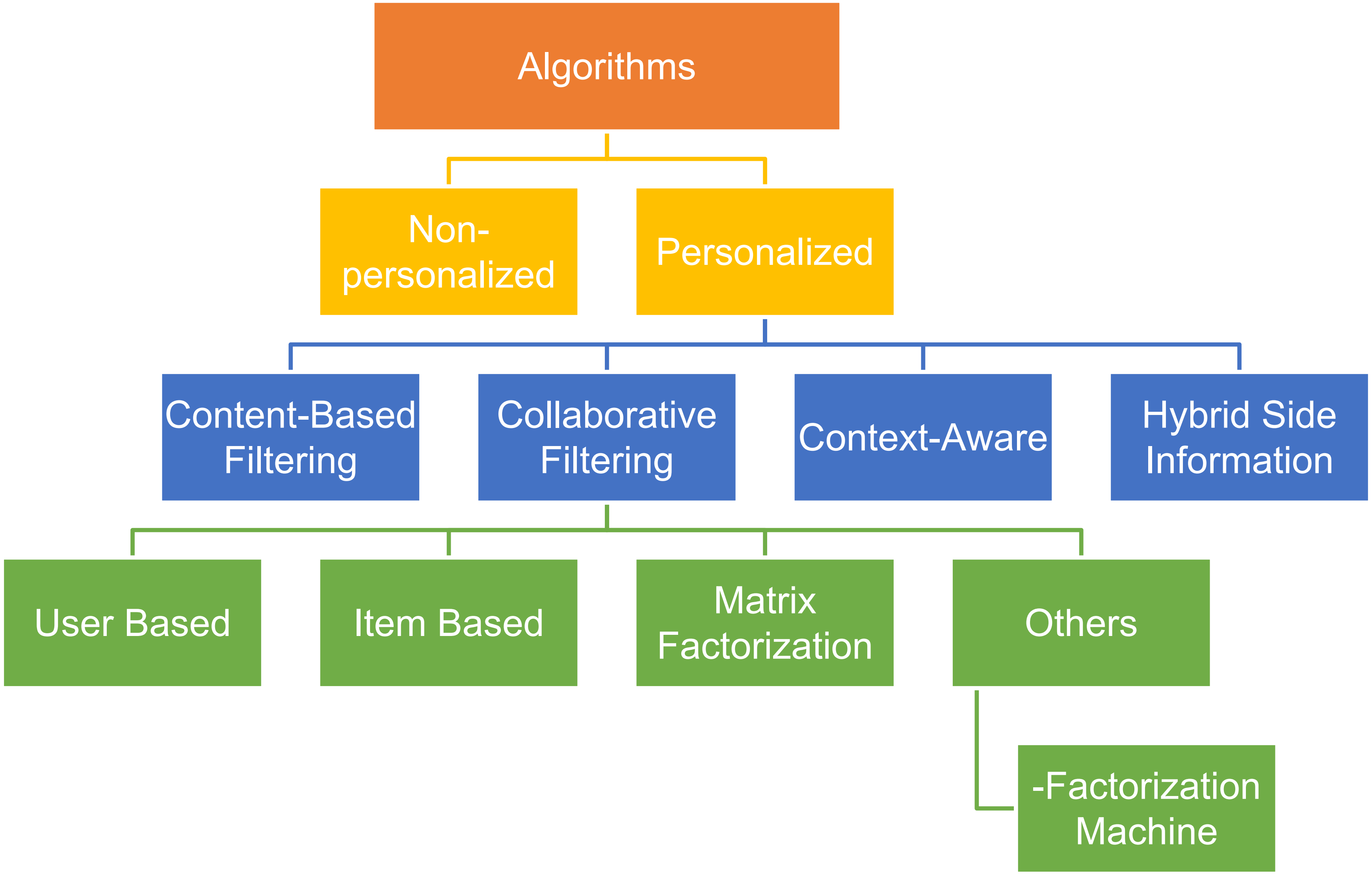
- Taxonomy of Recommender Systems
- Item-context and User Rating Matrix
- Quality of Recommender Systems with Online/Offline Evaluation Techniques
- Filtering Types and Techniques
- Important Concepts of Recommender Systems

