Recommendation Systems

Introduction

A Model for Recommendation Systems

Collaborative-Filtering System

Content Based System

Ravi Kumar Gupta
https://kravigupta.in

Use of Recommender System

For businesses:

- Increases sales.
- Enables personalized customer service.
- Helps gain customer trust and loyalty.
- Increases knowledge about the customers.
- Provides opportunities to persuade customers.
- Assists in deciding on discount offers.

For customers:

- Helps narrow down choices.
- Aids in finding items of interest.
- Makes navigation through lists easier.
- Allows for discovery of new things.

Where do we see recommendations?

E-commerce systems.

LinkedIn.

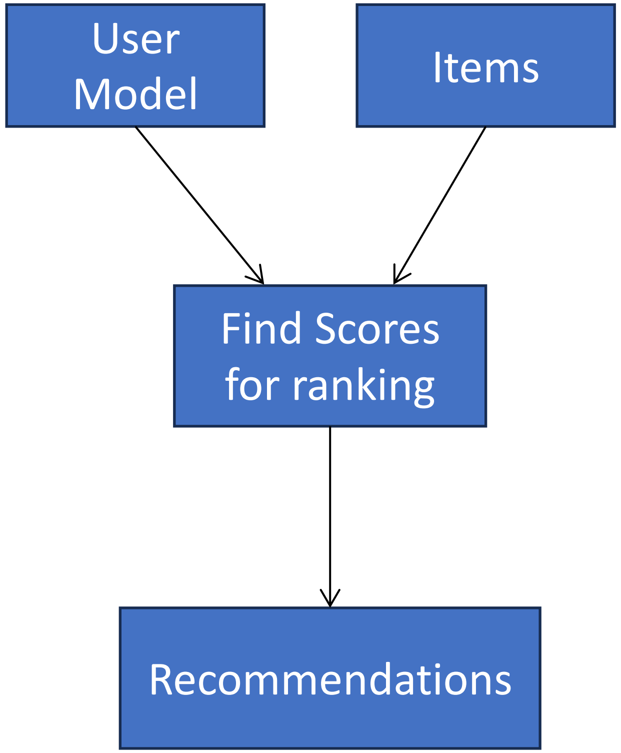
Friend recommendation on Facebook.

Song recommendation on FM.

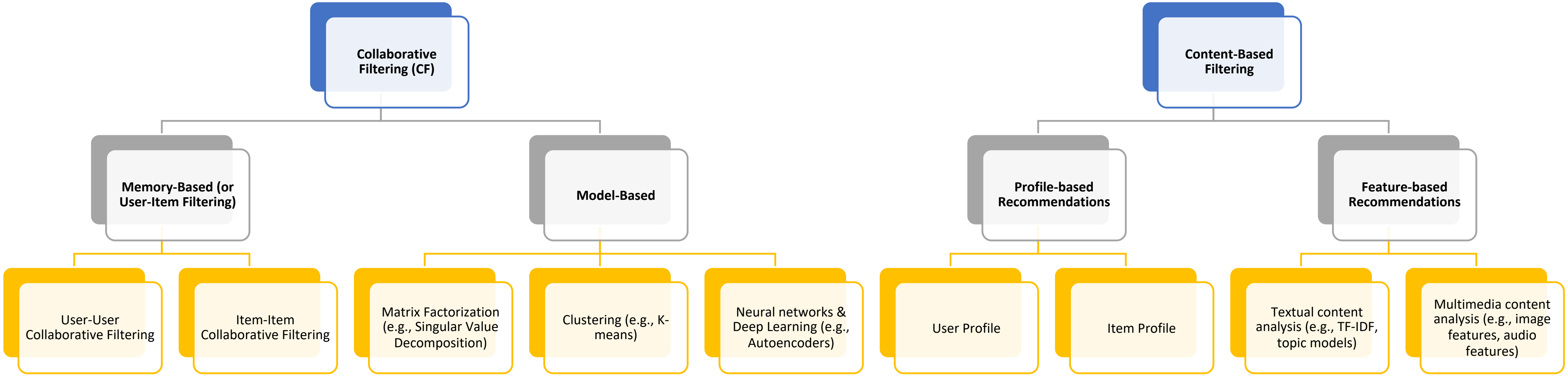
News recommendations at Forbes.com.

