بسم الله الرحمن الرحیم

Information Technology Fundamentals

Mohammad Hossein Manshaei

manshaei@gmail.com

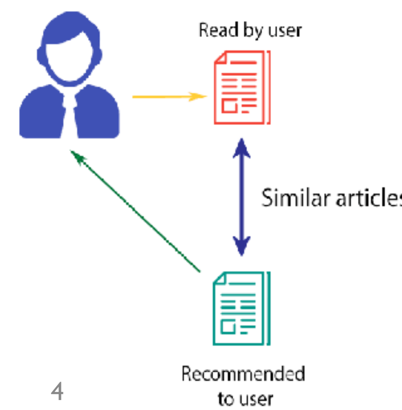
Web Systems: Recommender Systems
Module 5: Part 7

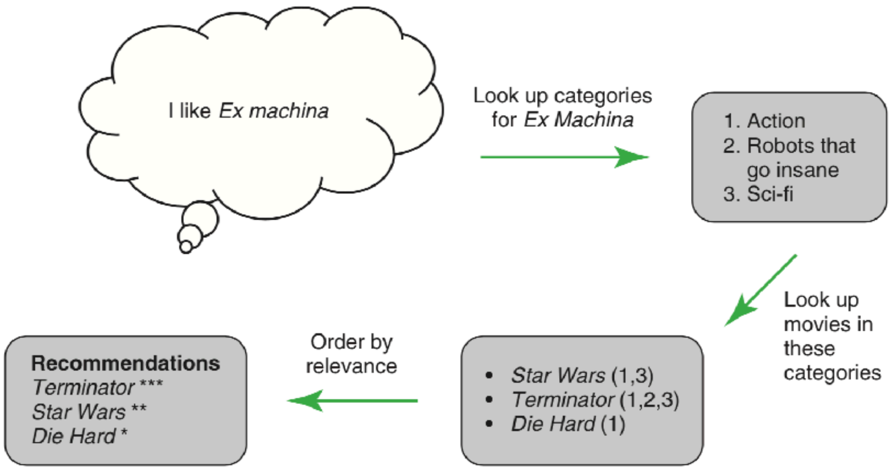
Contents

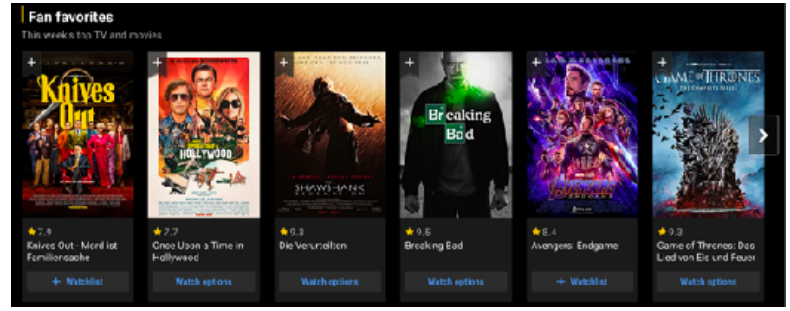
- ➢ Collaborative Filtering Recommendation
  - ✓ Memory-Based:
    - User-Based and Item-Based
  - ✓ Model-Based
    - Matrix Factorization/latent factor models and Association Rules
- ➢ Content-Based Recommendation
- ➢ Knowledge Based Recommendation
- ➢ Demographic Recommendation
- ➢ Hybrid Recommendation Systems
- ➢ RS Evaluation
  - ✓ Online and Offline Evaluations
  - ✓ Design Issues and RS Properties

Content-Based Filtering (CBF)