Shahzaib Hamid
AI Sciences Instructor

Overview

- Taxonomy of Recommender Systems
- Item-context Matrix
- User-Rating Matrix
- Inferring Preferences
- Quality of Recommender Systems
- Online and Offline Evaluation Techniques
- Dataset Partitioning
- Overfitting
- Error Matrix
- Content-based Filtering
- Collaborative Filtering
- User-based and Item-based Collaborative Filtering
- Model-based and Memory-based Collaborative Filtering

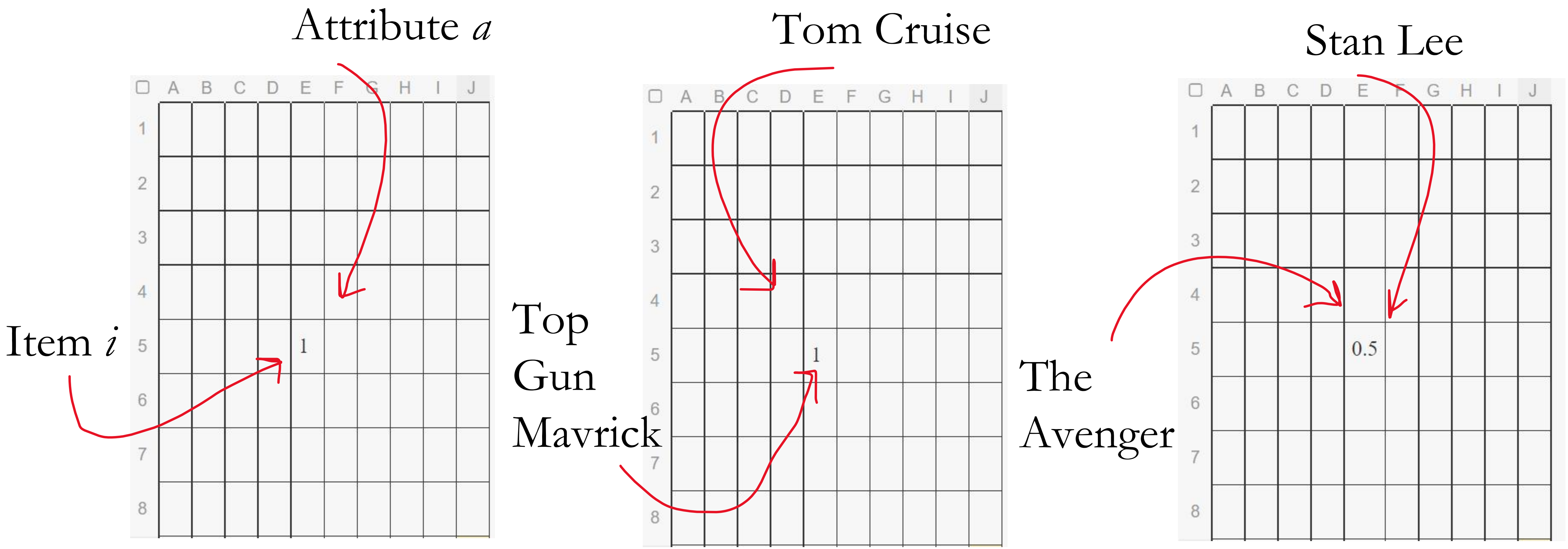
Taxonomy Recommender Systems



Item-Context Matrix (ICM)

- One important input to a recommender system are the list of items and their attributes
- ICM is used to describe such problems
- Rows in the item content matrix represent items and columns represent attributes
- The values in the item content matrix are in binary format, either zero or one
- If an item contains a specific attribute, the corresponding value in the matrix will be set to one, zero, or otherwise
- Let's have an Example

Item-Context Matrix (ICM) - Example



User-Rating Matrix (URM)

Item i

User u

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5					5					
6										
7										
8										

