**Recommendation Systems**

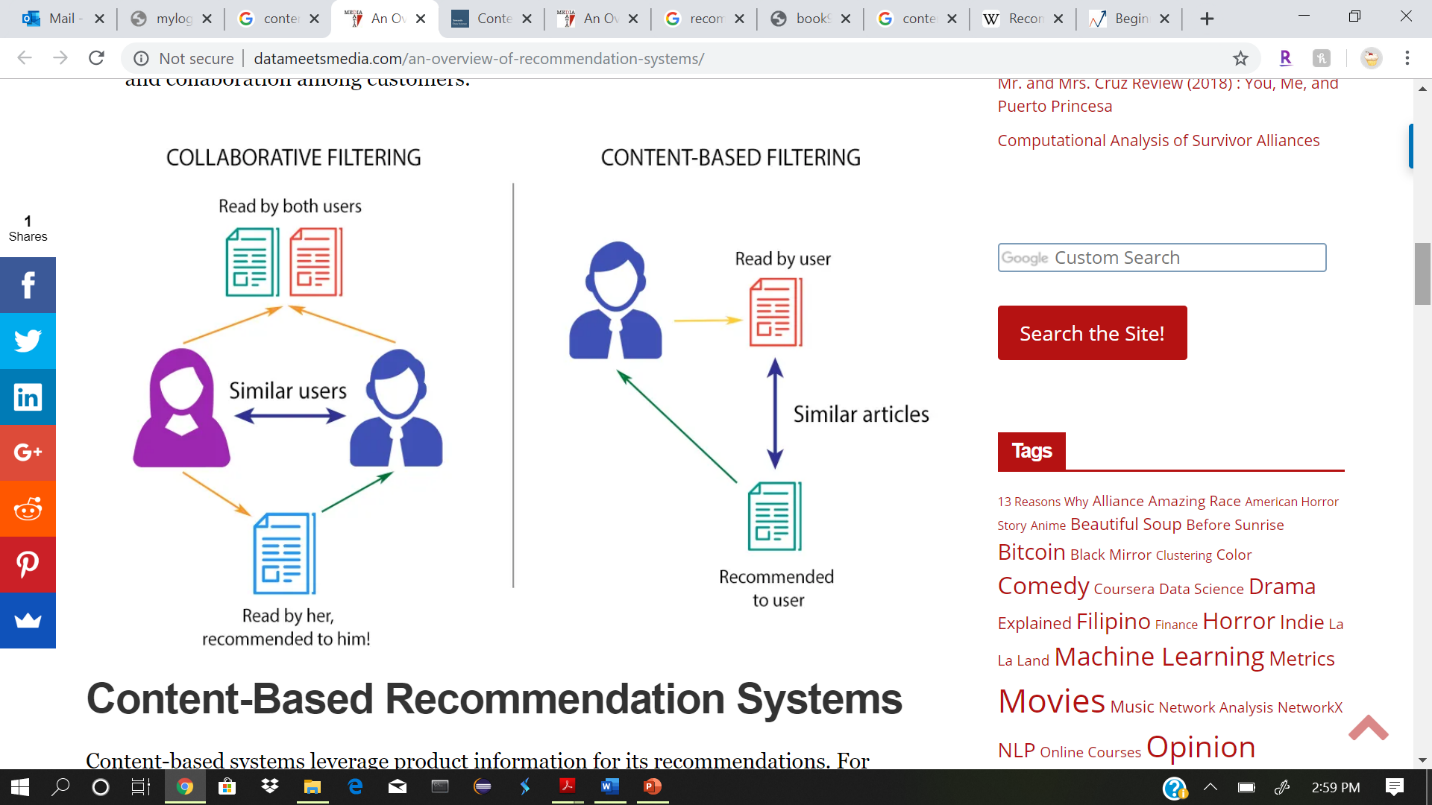
A recommender system is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. They are primarily used in commercial applications.

Recommender systems are utilized in a variety of areas and are most commonly recognized as playlist generators for video and music services like Netflix, YouTube and Spotify, product recommenders for services such as Amazon, or content recommenders for social media platforms such as Facebook and Twitter. These systems can operate using a single input, like music, or multiple inputs within and across platforms like news, books, and search queries. There are also popular recommender systems for specific topics like restaurants and online dating. Recommender systems have been developed to explore research articles and experts, collaborators, financial services, and life insurance.

Recommendation systems use several different technologies. We can classify these systems into two types.

• **Content-based systems:** examine properties of the items recommended. For instance, if a Netflix user has watched many cowboy movies, then recommend a movie classified in the database as having the “cowboy” genre.

• **Collaborative filtering systems:** recommend items based on similarity measures between users and/or items. The items recommended to a user are those preferred by similar users.