What does Recommender system do?

- Takes in user model, which includes:
  - Ratings.
  - Preferences.
  - Demographics, etc.
- Takes in items with their descriptions.
- Finds relevant scores for ranking.
- Recommends items relevant to the user.
- Aims to reduce information overload by estimating relevance.
- Relevance can be context-dependent.

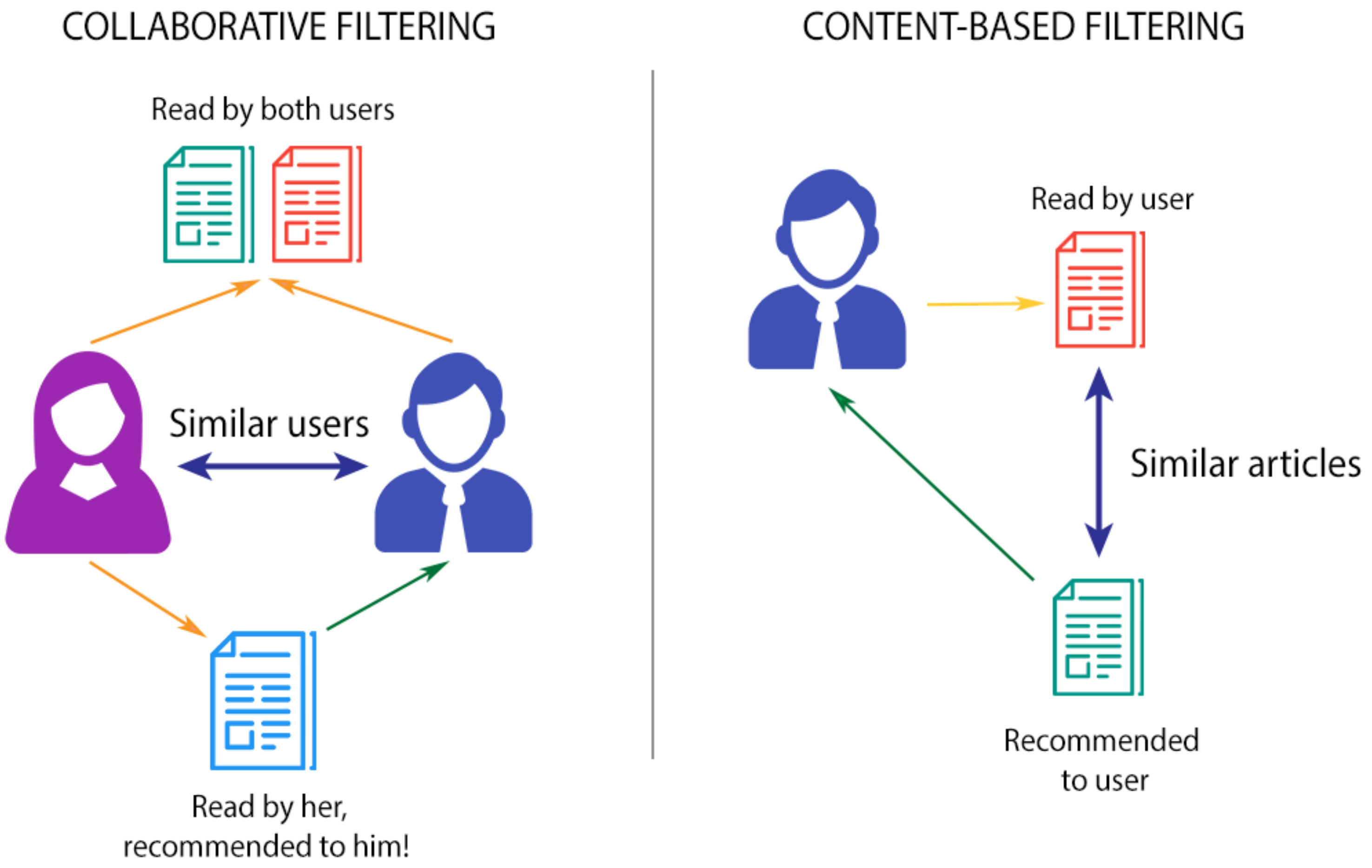


Recommendation Systems


Collaborative Filtering (CF)
Memory-Based (or User-Item Filtering)
User-User Collaborative Filtering
Item-Item Collaborative Filtering
Model-Based
Matrix Factorization (e.g., Singular Value Decomposition)
Clustering (e.g., K-means)
Neural networks & Deep Learning (e.g., Autoencoders)
Content-Based Filtering
Profile-based Recommendations
User Profile
Item Profile
Feature-based Recommendations
Textual content analysis (e.g., TF-IDF, topic models)
Multimedia content analysis (e.g., image features, audio features)

There are other types of recommendation Systems also.
Hybrid, Knowledge Based, Demographic based, Context-aware, and Session Based.

Recommendation systems



Recommendation systems

Collaborative-Filtering System:

- Utilizes community data from peer groups for recommendations.
- Recommends popular items among peer groups.
- Recommends items based on similarity measures between users and/or items.
- Uses user profile, contextual parameters, and community data for personalization.

Content-Based Systems:

- Examines properties of the items recommended.
- Recommends items based on user preferences and item characteristics.
- For example, if a user has watched many "scientific fiction" movies, it recommends movies classified in the database as having the "scientific fiction" genre.
- Takes input from user profiles, contextual parameters, and product features for recommendations.

Collaborative-Filtering System

Collaborative Filtering Systems

- Focus on the relationship between users and items.
- Determine item similarity based on the similarity of ratings given by users who have rated both items.
- Prominent approach in commercial e-commerce sites.