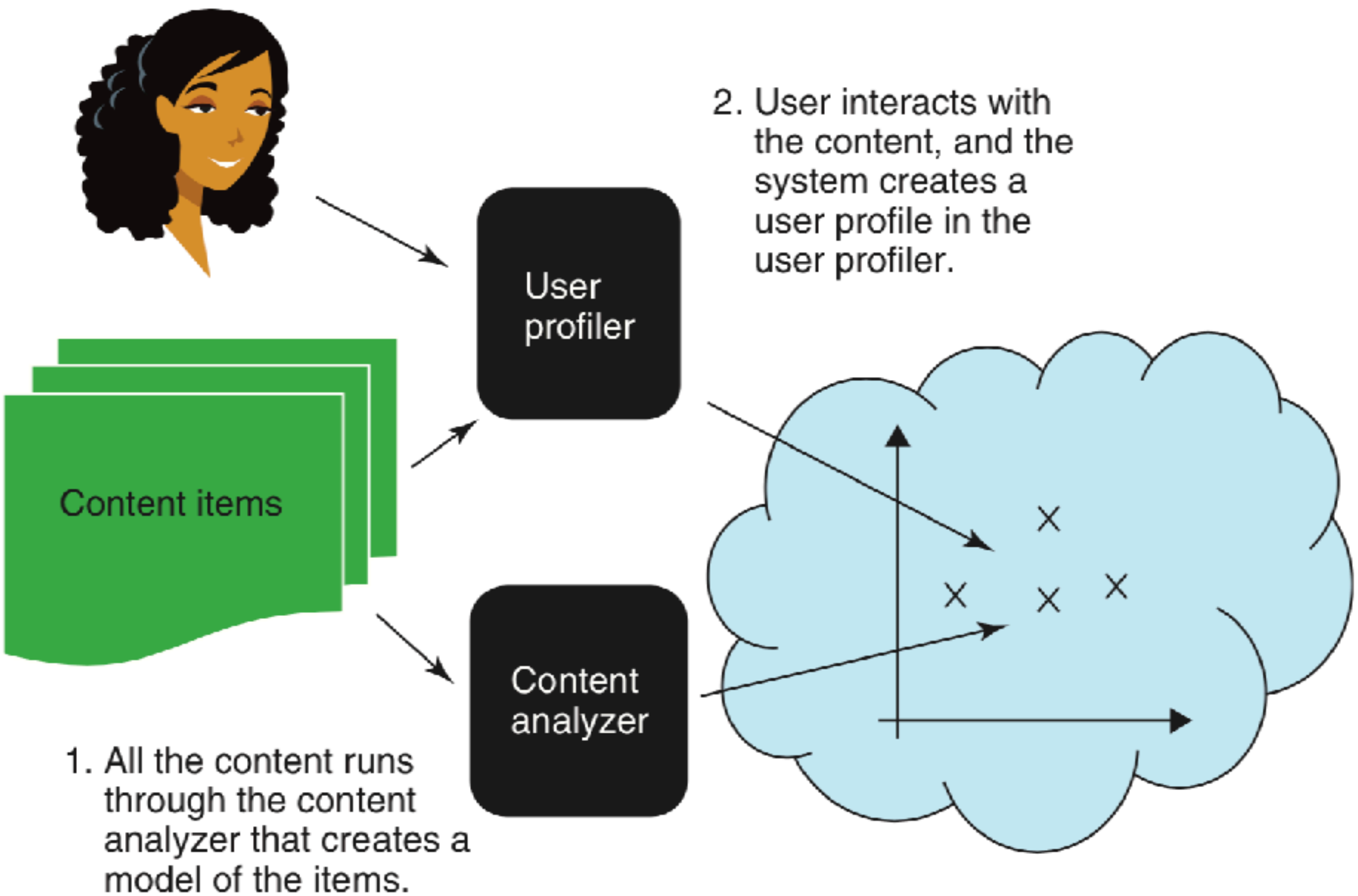
- Recommendation is made based on the user profiles using features extracted from the content of the items the user has evaluated in the past.
- CBF uses different types of models to find similarity between documents:
  1. **Vector Space Model** such as Term Frequency Inverse Document Frequency (TF/IDF) or
  2. **Probabilistic Models** such as Naive Bayes Classifier, Decision Trees or Neural Networks