**Applications of Recommendation Systems:**

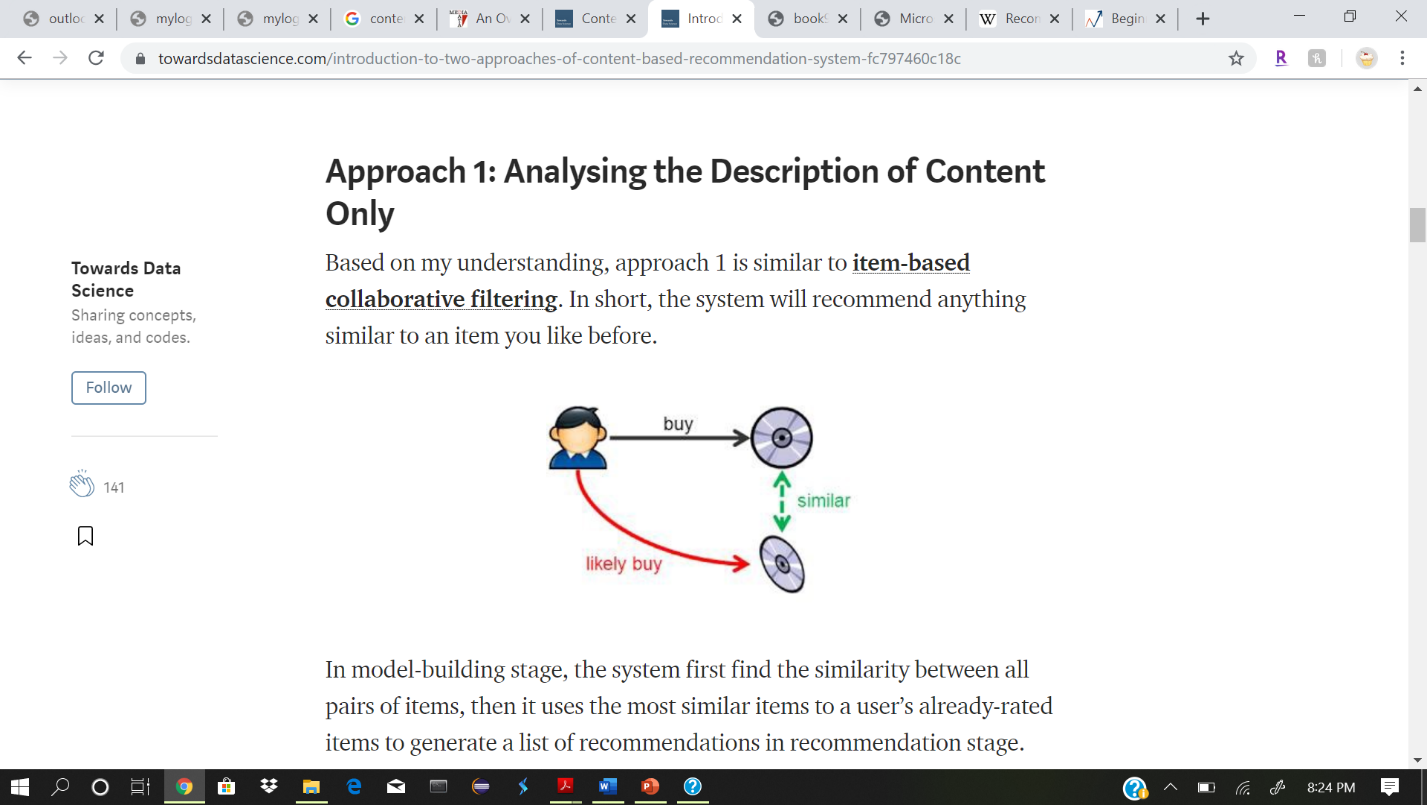
1. Product Recommendations: Perhaps the most important use of recommendation systems is at on-line retailers. We have noted how Amazon or similar on-line vendors strive to present each returning user with some. suggestions of products that they might like to buy. These suggestions are not random but are based on the purchasing decisions made by similar customers.
2. Movie Recommendations: Netflix offers its customers recommendations of movies they might like. These recommendations are based on ratings provided by users, much like the ratings.
3. News Articles: News services have attempted to identify articles of interest to readers, based on the articles that they have read in the past. The similarity might be based on the similarity of important words in the documents, or on the articles that are read by people with similar reading tastes.

**Content-Based Recommender Systems:**

A content-based recommender works with data that the user provides, either explicitly (rating) or implicitly. Based on that data, a user profile is generated, which is then used to make suggestions to the user. As the user provides more inputs or takes actions on the recommendations, the engine becomes more and more accurate.

Approach 1 — Analyzing Description of the Content Only

Approach 2 — Building User Profile and Item Profile from User Rated Content

Based on my understanding, approach 1 is like item-based collaborative filtering. In short, the system will recommend anything similar to an item you like before.

In model-building stage, the system first finds the similarity between all pairs of items, then it uses the most similar items to a user’s already-rated items to generate a list of recommendations in recommendation stage.