- Applicable in domains such as books, movies, and video recommendations.
- Group people with similar tastes together.
- Feedback from any or some users from the group can be used to recommend items to the group.
- Basic assumptions for this approach:
  - Users provide ratings to items in the catalog.
  - Customers with similar tastes in the past will have similar tastes in the future.
  - Users who agreed on subjective evaluations in the past will agree in the future as well.
    - (i.e., they gave similar ratings or liked the same items)

User Based CF

Nearest Neighbor Technique

Purpose –

To predict the rating an Active user would give for an unseen item.

Determine if Alice will like or dislike the item5.

Process –

- Selecting Peers
- Calculating Similarities
- Prediction function
- Predicting the rating

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

Nearest Neighbor Technique

Process –

- Selecting Peers
  - Choose users (peers) who've liked the same items as the “Active User” and rated them in the past.
  - For unseen items, use the average rating given by these peers.
- Calculating Similarities
- Prediction function
- Predicting the rating

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

Nearest Neighbor Technique

Process –

- Selecting Peers
- Measuring Similarities
  - Why? To Determine which peers' ratings should be considered.
  - Use **Pearson Correlation Coefficient** to measure correlation:

$$sim(a,b)=\frac{\sum_{p\in P}(r_{a,p}-\bar{r}_a)(r_{b,p}-\bar{r}_b)}{\sqrt{\sum_{p\in P}(r_{a,p}-\bar{r}_a)^2}\sqrt{\sum_{p\in P}(r_{b,p}-\bar{r}_b)^2}}$$

- “a” and “b” are users.
- $r_{a,p}$ is the rating of user “a” for item “p”.
- “P” is the set of items rated by both “a” and “b”.

  - The similarity measure lies between -1 and 1.
- Prediction function
- Predicting the rating

Nearest Neighbor Technique

$$sim(a,b)=\frac{\sum_{p\in P}(r_{a,p}-\bar{r}_a)(r_{b,p}-\bar{r}_b)}{\sqrt{\sum_{p\in P}(r_{a,p}-\bar{r}_a)^2}\sqrt{\sum_{p\in P}(r_{b,p}-\bar{r}_b)^2}}$$

- “a” and “b” are users.
- $r_{a,p}$ is the rating of user “a” for item “p”.
- “P” is the set of items rated by both “a” and “b”.

