My Goal?

I want to recommend healthy recipes to people. How? By checking their BMI index and calorie required for user. This is not something new. So what can you propose? Comparison of different algorithms and see which performs best.

Title – FeedMeRight – Comparison of Recipe Recommender systems.

Abstract:-----

With so many rapid changes happening around us, people are moving towards healthy lifestyle which includes choosing right food. Nowadays we are surrounded by overwhelming information which hinders the ability to choose right information. Recommendation system is a technique that would filter information and narrow it down based on our preferences and helps us to choose which we may like. Recommender systems have been widely used in e-commerce sites, social networking and entertainment industries. To help us in choosing healthy lifestyle recommender systems can be used in recipe domain to choose a right food for us not just based on user’s taste but also considering user’s lifestyle and calories required for user.

This thesis presents the design, implementation and evaluation of various recommender approaches within recipe domain. I will combine different recommendation techniques and use machine learning to recommend healthy recipes based on user profile. The overall goal is to discover which approach to recommend recipes offers better performance.

Keywords – Recommender system, Machine Learning, content-based, collaborative filtering, matrix factorization.

Contents

1. Introduction
   1. Motivation
   2. Problems Addressed and Challenges
   3. Thesis Contribution
   4. High level overview of proposed algorithm..(remove if repetitive)
2. Background

2.1 The World Wide Web

2.2 Information Retrieval

2.3 Information Filtering

2.4 Recommender System

2.4.1 Content-based Filtering

2.4.2 Collaborative Filtering

2.4.3 Hybrid approach

2.5 Recommendation Approaches

2.5.1 Basic approaches to find similar items

2.5.2 Vector Space Model

2.5.3 Extended Vector Space Model for Content-based Filtering

1. Implementation

3.1 Data Filtering

3.2 User-User CF (KNN)

3.3 Item-Item CF

3.4 Matrix Factorization

3.5

1. Experimental Results
   1. Results obtained from Content Based
   2. Results obtained from Collaborative Filtering
   3. Results obtained from Matrix Factorization
   4. Comparison
2. Experimental Analysis

5.1

1. Conclusion (& future work if I could think of something)

References

1. **Introduction**
   1. Motivation

The internet is huge network of machines that connects large number of computers together worldwide allowing them to communicate with any other computer. The World Wide Web is an information sharing model that is built on top of the internet in which information can be accessed or manipulated easily hence experiencing dramatic growth in increased usage of internet which results in BigData.

BigData is an exponentially increasing data with high volume, high velocity with variety. This huge amount of data has intrinsic value but it’s of no use until it’s discovered [bigdata]. One of the ways of finding value in BigData is analyzing it with its interrelated features such as new products, corresponding reviews, ratings and user preferences. Forming information from raw data is an entire discovery process that requires insightful analysis which would recognize patterns to predict user behaviors to recommend products [].

Handling BigData by manual process is very inefficient. More efficient way of processing such huge amount of data is automating the process of classifying, filtering data of user’s opinions, features, and preferences in order to understand and predict new set of related products.

Recommender system can be defined as a tool designed to interact with large and complex information spaces to provide information or items that are relevant to the user [recommender\_overview].

Nowadays recommender systems are widely used in variety of applications. Initially it applied for commercial use to analyze data. Amazon is a good example of such one of E-commerce websites. However, it is now present in several different domains including entertainment, news, books, social tags and some more sophisticated products where personalization is critical such as recipes domain. This paper would further discuss the different approaches for recipe domain to recommend healthy recipes based on user’s profile.

1. **Background**
   1. The World Wide Web
   2. Information Retrieval
   3. Information Filtering
   4. Recommender System

A recommender system is an Information Filtering (IF) system that provides or suggests relevant items to user based on the user profile and preferences.