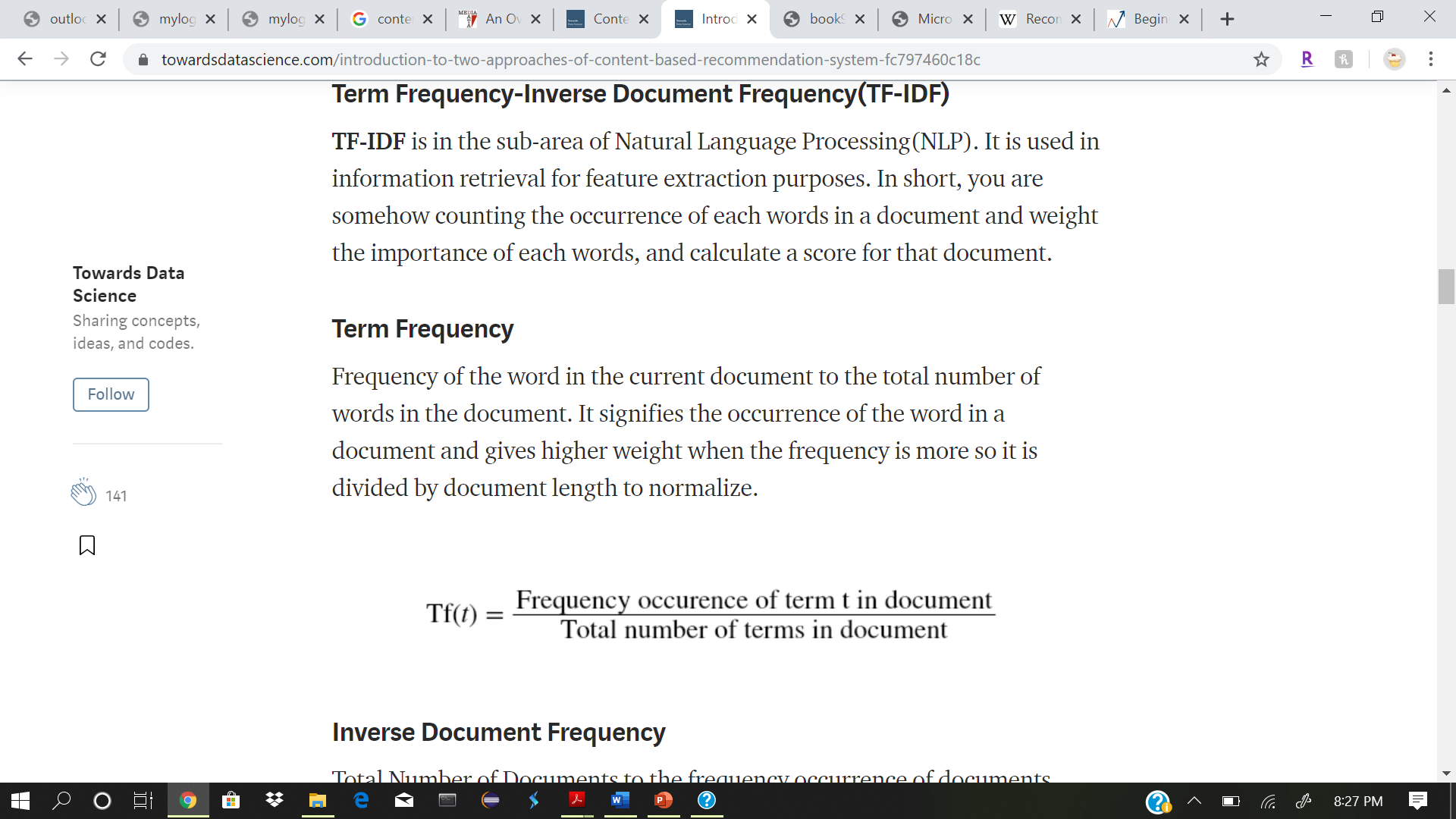
How can you find the similarity between items?

Usually the similarity will be derived from the description of the item and the concept of TF-IDF will be introduced. Then each item will be represented by a TF-IDF vector.

Term Frequency-Inverse Document Frequency (TF-IDF)

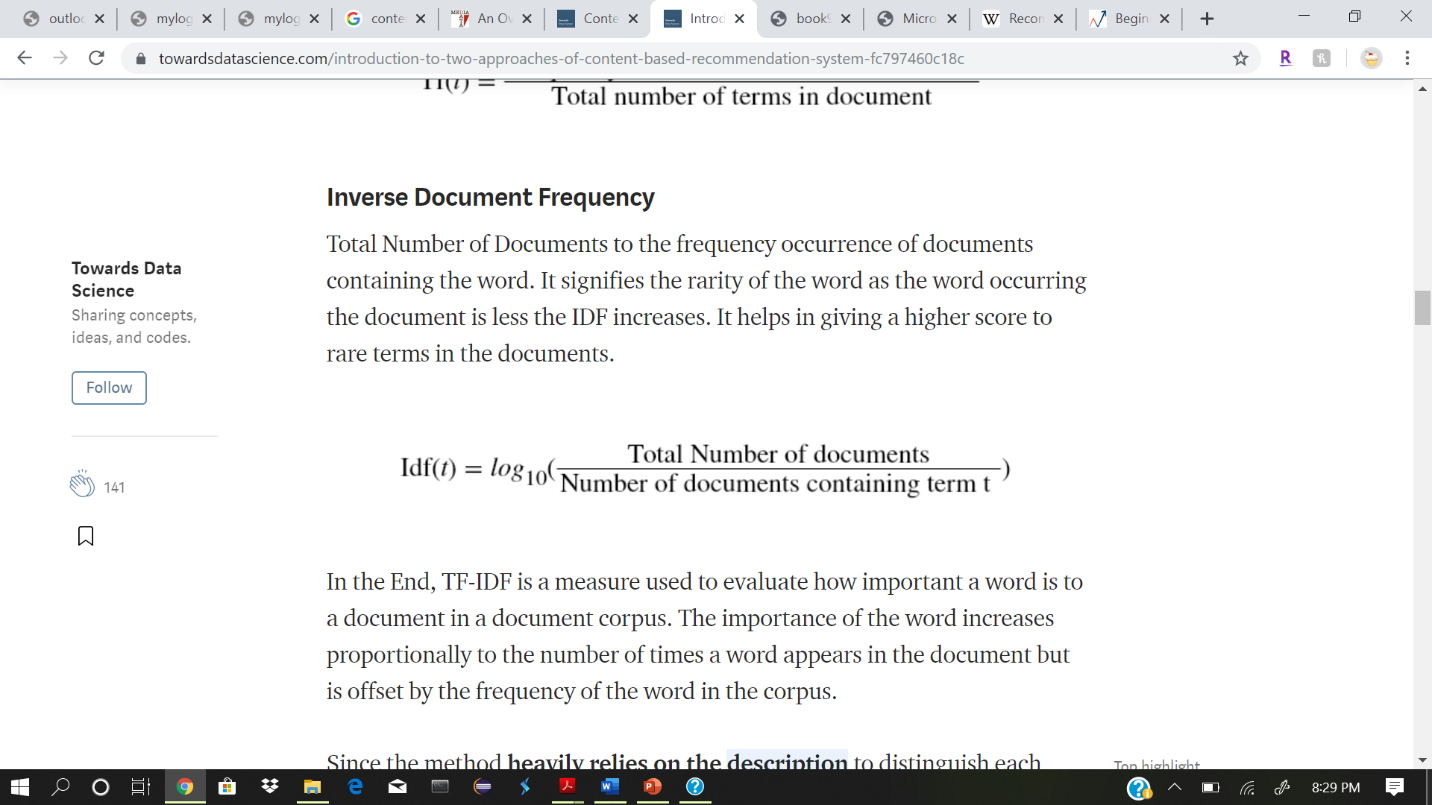
TF-IDF is in the sub-area of Natural Language Processing (NLP). It is used in information retrieval for feature extraction purposes. In short, you are somehow counting the occurrence of each words in a document and weight the importance of each words and calculate a score for that document.