R= u i

	A	B	C	D	E	F	G	H	I	J
1	0						0			
2										
3									0	
4							0			
5					5					
6			0						0	
7										
8										

$r_{ui} \in \{0,1\} \leftarrow$ implicit

$r_{ui} \in \{1,2,3,4,5\} \leftarrow$ explicit

Typically, URM density <0.01 %

Netflix, URM density ≈0.002 %

MovieLens, URM density ≈0.005 %

5 is the rating which user gave to the item

Inferring Preferences

- There are different ways to collect user opinion

Implicit Ratings

Explicit Ratings

- Rating Scale
- Even Rating Scale
- Odd Rating Scale
- Rating Distribution

Quality of Recommending System

- Quality of RS
- Topic Diversity
- Novelty
- Serendipity
- Temporal Diversity
- Temporal Awareness (Reactive; Trending)
- Temporal Awareness (Reactive; Seasonal)
- Temporal Awareness (Proactive)
- Location Awareness
- Risk Awareness
- Context Awareness
- Critique Acceptance
- Curation
- Knowledge Based
- Social Awareness
- Privacy
- Protection from Malicious Voting

Evaluation Techniques for Recommender Systems

Online Evaluation Techniques

Direct User Feedback

- Simple questionnaires or surveys are used to ask from user
- It has two problems
  - Size of the sample should be meaningful
  - Opinion of the use could not be reliable

A/B Testing

- Behavior of the user is analyzed online, and A/B testing is applied
- User behavior is compared

Controlled Experiments

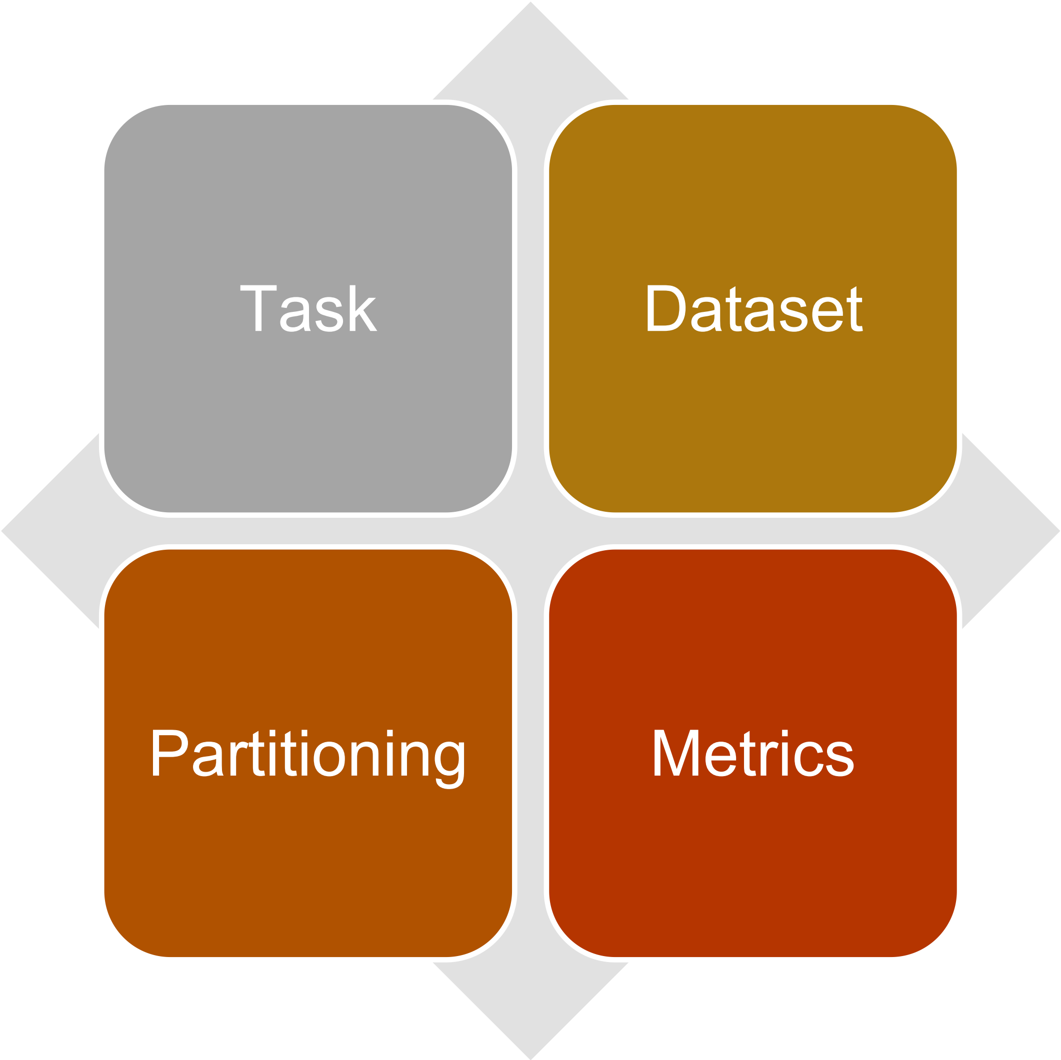
- A mockup application is made available to the users.
- Issue: User cannot be the real user.