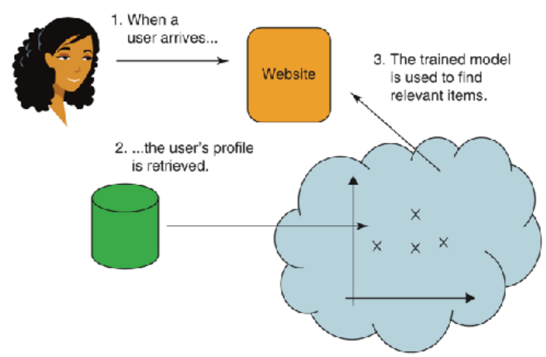
Example of content-based recommendation pipeline



Training a content-based recommender offline

Item Retriever with Content-based Filtering Online

- **Content Analyzer**—Creates a model based on the content. In a way, it creates a profile for each item. It's where the training of the model is done.
- **User Profiler**—Creates a user profile; sometimes the user profile is a simple list of items consumed by the user.
- **Item Retriever**—Retrieves relevant items found by comparing the user profiles to the item profiles as shown in this figure. If the user profile is a list of items, this list is iterated, and similar items are found for each item in the user's list.



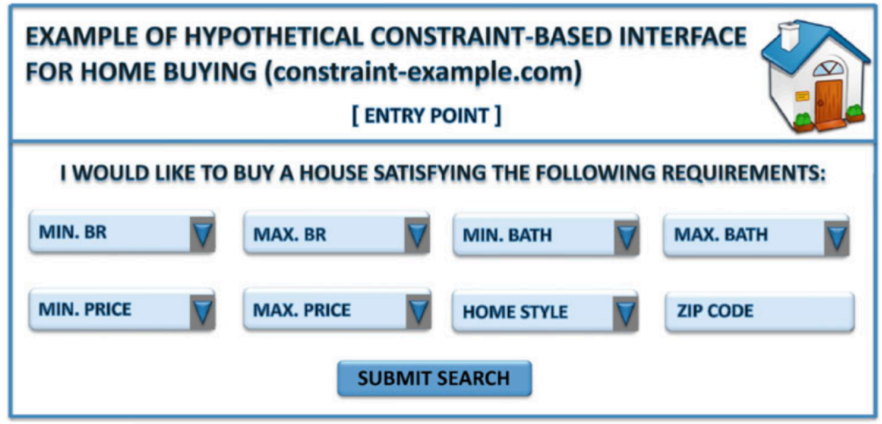
Contents

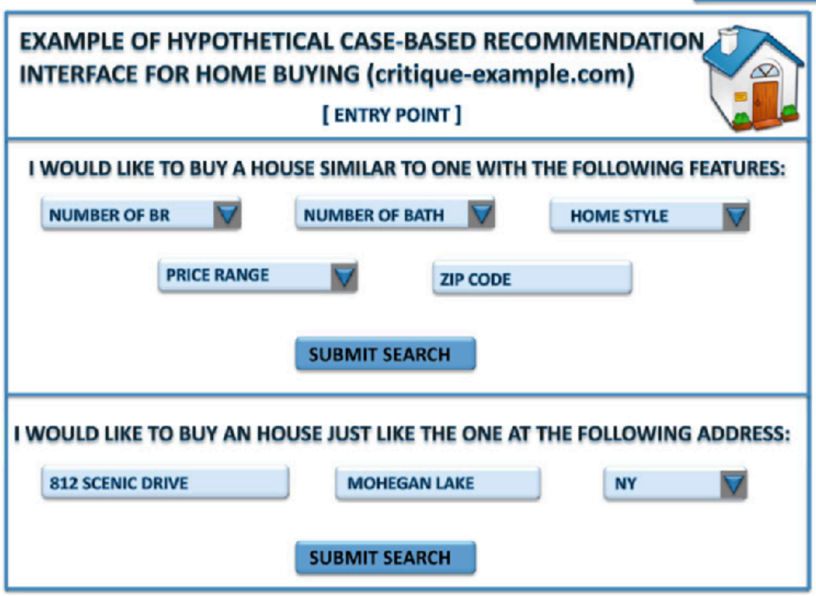
- ➢ Collaborative Filtering Recommendation
  - ✓ Memory-Based:
    - User-Based and Item-Based
  - ✓ Model-Based
    - Matrix Factorization/latent factor models and Association Rules
- ➢ Content-Based Recommendation
- ➢ Knowledge Based Recommendation
- ➢ Demographic Recommendation
- ➢ Hybrid Recommendation Systems
- ➢ RS Evaluation
  - ✓ Online and Offline Evaluations
  - ✓ Design Issues and RS Properties