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

sim = 0.85

sim = 0.00

sim = 0.70

sim = -0.79

Nearest Neighbor Technique

Process –

- Selecting Peers
- Measuring Similarities
- Prediction function

$$pred(a,p) = \overline{r_a} + \frac{\sum_{b \in N} sim(a,b) \times (r_{b,p} - \overline{r_b})}{\sum_{b \in N} sim(a,b)}$$

  - *pred(a,p)*: This function predicts the rating that user "a" would give to item "p".
  - r_a : Average rating by user across all of the items they have rated
  - *sim(a,b)* : similarity between user a and b.

- Predicting the rating

Nearest Neighbor Technique

Process –

- Selecting Peers
- Measuring Similarities
- Prediction function
- Predicting the rating
  - Assess if the neighbors’ ratings for an unseen item are higher or lower than their average.
  - Combine the rating differences using similarity as weight.
  - Adjust the Active User’s average based on the neighbor’s bias for prediction.

"Let's guess how much our Active User might like this item by looking at how much similar users liked it. But, when doing so, let's give more importance to the opinions of users who are very similar to our Active User."

Improving the prediction

- Not all neighbor ratings might be equally "valuable"
  - Agreement on commonly liked items is not so informative as agreement on controversial items

Possible solution:

- Give more weight to items that have a higher variance
  - If two users both love a controversial movie, that's a stronger indication they share unique tastes.
- Give more weight to "very similar" neighbors, i.e., where the similarity value is close to 1.
- Use similarity threshold or fixed number of neighbors

Memory-based and model-based approaches

User-based CF is said to be "memory-based"

- The rating matrix is directly used to find neighbors / make predictions
- Does not scale for most real-world scenarios
- Large e-commerce sites have tens of millions of customers and millions of items

Model-based approaches

- Based on an offline pre-processing or "model-learning" phase
- At run-time, only the learned model is used to make predictions
- Models are updated / re-trained periodically
- Large variety of techniques used
- Model-building and updating can be computationally expensive

Item Based CF

Item Based CF

- Basic idea:
  - Use the similarity between items (and not users) to make predictions
- Example:
  - Look for items that are similar to Item5
  - Take Alice's ratings for these items to predict the rating for Item5

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

Item Based CF

- Produces better results in item-to-item filtering
- Ratings are seen as vector in *n*-dimensional space
- Similarity is calculated on the basis of angle between the vectors:

$$sim(\vec{a},\vec{b}) = \frac{\vec{a}\cdot\vec{b}}{|\vec{a}|\times|\vec{b}|}$$

Adjusted cosine similarity:

- Take average user ratings into account and transform the original ratings
- U is the set of users who have rated both a and b.

$$sim(a,b) = \frac{\sum_{u\in U}(r_{u,a}-\overline{r}_u)(r_{u,b}-\overline{r}_u)}{\sqrt{\sum_{u\in U}(r_{u,a}-\overline{r}_u)^2}\sqrt{\sum_{u\in U}(r_{u,b}-\overline{r}_u)^2}}$$

Problems

- Cold Start - arises when the system doesn't have sufficient information about new users or new items
  - How to recommend new items? What to recommend to new users?
- User Cold Start
  - Prompt new users to rate a set of items.
  - Use demographic data - Use user-provided information (e.g., age, gender, location). Recommend based on similar demographic groups.
  - Recommend trending items or top-rated items.
  - Suggest New releases
- Item Cold start
  - Use item attributes (e.g., genre, director) for recommendations.
  - Feature in "New Arrivals" or "Just Added" sections.
- Transitivity
  - Use indirect similarities (e.g., if A is similar to B, and B to C, infer similarity between A and C).

Principles

User Based CF

- If two users agreed in the past, they will agree in the future about the preference for certain items.

Item Based CF

- Users who liked this item also liked these other items.

Content based Filtering

- Users who liked this item's features will also like other items with similar features.