Crowed Sourcing

- It asks people to volunteer to test and review answers by giving them compensations

Offline Evaluation Techniques

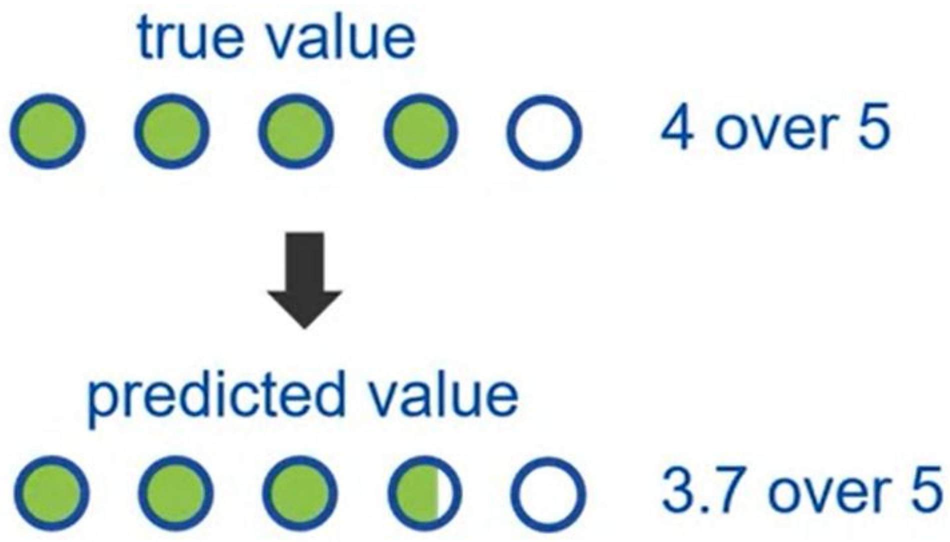
It has few important aspect



Offline Evaluation: Task

Rating Prediction

- Goals is go to as near as to the true value



Offline Evaluation: Task

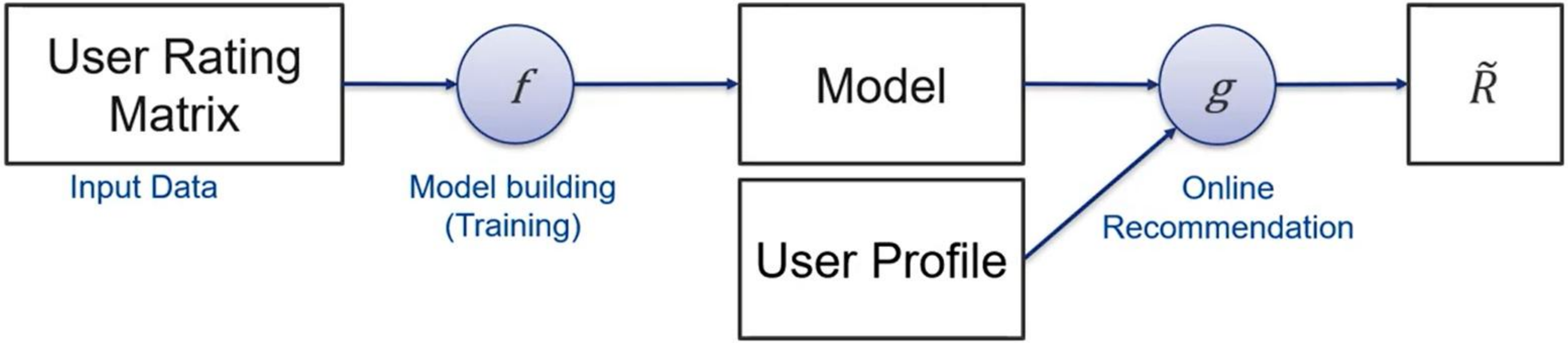
Top N-Recommendation

Offline Evaluation: Data

- Data represents all the information available
- Dataset is always represented by URM
- We know very limited information of all ratings
- Ratings are classified into part
  - Relevant
  - Non-relevant
- Dataset also contains unknown ratings