Knowledge-Based Recommendation

- The recommendation process of knowledge-based recommender applications is highly interactive, a foundational property that is a reason for their characterization as conversational systems.
- It makes recommendations based not on a user's rating history, but on **specific queries made by the user**
- Two basic types of knowledge-based recommender systems are:

1. Constraint-based
2. Case-based systems


EXAMPLE OF HYPOTHETICAL CONSTRAINT-BASED INTERFACE FOR HOME BUYING (constraint-example.com)
[ENTRY POINT]
I WOULD LIKE TO BUY A HOUSE SATISFYING THE FOLLOWING REQUIREMENTS:
MIN. BR
MAX. BR
MIN. BATH
MAX. BATH
MIN. PRICE
MAX. PRICE
HOME STYLE
ZIP CODE
SUBMIT SEARCH


EXAMPLE OF HYPOTHETICAL CASE-BASED RECOMMENDATION INTERFACE FOR HOME BUYING (critique-example.com)
[ENTRY POINT]
I WOULD LIKE TO BUY A HOUSE SIMILAR TO ONE WITH THE FOLLOWING FEATURES:
NUMBER OF BR
NUMBER OF BATH
HOME STYLE
PRICE RANGE
ZIP CODE
SUBMIT SEARCH
I WOULD LIKE TO BUY AN HOUSE JUST LIKE THE ONE AT THE FOLLOWING ADDRESS:
812 SCENIC DRIVE
MOHEGAN LAKE
NY
SUBMIT SEARCH

Example

Interactivity in Knowledge-Based Recommender Systems

1. **Conversational Systems**: The user preferences are determined iteratively in the context of a feedback loop.
2. **Search-Based Systems**: The user preferences are elicited by using a preset sequence of questions such as the following: "Do you prefer a house in a suburban area or within the city?"
3. **Navigation-Based Recommendation**: The user specifies a number of change requests to the item being currently recommended. Through an iterative set of change requests, it is possible to arrive at a desirable item.

Approach	Conceptual Goal	Input
Collaborative	Give me recommendations based on a collaborative approach that leverages the ratings and actions of my peers/myself.	User ratings + community ratings
Content-based	Give me recommendations based on the content (attributes) I have favored in my past ratings and actions.	User ratings + item attributes
Knowledge-based	Give me recommendations based on my explicit specification of the kind of content (attributes) I want.	User specification + item attributes + domain knowledge

Paradigm	User profile and contextual parameters	Community data	Product features	Knowledge models
Collaborative	Yes	Yes	No	No
Content-based	Yes	No	Yes	No
Knowledge-based	Yes	No	Yes	Yes

The conceptual goals of various recommender systems.
Input data requirements of recommendation algorithms.

Contents

- ➢ Collaborative Filtering Recommendation
  - ✓ Memory-Based:
    - User-Based and Item-Based
  - ✓ Model-Based
    - Matrix Factorization/latent factor models and Association Rules
- ➢ Content-Based Recommendation
- ➢ Knowledge Based Recommendation
- ➢ Demographic Recommendation
- ➢ Hybrid Recommendation Systems
- ➢ RS Evaluation
  - ✓ Online and Offline Evaluations
  - ✓ Design Issues and RS Properties