Term Frequency

Frequency of the word in the current document to the total number of words in the document. It signifies the occurrence of the word in a document and gives higher weight when the frequency is more, so it is divided by document length to normalize.

Inverse Document Frequency

Total Number of Documents to the frequency occurrence of documents containing the word. It signifies the rarity of the word as the word occurring the document is less the IDF increases. It helps in giving a higher score to rare terms in the documents.



In the End, TF-IDF is a measure used to evaluate how important a word is to a document in a document corpus. The importance of the word increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

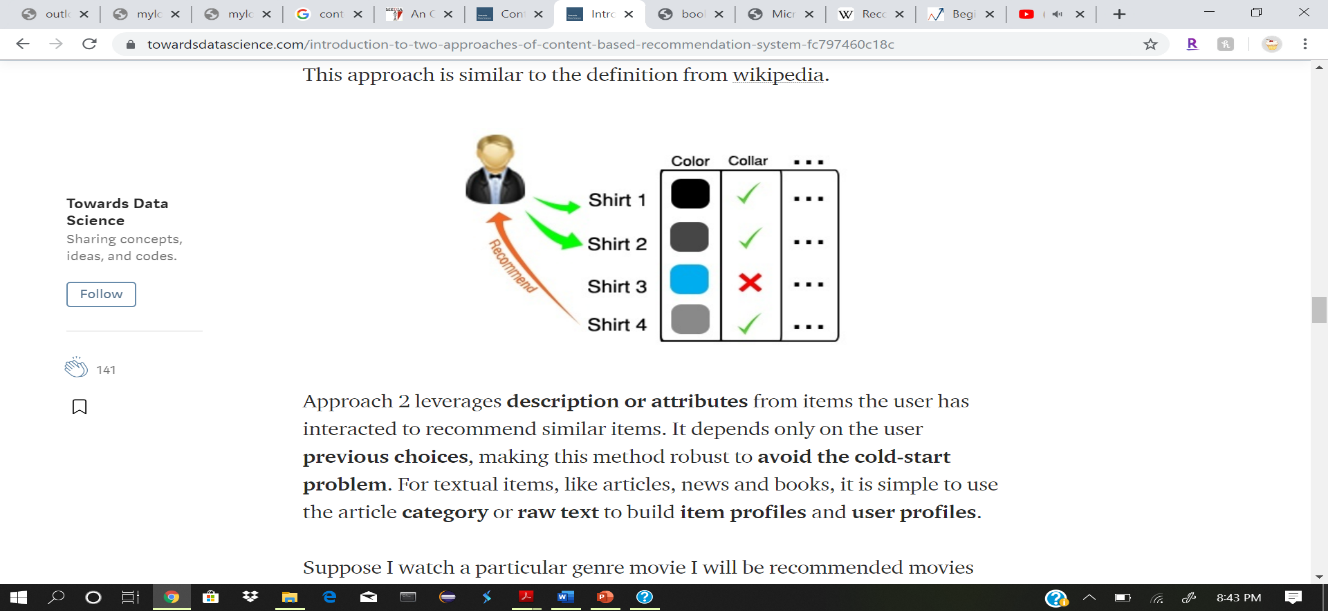
**Compare the Similarity of the item TF-IDF vector:**

To compute how similar the item vectors are, we will can use various methods such as:

1. Cosine Similarity
2. Euclidean Distance
3. Pearson’s Correlation

Then the recommender will give recommendation based on the most similar items.

