**NEWS RECOMMENDER SYSTEM**

****

**USING PYTHON**

**Project Report**

**Submitted by:-**

**Vijay Garg**

**Aakriti Mittal**

**Gurleen Kaur Narang**

**Arshdeep Singh Randhawa**

**Business understanding**

**Agenda - To build a news recommender system for Jhakaas News Vala.**

Problem background :The company is developing an app that promises to deliver a unique news experience to its app users. The company has identified it target market as working professionals in the age group 21-40. Recognising the fact that retention (defined here as a visit after the first visit) is a huge issue for apps, they understand the need to make an impact on the first visit itself. The problem however is that they know nothing about the user interests or demographics at the time to personalise the news feed to them.

The company has acquired a corpus of news stories. The real estate available for providing news stories (the mobile phone’s screen) is limited and so without a scroll, only 10 stories can be displayed. Statistics show that the number of users scrolling beyond the first set of stories drops off very quickly unless a story on the first page catches the users eye (that is, results in a clickthrough). You have been tasked with the job of building two intelligent bots. 1. The article recommender: This bot selects articles to serve a user. Inputs to the bot is the corpus of new articles and a user profile if available. 2. The user profiler: Once the user starts consuming news stories, (s)he leaves behind a clickstream.

**Business Goals:**

The bot must extract user interests from such data that can then be used for further personalisation for (her)his news feed. The ultimate objective is to increase clickthrough and the frequency with which the user opens the app to consume stories. However, the objective in the first visit is to:

∙ Reduce bias in data collection (Example Bias: Stories that get served often and ranked higher, have a higher likelihood of being consumed (obtaining a clickthrough))

∙ Learn as much as possible about the users on their first visit

∙ Maximise coverage of the news corpus.

**DATA UNDERSTANDING**

There are mainly two problems so we have formed two data sets:

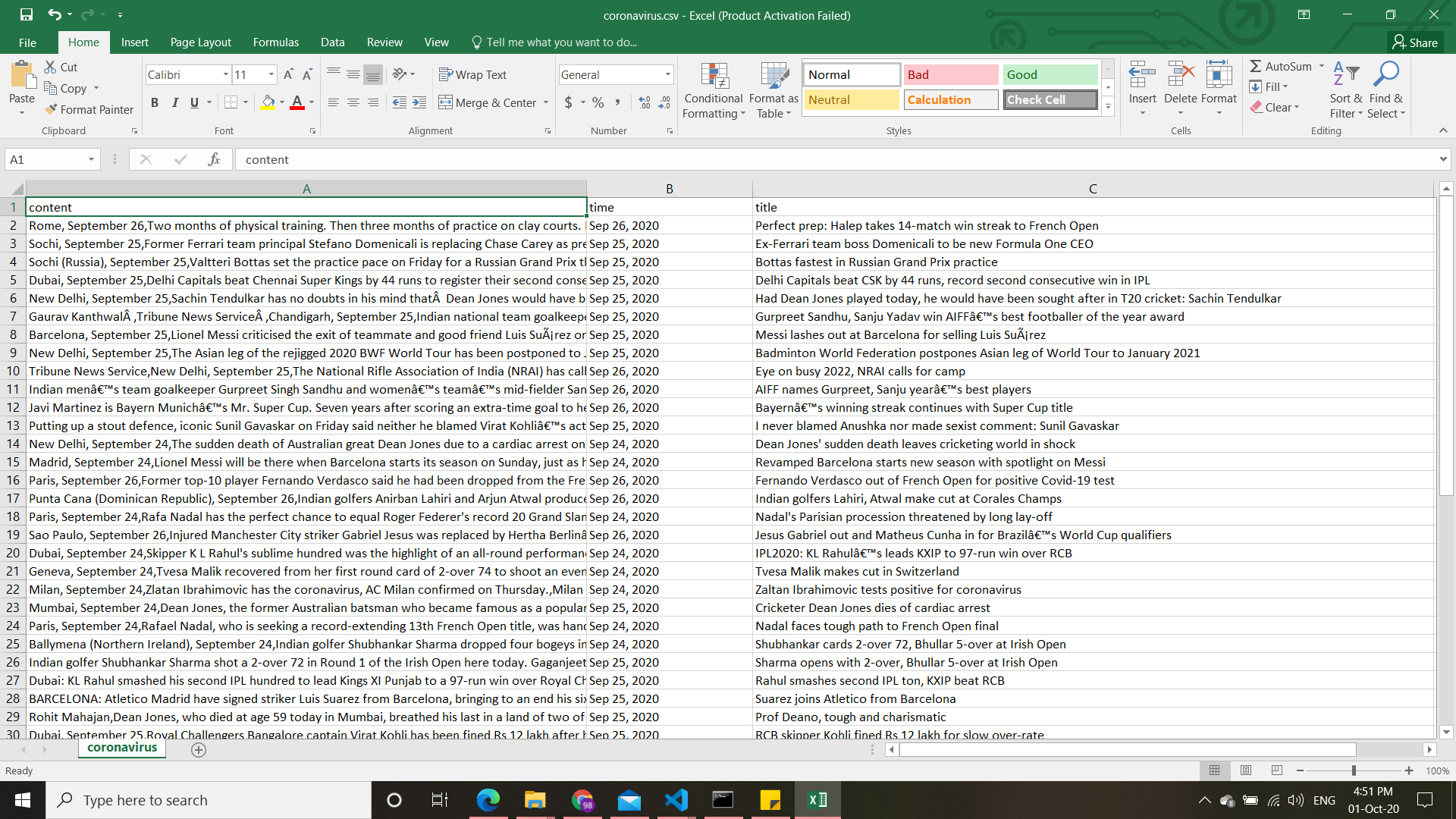
**Dataset for Cold start problem**

Have scraped two web sites for news content material and title material the websites are The Tribune and the Times of India. We have tried to scrap the different subject news.

**Dataset for Personalized recommender system**

There are mainly no data set provided to us as it is totally a new application and is a cold start problem with no previous data present. So we have artificially generated the user personal data for the articles in the 1st dataset. We will discuss in detail about dataset in the next section.

**DATA** **PREPRATION**

Main data set

We have scraped data and saved it in three type of rows as

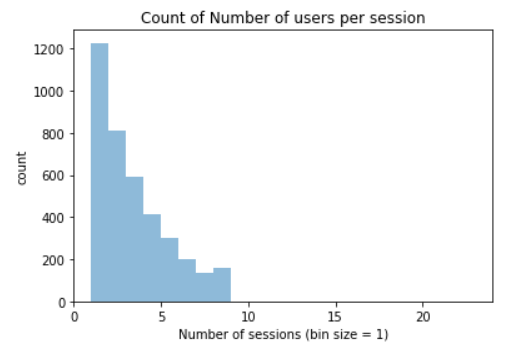
**Date and time** – date time of the article published

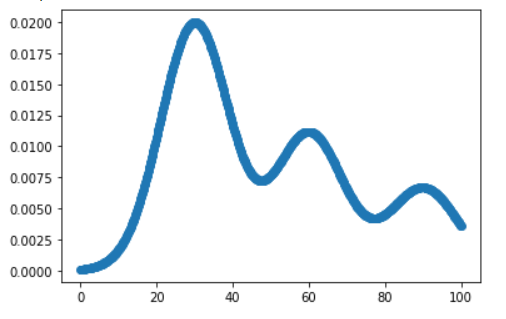
**Content** - content of the article of the particular news

**Title** – news title of the news article

**User Based data set**

* This process was an integration of various Distributions.
* For the process of making UserId, we made the list of 4000 users using simple Integers ranging from 1 to 4000.
* For the purpose of making sessions for each user we used Geometric distribution because Geometric distribution has one analogy that each user must have a minimum 1 session. Because Geometric Distribution says the number of failures before you get a success in a series of Bernoulli trials.
* Minimum session count for each user is 1 and maximum session count is 27.
* For the purpose of generating clicks per session for each we used Poisson distribution because Poisson distribution is a tool that helps to predict the probability of certain events from happening when you know how often the event has occurred. It gives us the probability of a given number of events happening in a fixed interval of time.
* **HOW IT WORKS**: - Assuming Poisson distribution has generated 2 clicks in one session for any user so we have assigned those 2 clicks randomly to 10 news articles assigned to each user in one session. We have randomly assigned these 2 clicks to articles because this data is generated by distribution not by actual clickstream data.
* For the purpose of generating average time spent by the user in reading we used 3 Gaussian Mixture models with different mean and standard deviation to cater the different reading speeds of each and also to take into account the variable of Length of articles which ranges from 250 words to 8100 words.