Demographic Recommender Systems

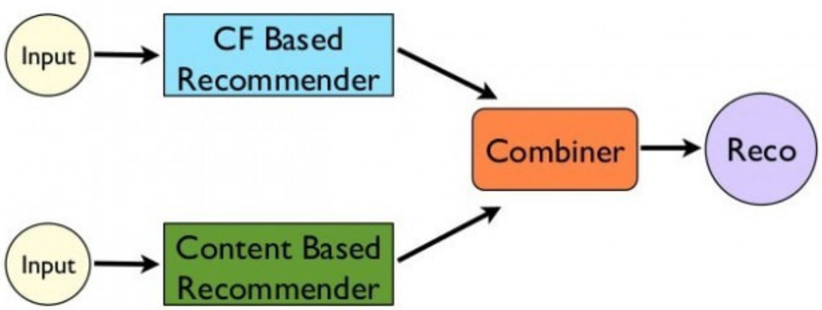
- The demographic information about the user is leveraged to learn classifiers that can map specific demographics to ratings or buying propensities.
- Based on the **demographic profile of the user.**
- For example, users are dispatched to particular websites based on their **language** or **country**. Or, suggestions may be customized according to the **age of the user**.
- While these approaches have been quite popular in the marketing literature, there has been relatively little proper RS research on demographic systems.

Contents

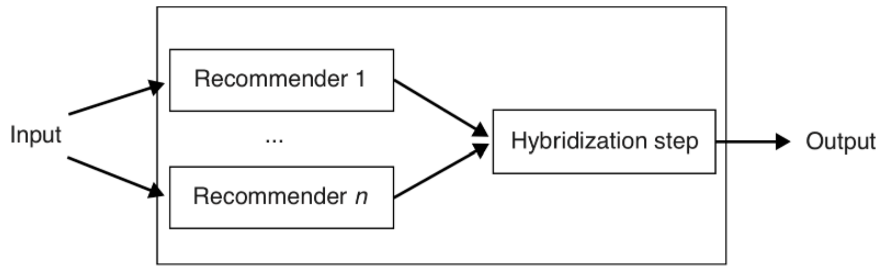
- ➢ Collaborative Filtering Recommendation
  - ✓ Memory-Based:
    - User-Based and Item-Based
  - ✓ Model-Based
    - Matrix Factorization/latent factor models and Association Rules
- ➢ Content-Based Recommendation
- ➢ Knowledge Based Recommendation
- ➢ Demographic Recommendation
- ➢ Hybrid Recommendation Systems
- ➢ RS Evaluation
  - ✓ Online and Offline Evaluations
  - ✓ Design Issues and RS Properties

Hybrid Recommendation system

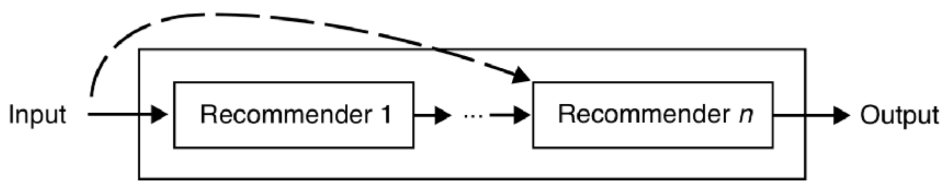
- Combines different recommendation techniques in order to gain better system optimization to avoid some limitations and problems of pure recommendation systems.
- A combination of algorithms will provide more accurate and effective recommendations than a single algorithm as the disadvantages of one algorithm can be overcome by another algorithm.



Two Designs for Hybrid Recommender Systems



Parallelized hybridization design



Pipelined hybridization design.

Contents

- ➢ Collaborative Filtering Recommendation
  - ✓ Memory-Based:
    - User-Based and Item-Based
  - ✓ Model-Based
    - Matrix Factorization/latent factor models and Association Rules
- ➢ Content-Based Recommendation
- ➢ Knowledge Based Recommendation
- ➢ Demographic Recommendation
- ➢ Hybrid Recommendation Systems
- ➢ RS Evaluation
  - ✓ Online and Offline Evaluations
  - ✓ Design Issues and RS Properties

"True genius resides in the capacity for evaluation of uncertain, hazardous, and conflicting information."
Winston Churchill

RS Evaluation

The quality of a recommendation algorithm can be evaluated using different types of measurement which can be accuracy or coverage. The type of metrics used depends on the type of filtering technique.