Content Based Filtering

Feature Extraction

- For items like images, documents, features extraction is not straight forward
- Recommendation systems can help to filter vast amounts of daily content e.g. relevant topics, news etc.
- A common method to extract features from documents is –
  - Removing stop words and common terms
  - Identify significant words
  - The significance of words is recognized using TF-IDF
- TF-IDF = Term Frequency – Inverse Document Frequency

Document Similarity

- Two types of document similarity
- Lexical Similarity
  - Documents have similar word sequences.
- Semantic Similarity
  - Documents have similar meanings or content.

Recommendation Systems & Similarity:

- Distinct notion of similarity.
- Focus on occurrence of key words in documents.
- Less emphasis on lexical similarity.
- Similar methodology for finding related documents. Similar methods used for Semantic Similarity.

Content Based Recommendation System

- Incorporates user profiles and contextual parameters.
- Uses product features to match user's past interests.
- Direct item comparisons.
- Doesn't rely on community reviews.
- Focuses on item properties. Some info about the product e.g. genre of the movie
  - Some, But not all of the properties
- Similarity is measured using these properties
- Requires the user profile – what user likes, dislikes. This is essential.
- After learning the user preferences, recommendation for matching items is made.

Item profiles

- Each item has an associated profile.
- This profile is a collection of records highlighting the item's significant characteristics.
- The content or features within the profile help the system make informed recommendations.

Example: In a movie recommendation system, the item profile could include:

1. Cast of the movie.
2. Director of the movie.
3. Release year.
4. Genre or type of the movie.
5. (Potentially) Other relevant features like plot, duration, awards, etc.

For Books it can be – author, year of publication, genre.

For CDs/MP3 songs it can be – artist, composer, genre

Dice Coefficient

- Dice co-efficient is a measure used to compute similarity.
- It is defined for two items, b1 and b2.
- The formula for similarity is:

$$\mathrm{sim}(b_1, b_2) = (2^*((\mathrm{keywords}(b_1) \cap (\mathrm{keywords}(b_2))) \,/\, ((\mathrm{keywords}(b_1) + (\mathrm{keywords}(b_2))$$

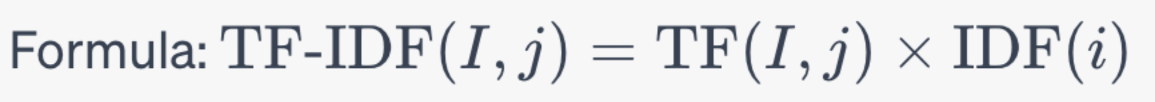
Assumptions and Concerns:

1. All keywords are treated as having the same importance.
2. Longer documents might show a higher similarity because they contain more keywords.

This can be improved using TF-IDF

Term Frequency – Inverse Document Frequency (TF-IDF)

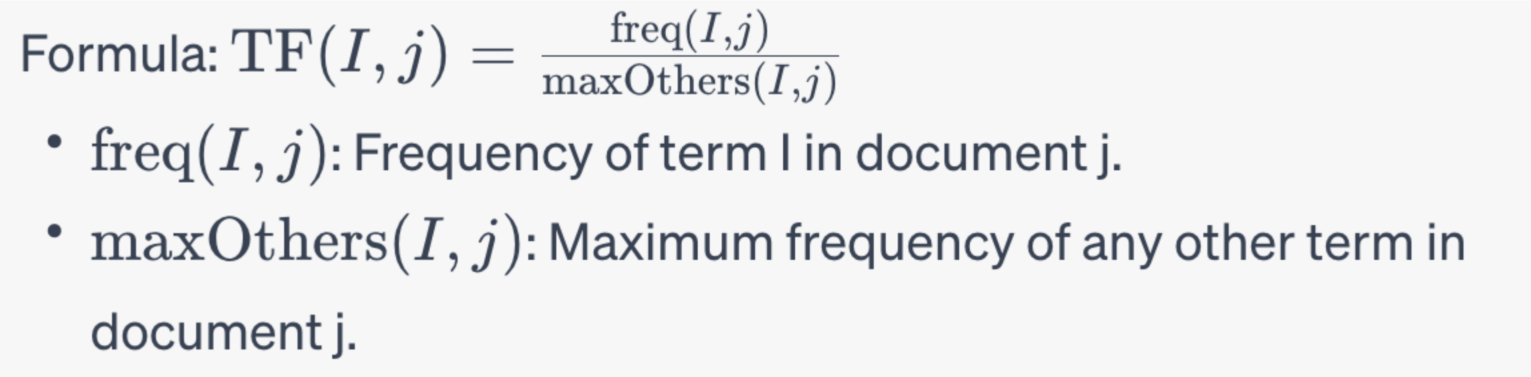
- Weighs each term's importance in a document by considering its local and global frequency.