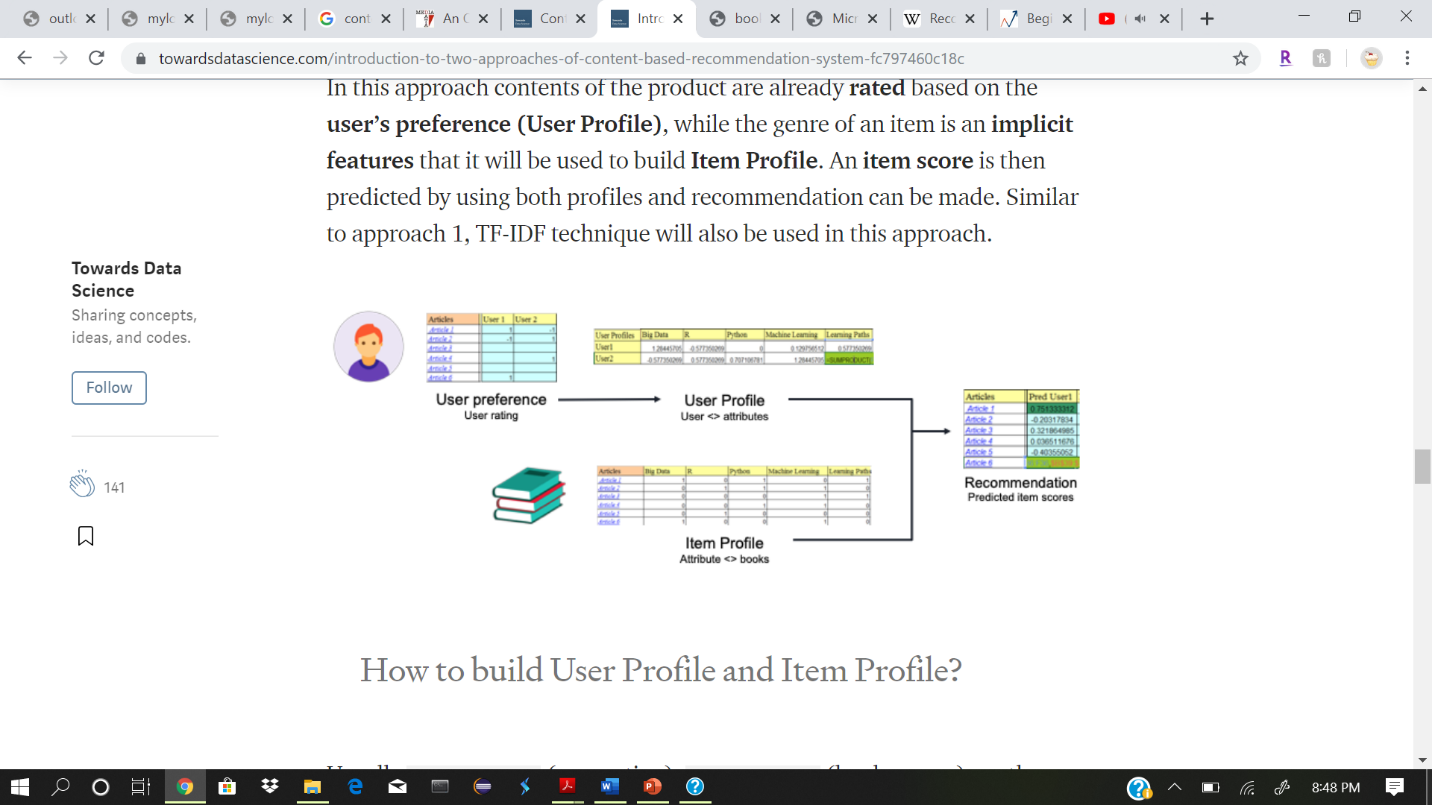
**Approach 2: Building User Profile and Item Profile from User Rated Content**



Approach 2 leverages description or attributes from items the user has interacted to recommend similar items. It depends only on the user previous choices, making this method robust to avoid the cold-start problem. For textual items, like articles, news and books, it is simple to use the article category or raw text to build item profiles and user profiles.

Suppose I watch a genre movie I will be recommended movies with respective to that specific genre. The Title, Year of Release, Director, Cast are also helpful in identifying similar movie content.

n this approach contents of the product are already rated based on the user’s preference (User Profile), while the genre of an item is an implicit feature that it will be used to build Item Profile. An item score is then predicted by using both profiles and recommendation can be made. Similar to approach 1, TF-IDF technique will also be used in this approach.



**Pros of content-based filtering:**

1. No need for data on other users
2. Able to recommend to users with unique tastes
3. Explanations for recommended items
4. content features that caused an item to be recommended
5. Able to recommend new & unpopular items: no first rater problem

**Cons of content-based approach:**

1. Finding the appropriate features is hard (ex: images, movies)
2. Over specialization

a) Never recommends items outside user's content profile

b) people might have multiple interests

c) Unable to exploit quality judgements

3) Cold start problem for new users a) how to build a user profile

**Collaborative filtering Algorithm:**

The motivation for collaborative filtering comes from the idea that people often get the best recommendations from someone with tastes similar to themselves. Collaborative filtering encompasses techniques for matching people with similar interests and making recommendations on this basis.

Collaborative filtering algorithms often require users' active participation, an easy way to represent users' interests, and algorithms that are able to match people with similar interests.

Typically, the workflow of a collaborative filtering system is:

1. A user expresses his or her preferences by rating items (e.g. books, movies or CDs) of the system. These ratings can be viewed as an approximate representation of the user's interest in the corresponding domain.
2. The system matches this user's ratings against other users' and finds the people with most "similar" tastes.
3. With similar users, the system recommends items that the similar users have rated highly but not yet being rated by this user (presumably the absence of rating is often considered as the unfamiliarity of an item)

A key problem of collaborative filtering is how to combine and weight the preferences of user neighbors. Sometimes, users can immediately rate the recommended items. As a result, the system gains an increasingly accurate representation of user preferences over time.