**Text Representation with Feature Engineering**

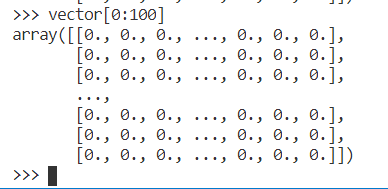
Feature Engineering is often known as the secret sauce to creating superior and better performing machine learning models. The importance of feature engineering is even more important for unstructured, textual data because we need to convert free flowing text into some numeric representations which can then be understood by machine learning algorithms.

Here we will explore the following feature engineering techniques:

* TF-IDF Model
* Word2Vec
* Document Similarity

**TF-IDF Model**

The TF-IDF model tries to combat this issue by using a scaling or normalizing factor in its computation. *TF-IDF* stands for Term Frequency-Inverse Document Frequency, which uses a combination of two metrics in its computation, namely: *term frequency (tf)* and *inverse document frequency (idf)*. This technique was developed for ranking results for queries in search engines and now it is an indispensable model in the world of information retrieval and NLP.



# Document Similarity[¶](https://render.githubusercontent.com/view/ipynb?commit=e4793b51aaba4a74e502d71f63e31699e9a2ba8c&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f646970616e6a616e532f6e6c705f657373656e7469616c732f653437393362353161616261346137346535303264373166363365333136393965396132626138632f6e6f7465626f6f6b732f30325f546578745f526570726573656e746174696f6e5f537461746973746963616c5f4d6f64656c732e6970796e62&nwo=dipanjanS%2Fnlp_essentials&path=notebooks%2F02_Text_Representation_Statistical_Models.ipynb&repository_id=274891753&repository_type=Repository#Document-Similarity)

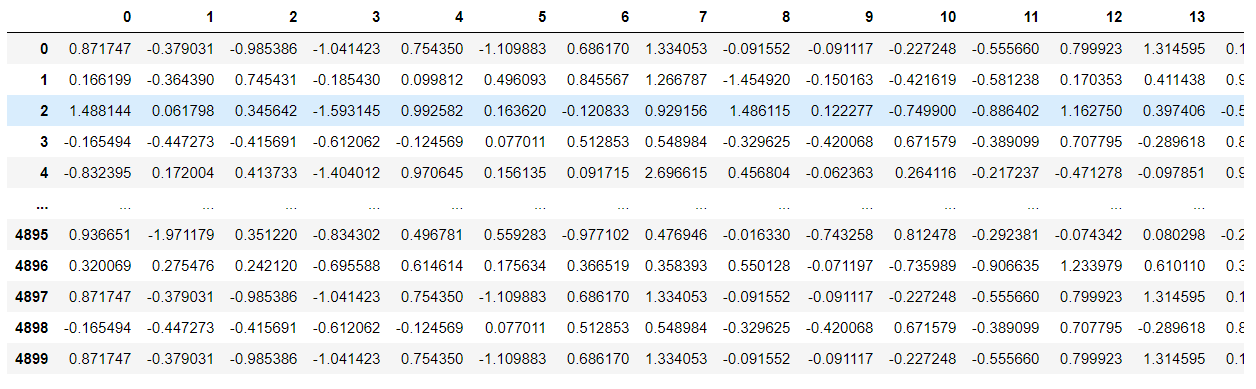
Document similarity is the process of using a distance or similarity based metric that can be used to identify how similar a text document is with any other document(s) based on features extracted from the documents like bag of words or tf-idf.

Thus you can see that we can build on top of the tf-idf based features we engineered in the previous section and use them to generate new features which can be useful in domains like search engines, document clustering and information retrieval by leveraging these similarity based features.

Pairwise document similarity in a corpus involves computing document similarity for each pair of documents in a corpus. Thus if you have C documents in a corpus, you would end up with a C x C matrix such that each row and column represents the similarity score for a pair of documents, which represent the indices at the row and column, respectively. There are several similarity and distance metrics that are used to compute document similarity. These include cosine distance/similarity, euclidean distance, manhattan distance, BM25 similarity, jaccard distance and so on. In our analysis, we will be using perhaps the most popular and widely used similarity metric, cosine similarity and compare pairwise document similarity based on their TF-IDF feature vectors.