Data Partitioning

- Model= f (URM)
- Estimated Ratings = g(Model, user profile)



Data Partitioning

- Model= f (URM)
- Estimated Ratings = g(Model, user profile)

- Example:
- Model = Top Gun is like The Avengers
- user profile = Timmy likes Top Gun
- Recommendation ↔ True opinion of the user

Data Partitioning (Hold Out of Rating)

- Model=(X)
- Estimated Ratings = (Model, Y)
- Estimated Ratings ↔ Z

	A	B	C	D	E	F	G
1	1						
2			1			1	
3	1						1
4				1		1	
5	1						
6		1				1	
7			1				

Data Partitioning (Hold Out of Rating)

- Model=(X)
- Estimated Ratings = (Model, Y)
- Estimated Ratings ↔ Z

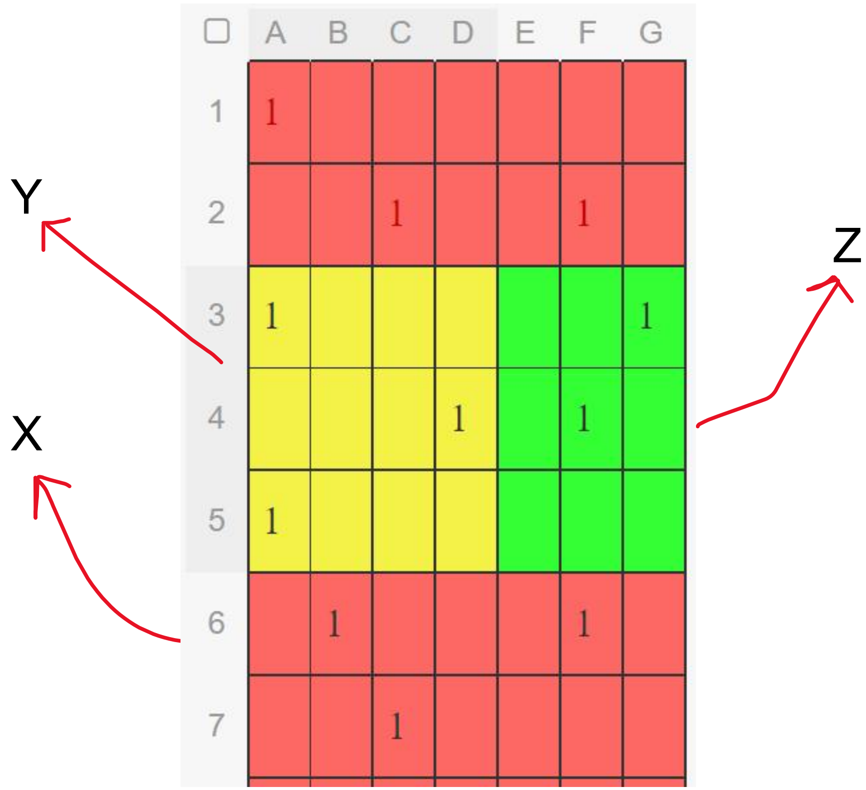
- Z = testing (hidden ratings)

	A	B	C	D	E	F	G
1	1						
2			1			1	
3	1						1
4				1		1	
5	1						
6		1				1	
7			1				

Z

Data Partitioning (Hold Out of Rating)