1. Decision support accuracy metrics
2. Statistical accuracy metrics

Types of Evaluation

- Methods for choosing the best technique based on the specifics of the application domain, identifying influential success factors behind different techniques, or comparing several techniques based on an optimality criterion are all required for effective evaluation research.
- Recommender systems can be evaluated using either **online methods** or **offline methods**.

Online Evaluation

- In an online system, the user reactions are measured with respect to the presented recommendations.
- Therefore, user participation is essential in online systems.
- A typical metric:
  - The **conversion rate** measures the frequency with which a user selects a recommended item.

Example of Online Evaluation

- In a **news recommender system**, one might compute the fraction of times that a user selects a recommended article
- If desired, expected **costs or profits** can be added to the items to make the measurement sensitive to the importance of the item
- These methods are also referred to as **A/B testing**, and they measure the direct impact of the recommender system on the end user.

A/B Testing

- The basic idea in these methods is to compare t**wo algorithms** as follows:
  - Segment the users into two groups A and B.
  - Use one algorithm for group A and another algorithm for group B for a period of time, while keeping all other conditions (e.g., selection process of users) across the two groups as similar as possible.
  - At the end of the process, compare the conversion rate (or other payoff metric) of the two groups.

Challenge of Online Evaluation

- Require active user participation, it is often not feasible to use them in benchmarking and research.
- Significant challenges in **gaining access to user conversion data** from systems with large-scale user participation.
- Even if such access is gained, it is usually specific to a **single large-scale system** (Not from different Types and Domains).
- Testing over **multiple data sets** is particularly important for assuring greater generalization power of the recommender system so that one can be assured that the algorithm works under a variety of settings.

Offline Evaluation

Offline methods are, by far, the most common methods for evaluating recommender systems from a **research** and **practice perspective**.

Accuracy (Statistical)

The system generates predicted ratings $\hat{r}_{ui}$ for a test set T of user-item pairs (u, i) for which the true ratings r_{ui} are known.

$$\mathrm{MAE} = \frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} |\hat{r}_{ui} - r_{ui}|$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} (\hat{r}_{ui} - r_{ui})^2}$$

Prediction (Decision Support)

- Measuring Usage Prediction

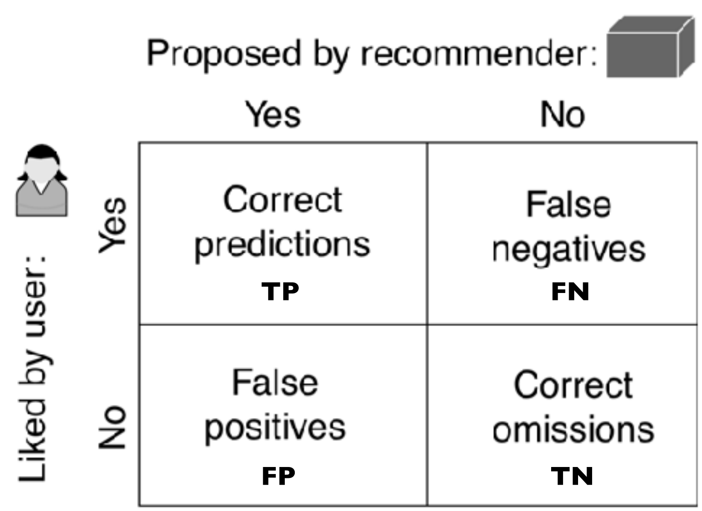
$$\textbf{Precision} = \frac{\#\text{tp}}{\#\text{tp} + \#\text{fp}}$$

$$\textbf{Recall (True Positive Rate)} = \frac{\#\text{tp}}{\#\text{tp} + \#\text{fn}}$$

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

Precision: Of all the items recommended to the user,
how many were actually relevant to the user's interests or preferences?