**Few approaches for User and Item-based collaborative recommendation techniques are as follow:**

1. Neighborhood-based approach

2. Item-based approach

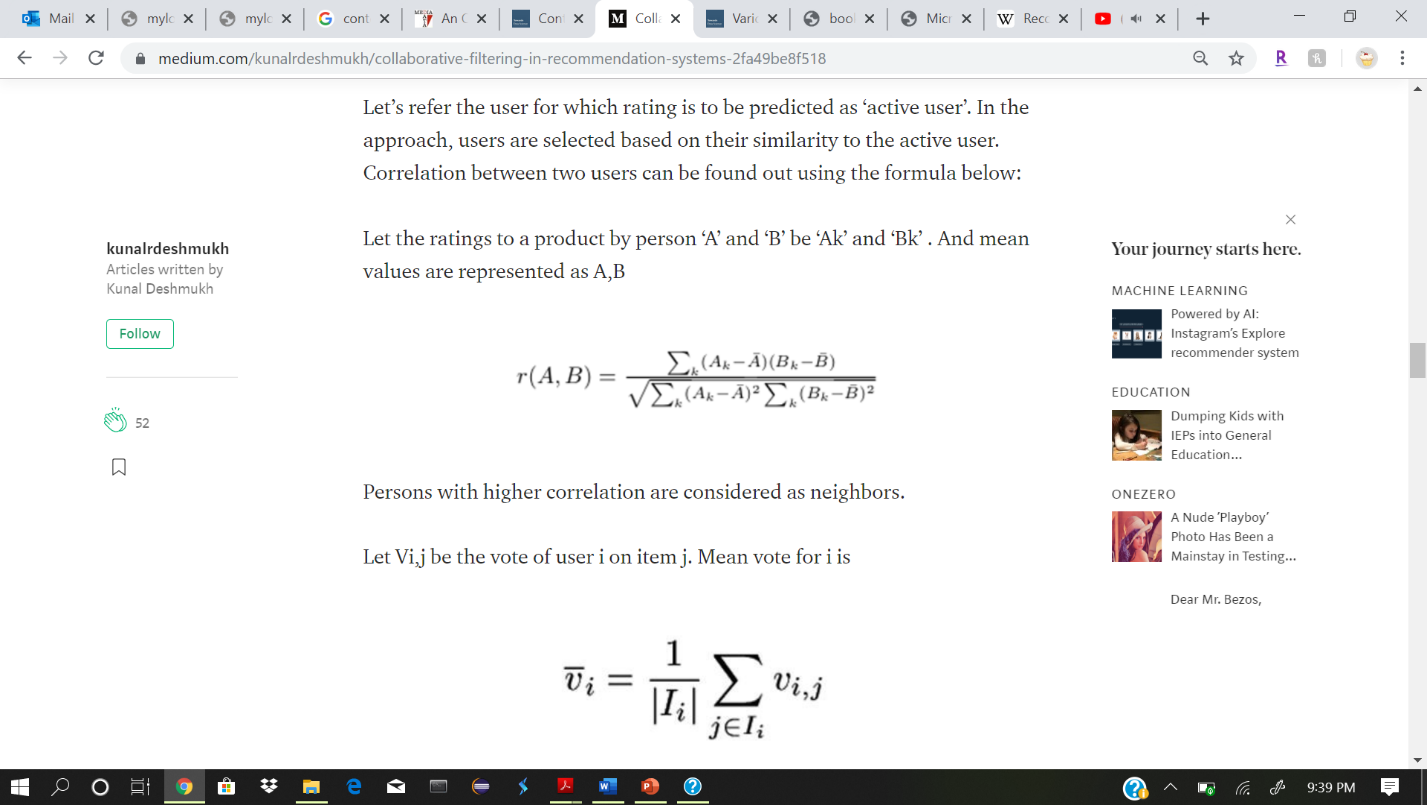
3. Classification approach

4. Neural Collaborative Filtering

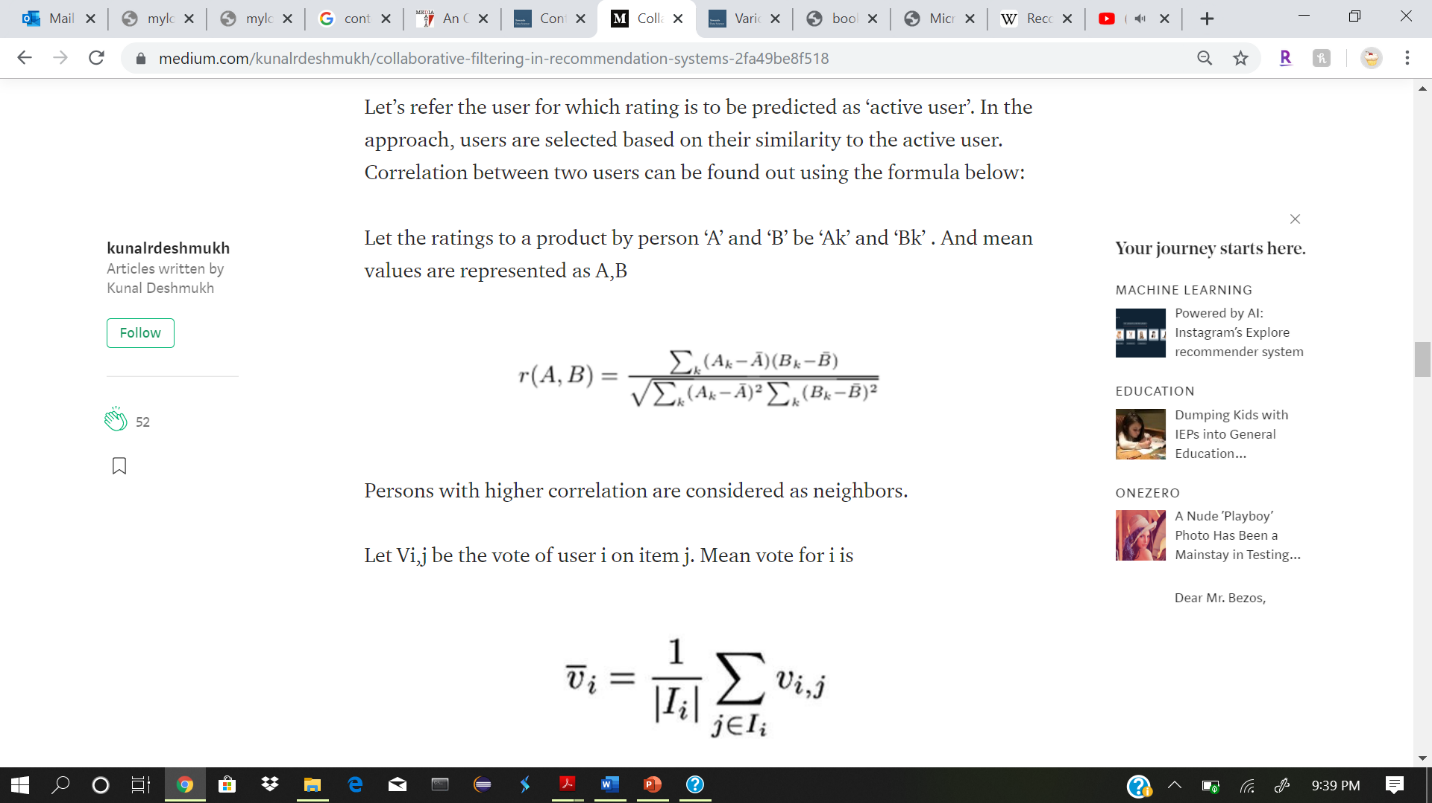
1) Neighborhood-based approach

Let’s refer the user for which rating is to be predicted as ‘active user’. In the approach, users are selected