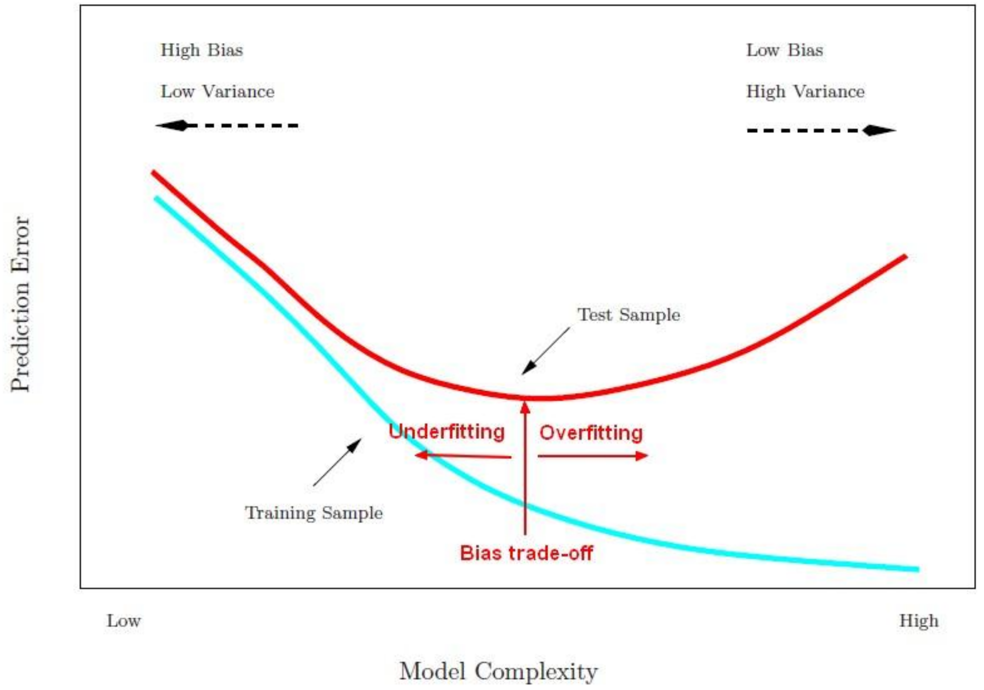
- Model=(X)
- Estimated Ratings = (Model, Y)
- Estimated Ratings ↔ Z