**Word2Vec**

While TFIDF is an effective methods for extracting features from text, due to the inherent nature of the model being just a bag of unstructured words, we lose additional information like the semantics, structure, sequence and context around nearby words in each text document. Word2Vec is a predictive deep learning based model to compute and generate high quality, distributed and continuous dense vector representations of words, which capture contextual and semantic similarity. Essentially these are unsupervised models which can take in massive textual corpora, create a vocabulary of possible words and generate dense word embeddings for each word in the vector space representing that vocabulary.



**Clustering:**

Clustering is an unsupervised machine learning task. It involves automatically discovering natural grouping in data. Unlike supervised learning (like predictive modeling), clustering algorithms only interpret the input data and find natural groups or clusters in feature space.

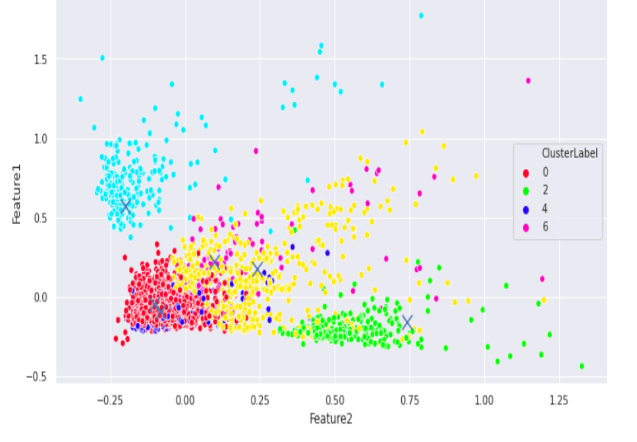
This is the next step to make clusters of data. We have used two approaches to make the clusters.

1. K-means
2. Agglomerative Clustering or hierarchical clustering

We have also calculated the Silhouette score to find the optimal number of clusters.

**K-means**

K-Means Clustering is an [Unsupervised Learning algorithm](https://www.javatpoint.com/unsupervised-machine-learning), which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

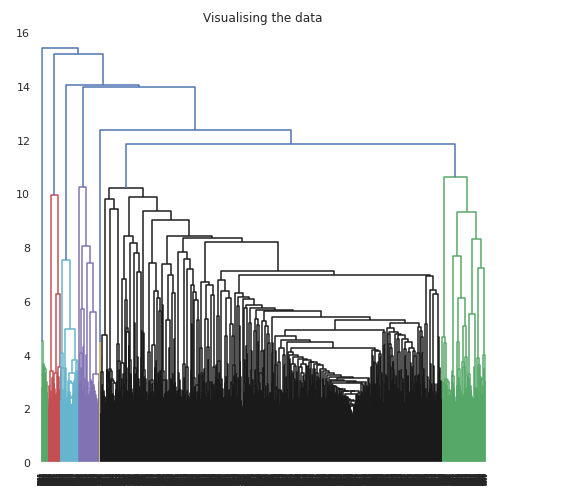


**K-means**

**Agglomerative Clustering or hierarchical clustering**

Hierarchical clustering is another unsupervised machine learning algorithm, which is used to group the unlabeled datasets into a cluster and also known as hierarchical cluster analysis or HCA.

In this algorithm, we develop the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the dendrogram.

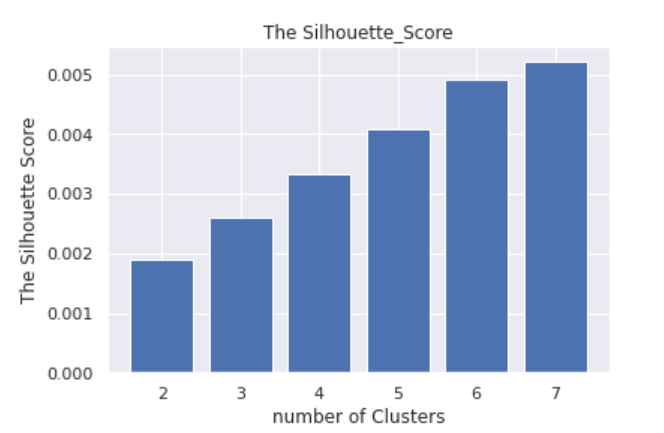


**Dendrogram**

The time complexity of K-means is O(n) whereas the time complexity of Agglomerative Clustering is O(n\*\*2). Our Dataset has 5000 news so we are able to make the clusters with both these algorithms.