based on their similarity to the active user. Correlation between two users can be found out using the formula below:

Let the ratings to a product by person ‘A’ and ‘B’ be ‘Ak’ and ‘Bk’. And mean values are represented as A, B

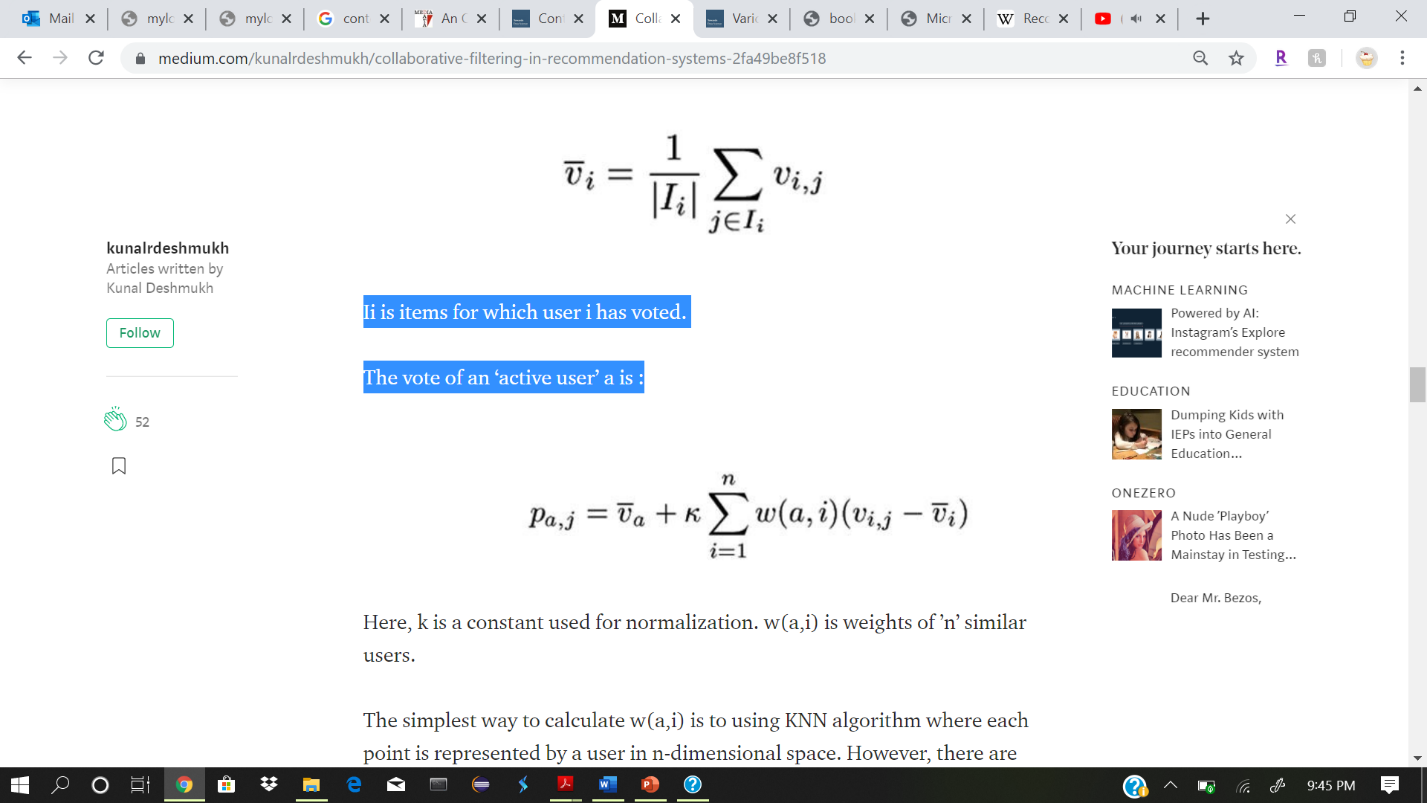


Persons with higher correlation are considered as neighbors.

Let Vi,j be the vote of user i on item j. Mean vote for i is

Ii is items for which user i has voted.