Formula: $\text{TF-IDF}(I, j) = \text{TF}(I, j) \times \text{IDF}(i)$

Term Frequency (TF)

- Represents the relative frequency of a term in a document.
- Assumption: Important terms appear more frequently in a document.
- TF is normalized to account the document length

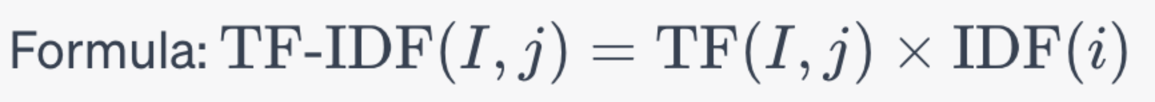


Formula: $\text{TF}(I, j) = \frac{\text{freq}(I,j)}{\text{maxOthers}(I,j)}$

- $\text{freq}(I, j)$: Frequency of term I in document j.
- $\text{maxOthers}(I, j)$: Maximum frequency of any other term in document j.

Term Frequency – Inverse Document Frequency (TF-IDF)

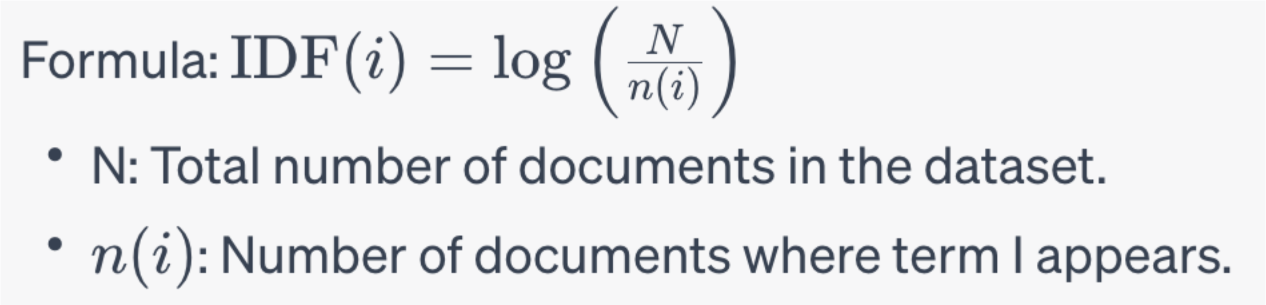
- Weighs each term's importance in a document by considering its local and global frequency.



Formula: $\mathrm{TF\text{-}IDF}(I, j) = \mathrm{TF}(I, j) \times \mathrm{IDF}(i)$

Inverse Document Frequency (IDF)

- Downweighs terms that appear in many documents because they are less informative.
- Assumption: If a term appears in many documents, it's less distinguishing.



Formula: $\mathrm{IDF}(i) = \log\left(\frac{N}{n(i)}\right)$

- N: Total number of documents in the dataset.
- $n(i)$: Number of documents where term I appears.

TF-IDF

- TF-IDF gives weight to every term in a document
  - The weight increases with the term’s frequency
  - The weight decreases if the term is common across many documents
- Ensures that terms which are unique and important to a specific document get higher weights

This approach can be improved by

1. Removing stop words/articles. - removing is/am/are/the etc
2. Using stemming. - replacing runner, running, ran to run
3. Limiting to the top *n* keywords.
4. Usage of word in negative context. - good and not good have opposite meaning
5. Using better methods for keyword/feature selection.
   1. Using TF-IDF or other technique should be used instead of using just frequency
6. Using phrases as terms. - instead of using each word separately use multi-word phrases.
   1. “Machine Learning” – machine and learning have different meaning when used separately