**Silhouette score**

Silhouette score is a metric used to calculate the goodness of a clustering technique. Its value ranges from -1 to 1.



1: Means clusters are well apart from each other and clearly distinguished.

0: Means clusters are indifferent, or we can say that the distance between clusters is not significant.

-1: Means clusters are assigned in the wrong way.

Silhouette Score = (b-a)/max(a,b)

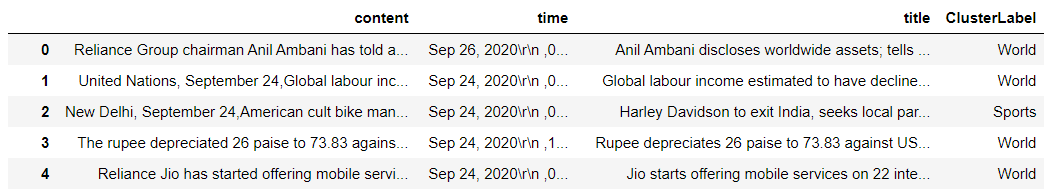
Where

a= average intra-cluster distance i.e the average distance between each point within a cluster.

b= average inter-cluster distance i.e the average distance between all clusters.

We have completed our project on BOT-1. It is recommending News articles in two different ways.

1. Recommending Latest News



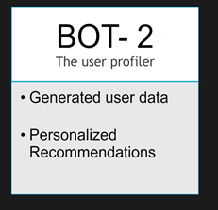
1. Recommending News articles randomly



**Limitation:-**This type of recommender system has a limitation that it cannot generate Personalized Recommendations. To overcome this problem we will design BOT-2.

**BOT-2:-**

Personalized recommender Systems' aim to identify news that best fits user preferences. These techniques are advantageous to users, as it enables the user to rapidly find what he needs.

****

**DATA GENERATION PROCESS**

Generating Click Stream Data

1. This process was an integration of various Distributions.

2. For the process of Making​ UserId ​we made the list of 4000 users using simple  Integers ranging from 1 to 4000.

3. For the purpose of making sessions for each user we used Geometric distribution  Because Geometric distribution has one analogy that each user must have a  minimum 1 session.  Because Geometric Distribution says​ the number of failures before you get a  success in a series of ​Bernoulli trials​.

4. Minimum Session for each user is 1 and Maximum Session are 27.

5. For the purpose of Generating Clicks Per session for each we used Poisson  Distribution because ​Poisson Distribution​ says:-  It is a tool that helps to predict the probability of certain events from happening  when you know how often the event has occurred. It gives us the ​probability ​of a  given number of events happening in a fixed interval of time​.

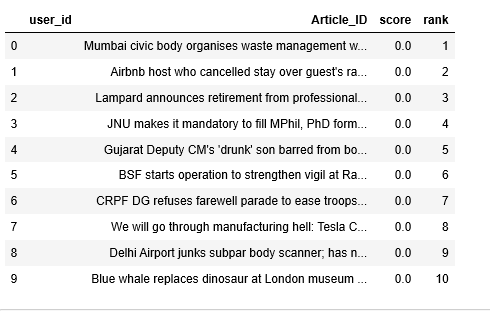
6. HOW IT WORKS :-  And Say ​Poisson distribution​ has generated 2 clicks in One session for Anyone user  So we have assigned those 2 clicks randomly to 10 news articles assigned to each  user in one session. We have randomly assigned these 2 clicks to articles because  this data is generated by Distribution not by actual clickstream data.  7. For the purpose of generating Average time spent by the user in reading we used 3  Gaussian mixture models​ with different mean and standard deviation to cater  the different reading speeds of each and also to take into account the variable of  Length of articles. Which ranges from 250 words to 8100 words.

**Personalized recommender system**

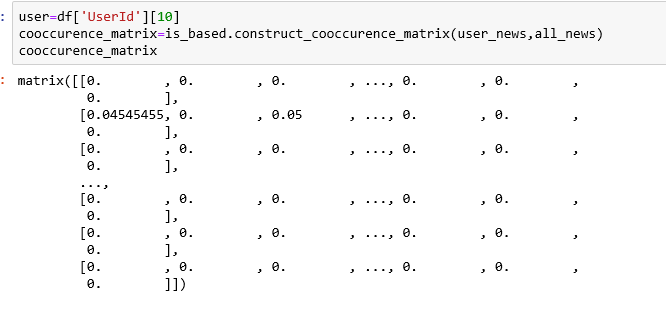
We will use three approaches to make personalized recommender systems.