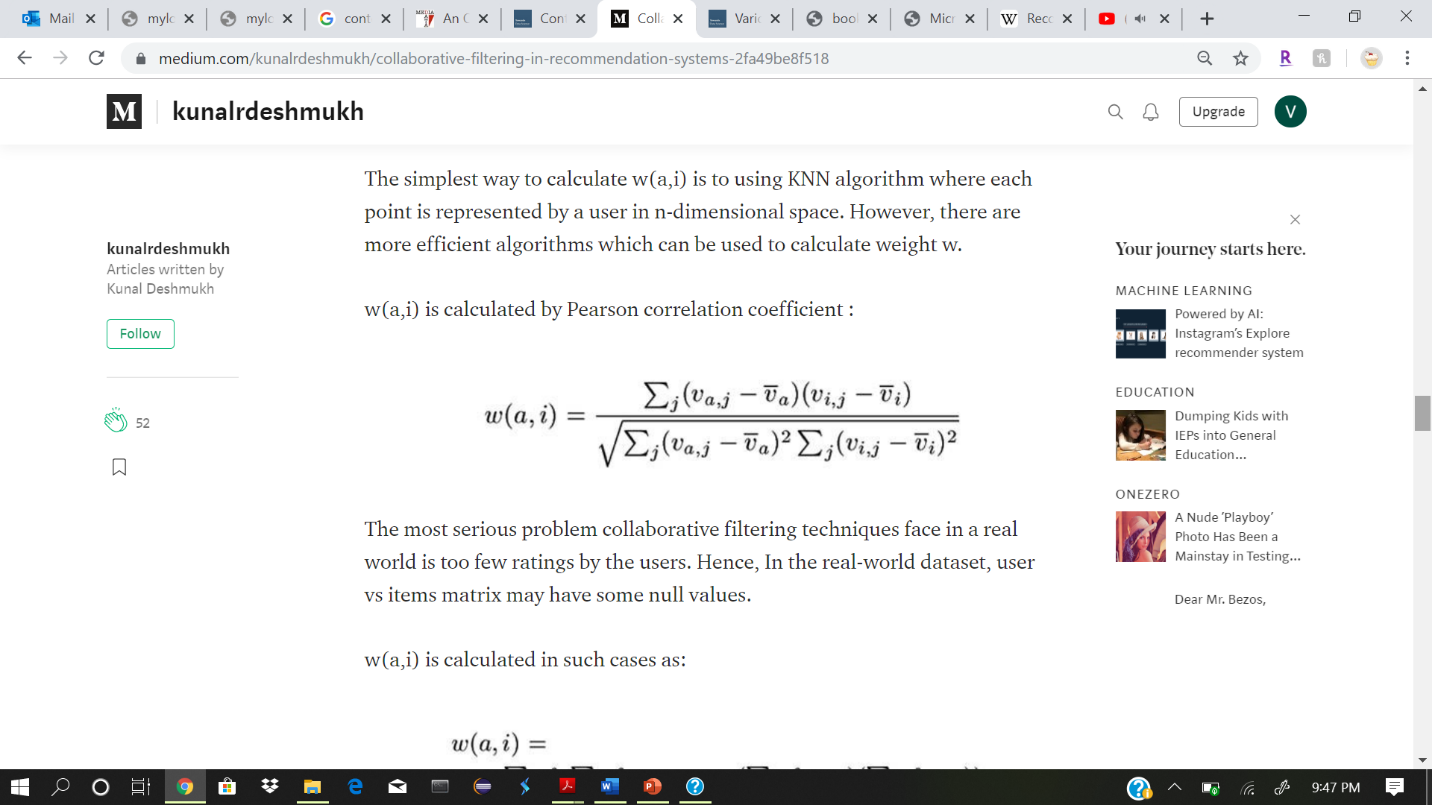
The vote of an ‘active user’ a is:



Here, k is a constant used for normalization. w(a,i) is weights of ’n’ similar users.

The simplest way to calculate w(a,i) is to using KNN algorithm where each point is represented by a user in n-dimensional space. However, there are more efficient algorithms which can be used to calculate weight w.

w(a,i) is calculated by Pearson correlation coefficient :



The most serious problem collaborative filtering techniques face in a real world is too few ratings by the users. Hence, In the real-world dataset, user vs items matrix may have some null values.

2. Item -to -Item approach

Instead of using ratings given by the users to calculate neighborhood, the ratings are used to find similarity between items. The same Pearson coefficient can be used for this approach.

3. Classification approach

In classification approach, items are represented as vectors and they are classified and suggested to the user based on the ratings provided by the active user to each class of items. With this approach, collaborative filtering is visualized as classic classification approach. Classification techniques such as Support Vector Machines or Bayesian classifier can be used. Random Forest proves useful in case of the unbalanced dataset.

4. Neural Collaborative Filtering

Neural networks are being used increasingly for collaborative filtering. In this use, User-item interaction matrix data is treated as an implicit data. While observed entries at least reflect users’ interest on items, the unobserved entries can be just missing data and there is in most cases, natural scarcity of negative feedback. The recommendation problem with implicit feedback is formulated as the problem of estimating the scores of unobserved entries. Matrix factorization is used to estimate predicted output. The missing data is replaced by using this input. Filled input space is then passed to a multi-layer perceptron network to estimate ratings for an active user.

**Hybrid Recommendation System**

The hybrid recommendation system is a combination of collaborative and content-based filtering techniques. In this approach, content is used to infer ratings in case of the sparsity of ratings. This combination is used in most recommendation systems at present. Netflix movie recommendation system is an example of hybrid recommendation system.

Scikit-surprise package is in python is useful to implementation of recommendation system. Since there is no single ‘correct’ way to implement recommendation system, various machine learning algorithms for classification can be explored

**Advantages of Collaborative filtering**:

* No domain knowledge necessary
* The model can help users discover new interests.
* the system needs only the feedback matrix to train a matrix factorization model.

**Disadvantages of Collaborative filtering**:

* Cannot handle fresh items
* Hard to include side features for query/item