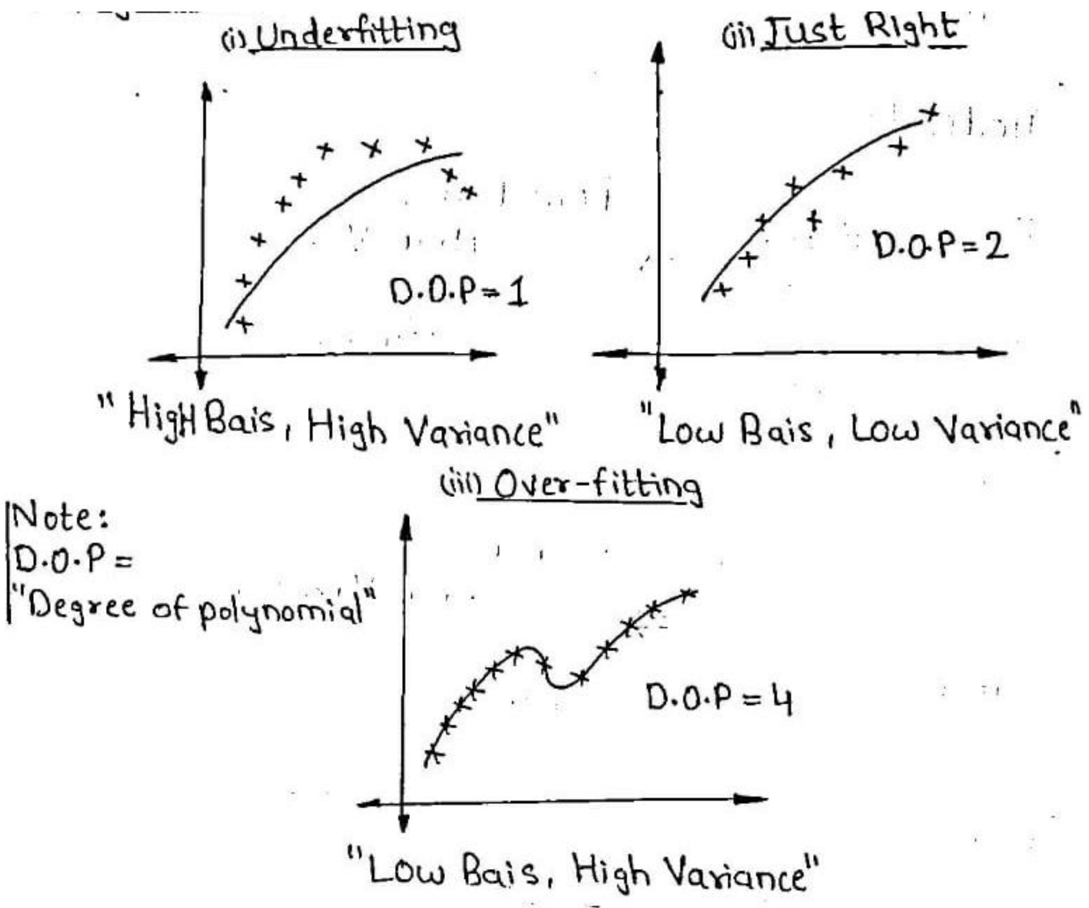


- Z = testing (hidden ratings)
- X = training
- Y = testing (user profile)
- Y belongs to X

Important Parameters in Recommender Systems

- Bias
- Variance
- Underfitting
- Overfitting

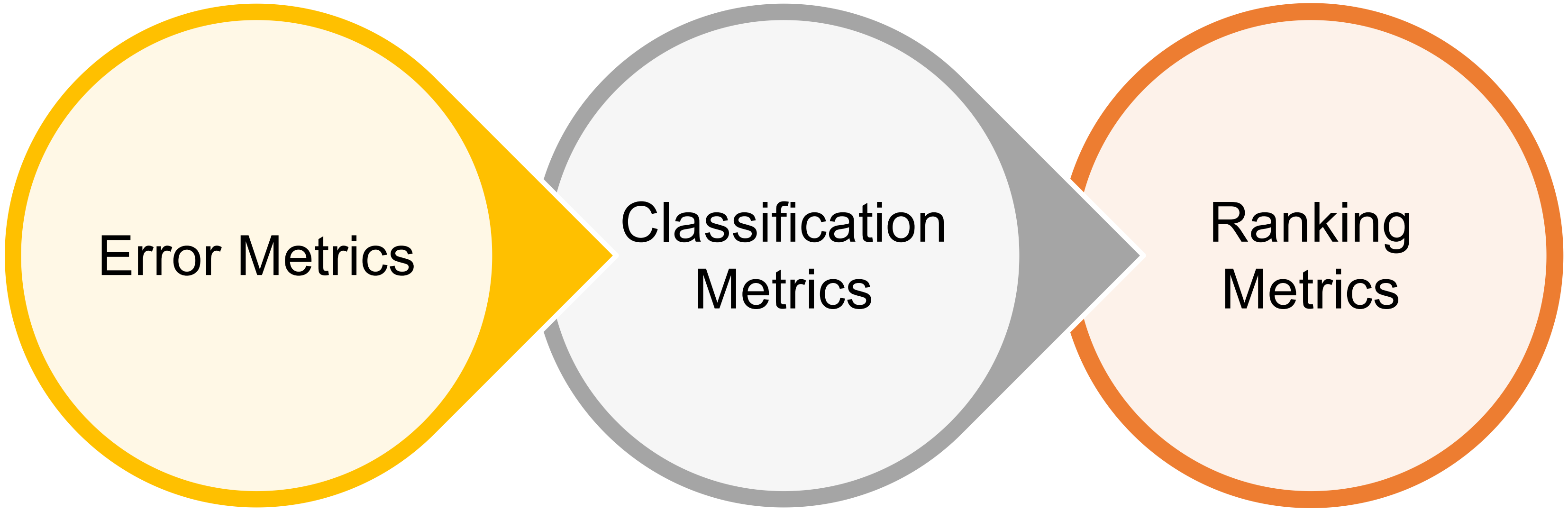




(i) Model 1 (Underfitting): Train Accuracy ↓
Test Accuracy ↓

(ii) Model 3 (Overfitting): Train Accuracy ↑
Test Accuracy ↓

Quality Metrix in the Recommender Systems



- Classification and Ranking Matrix will be discussed in Next Module

Error Matrix

true value r_{ui}

4 over 5

estimated value $\hat{r}_{ui}$

3.7 over 5

$$e_{ui} = |r_{ui} - \hat{r}_{ui}| = |4 \ - 3.7| = 0.3$$

$\hat{r}_{ui}$: rating estimated by the recommender system
r_{ui}: true rating in the test set
e_{ui}: error

Error Matrix

Mean absolute error:

$$MAE = \frac{\sum_{u,i\in T} |r_{ui} - \hat{r}_{ui}|}{N_T}$$

Mean squared error:

$$MSE = \frac{\sum_{u,i\in T} (r_{ui} - \hat{r}_{ui})^2}{N_T}$$

$\hat{r}_{ui}$: rating estimated by recommender system
r_{ui}: true rating in the test set
T: test set
N_T: number of interactions in the test set (non-zero ratings)

Content-Based Filtering

- Content-Based recommender system tries to guess the features or behavior of a user given the item's features, he/she reacts positively to.