* Content based filtering
* Collaborative filtering
* Hybrid methods

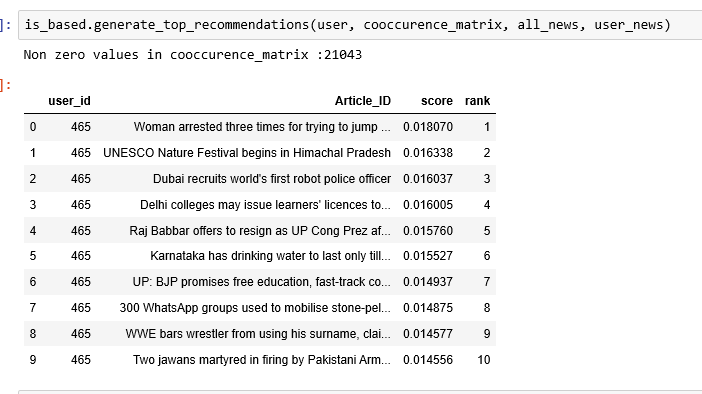
**Content based filtering :**  Suppose some likes science, movie articles and doesn’t like politics articles. Overtime the algorithm can accumulate this and figure out that the user has positive scores on science, movie articles and lower scores on politics articles. Content based filtering uses this information to map user ratings against the attributes of the news articles.



**Collaborative filtering :** In collaborative filtering, user ratings of other people are used rather than attribute data to predict and recommend Collaborative filtering builds on the idea of a user model that is a set of ratings and an item model that is a set of ratings. Combining the two models, we get a sparse matrix of ratings, some of its cells are filled and most are not. So, here there are two main tasks, one is to fill the empty cells or predict a rating and second is to choose a filled cell or recommend an item.

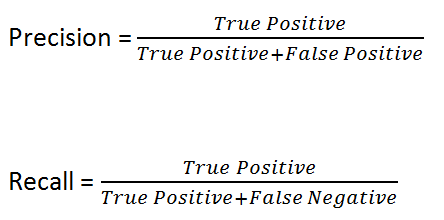


**Co-occurrence Matrix**



**Hybrid methods** : In these methods, a combination of two or more recommendation algorithms are used to take or maximize advantage of some techniques and avoid or minimize the drawbacks of another.

**Evaluation of models**

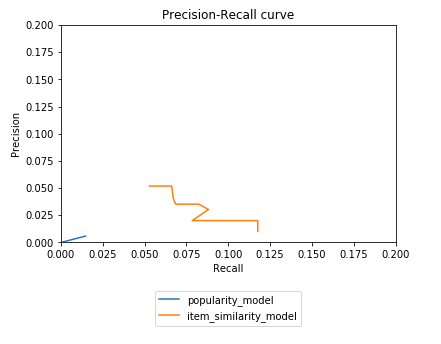


Precision is a good measure to determine, when the costs of False Positive is high. However, Recall actually calculates how many of the Actual Positives our model capture through labelling it as Positive (True Positive). Applying the same understanding, we know that Recall shall be the model metric we use to select our best model when there is a high cost associated with False Negative.

## Recommender Precision and Recall

Recommender precision and recall uses all recommended news over all users to calculate traditional precision and recall. Recommended news that was actually interacted with in the test data is considered an accurate prediction, and a recommended item that is not interacted with, or received a poor interaction value, can be considered an inaccurate recommendation. The user can assign these values based on their judgment.

The Precision and Recall plot is used to visualize the trade-off between precision and recall for one class in a classification.



**Future work**

**In future work, we would like to test our approach on a dataset where classes are implicit feedback rather than explicit feedback thereby performance would be more real world and we would also be able to observe any difference in performance of implicit versus explicit feedback.**

**There are many other options of algorithms that perform unsupervised learning. In the project, we have worked with only two of them, however it would be interesting to evaluate others. Another interesting approach would be to solve the problem of news recommendation as unsupervised learning problem. For example, models that are able to detect underlying variables that are enclosed in the news articles (i.e. LDA).**

**After model evaluation the final model should go through a thorough testing and then it should be deployed.**

**However, our approach is one of the many approaches and in the future we would like to propose different and more complex approaches with ultimate goal of designing a news recommendation system which are able to improve the state-of-the-art.**