Recall: Of all the relevant items available,
how many were successfully recommended to the user?



Contents

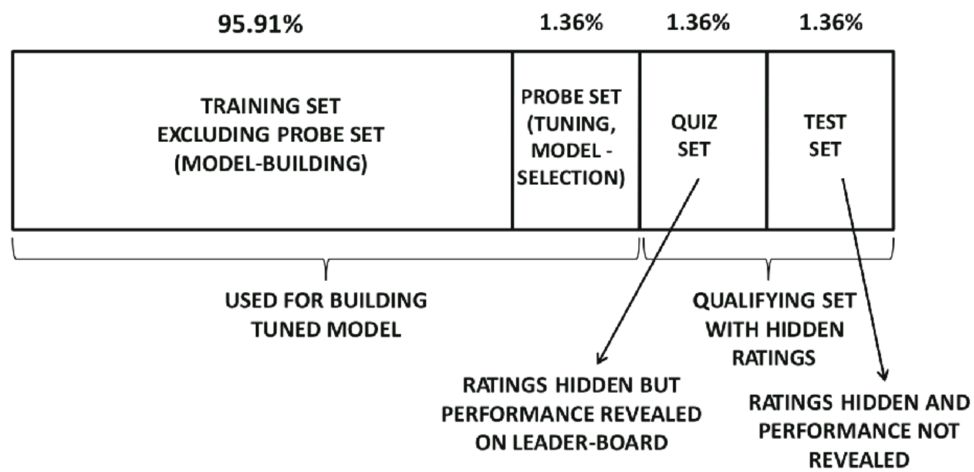
- ➢ Collaborative Filtering Recommendation
  - ✓ Memory-Based:
    - User-Based and Item-Based
  - ✓ Model-Based
    - Matrix Factorization/latent factor models and Association Rules
- ➢ Content-Based Recommendation
- ➢ Knowledge Based Recommendation
- ➢ Demographic Recommendation
- ➢ Hybrid Recommendation Systems
- ➢ RS Evaluation
  - ✓ Online and Offline Evaluations
  - ✓ Design Issues and RS Properties

Design Issues in Offline Recommender Evaluation

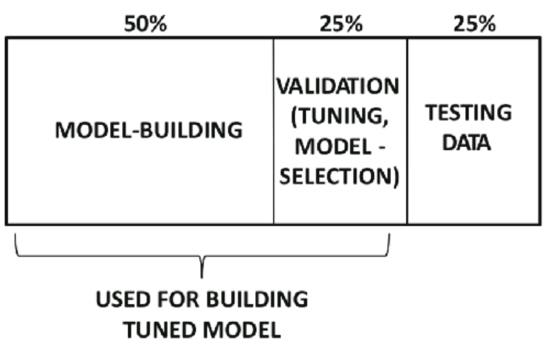
- **Training data**: Used to build the training model.
  For example, in a latent factor model, this part of the data is used to create the latent factors from the ratings matrix. One might even use these data to create multiple models in order to eventually select the model that works best for the data set at hand.

- **Validation data**: Used for model selection and parameter tuning.
  For example, the regularization parameters in a latent factor model may be determined by testing the accuracy over the validation data. In the event that multiple models have been built from the training data, the validation data are used to determine the accuracy of each model and select the best one.

- **Testing data**: Used to test the accuracy of the final (tuned) model.
  It is important that the testing data are not even looked at during the process of parameter tuning and model selection to prevent overfitting. The testing data are used only once at the very end of the process. Furthermore, if the analyst uses the results on the test data to adjust the model in some way, then the results will be contaminated with knowledge from the testing data.



Division in Netflix Prize data set (not drawn to scale)



Proportional division of ratings

Partitioning a ratings matrix for evaluation design.

Recommender System Properties

1. User Preference
2. Prediction Accuracy
3. Coverage
4. Trust
5. Novelty
6. Serendipity
7. Diversity
8. Utility
9. Risk
10. Robustness
11. Privacy
12. Adaptivity
13. Scalability