Movies	User 1	User 2	User 3	User 4	Action	Comedy
Item 1	1		4	5	Yes	No
Item 2	5	4	1	2	No	Yes
Item 3	4	4		3	Yes	Yes
Item 4	2	2	4	4	No	Yes

- We can know which users like which genre, as a result, we can obtain features corresponding to that user, depending on how he/she reacts to movies of that genre.
- Content-based filtering does not require other users' data during recommendations to one user.

Pros & Cons: Content-Based Filtering

Pros:

- The model doesn't need any data about other users, since the recommendations are specific to this user. This makes it easier to scale to a large number of users
- The model can capture the specific interests of a user, and can recommend niche items that very few other users are interested in.

Cons:

- Since the feature representation of the items are hand-engineered to some extent, this technique requires a lot of domain knowledge. Therefore, the model can only be as good as the hand-engineered features.
- The model can only make recommendations based on existing interests of the user. In other words, the model has limited ability to expand on the users' existing interests.

Collaborative Filtering

- Collaborative filtering filters information by using the interactions and data collected by the system from other users.
- It's based on the idea that people who agreed in their evaluation of certain items are likely to agree again in the future.
- Most collaborative filtering systems apply the so-called similarity index-based technique.
- Collaborative-filtering systems focus on the relationship between users and items.
- The similarity of items is determined by the similarity of the ratings of those items by the users who have rated both items.

User-Based Collaborative Filtering

- User-Based Collaborative Filtering is a technique used to predict the items that a user might like on the basis of ratings given to that item by the other users who have similar taste with that of the target user.
- A specific application of this is the user-based Nearest Neighbor algorithm.

Steps to Compute it:

- Finding the similarity of users to the target user U.
- Prediction of missing rating of an item

User-Based Collaborative Filtering

Pros

- Easy to implement.
- Context independent.
- Compared to other techniques, such as content-based, it is more accurate.

Cons

- **Sparsity:** The percentage of people who rate items is really low.
- **Scalability:** The more K neighbors we consider (under a certain threshold), the better my classification should be. Nevertheless, the more users there are in the system, the greater the cost of finding the nearest K neighbors will be.
- **Cold-start:** New users will have no to little information about them to be compared with other users.
- **New item:** Just like the last point, new items will lack of ratings to create a solid ranking.

Item-Based Collaborative Filtering

- It was first invented and used by Amazon in 1998.
- Rather than matching the user to similar customers, item-to-item collaborative filtering matches each of the user's purchased and rated items to similar items, then combines those similar items into a recommendation list.

Steps to Compute it:

- Firstly, similarities between items are computed.
- Secondly, based on the computed similarities, items similar to already consumed/rated are looked at and recommended accordingly.

Model-Based Collaborative Filtering

- Model-based collaborative filtering provides recommendations by developing a model from user ratings.
- Model-based filtering can also use implicit information by observing the habits of users, such as music played, applications downloaded, websites visited, or books read.
- To develop a model, there are two approaches that can be used, which are probability approach or rating prediction
- The modeling process is conducted by machine learning techniques such as classification, clustering, and rule-based approach
- However, model-based approach requires a great resource, such as time and memory, to develop the model and may lose information when using dimensionality reduction

Memory-Based Collaborative Filtering

- Memory-based collaborative filtering utilizes the entire user-item data to generate predictions.
- The system uses statistical methods to search for a set of users who have similar transactions history to the active user.
- This method is also called nearest-neighbor or user-based collaborative filtering.

Three processes can be applied on it

- Choosing other users that are similar to a user
- Predicting rating of the item i to a user by calculating the results of aggregating similar users
- Providing recommendations based on the results predicted