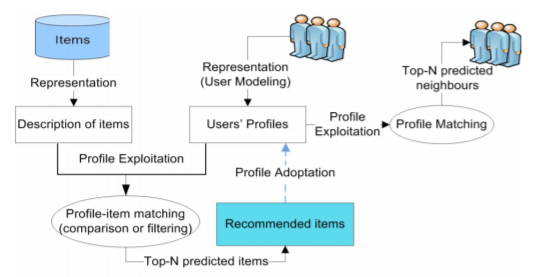


Fig – General Model of Recommender process ref - <https://link.springer.com/chapter/10.1007%2F978-981-10-0557-2_112>

Traditionally there are two basic models of recommender systems as below

* + 1. Content Based Filtering

In Content based method algorithm, user preference is considered based on item description. The rating and buying behavior of users are combined with content information available in the items. The main aim of content based filtering is to create profile for each item and each user to find similar items the user is looking for [contentbased].

In this algorithm each user’s information can be stored in vector form which contains past behavior of the user. This vector is known as profile vector or user profile. All the information about item is stored in item vector / item profile which contains all the details about item specific attributes. Based on similarity score between user profile and item profile most relevant items are recommended to user.

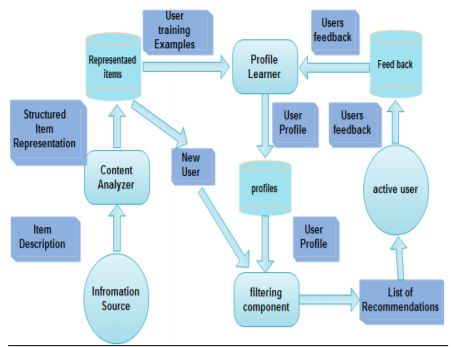


Fig. – Content based recommendation architecture [figures]

Ref - <https://www.researchgate.net/publication/331063850_Recommender_Systems_Challenges_and_Solutions_Survey>

Advantages of content-based recommenders are –

Content-based recommender systems are heavily reliable on the contents of the items that have been rated by the user. So, while making recommendations, this approach would consider user’s taste and accordingly recommend an item that matches user’s preferences. Generally, most popular items dominate less popular items. But this approach will not miss less popular item if it matches the user’s unique taste [contentbased].

Disadvantages of content-based recommenders

User profiles are generated based on rated items. But for any new user who has not rated any items yet, user profile will be empty. In that case, recommending perfect item that matches to user’s taste is difficult as system does not have user taste information. This problem is known as cold start. Also, to understand each items feature, system needs to examine content of every item. Therefore if number of items rises quickly, performance of the system decreases [contentbased].

2.4.2. Collaborative Filtering (CF)

Collaborative filtering uses other users’ behavior in the system to predict and recommend items. It depends on user’s contribution such as ratings, reviews which considered as filter for user preference information. The fundamental idea of collaborative filtering is it selects other users’ opinions and aggregate in such way that it provides prediction for active user based on his preferences [CF]. The main source of input for this algorithm is in the form of matrix of collected user-item ratings. Based on this input it provides recommendations as an output. The first step of output is to predict ratings for items that user may like. Second step is to recommend a list of top rated items as top-N items.

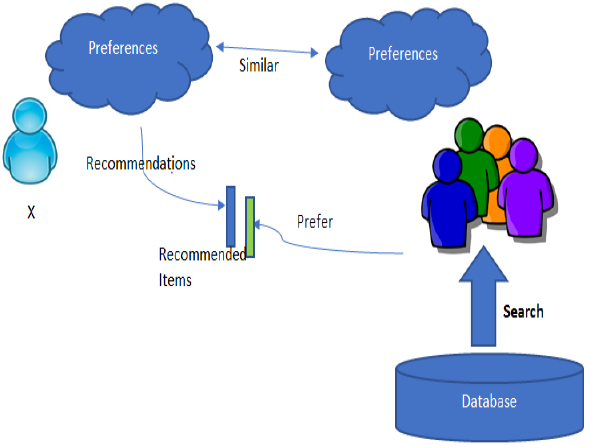


Fig . Collaborative filtering reference – report close to me – [cf\_figure] <https://www.researchgate.net/publication/328231954_Comparative_analysis_of_recommender_systems_and_its_enhancements>

Collaborative Filtering is broadly divided into 2 categories.

* + - 1. Memory-Based (user based)

A memory-based collaborative filtering approach predicts item ratings based on ratings given by different users for an item. There are primary two forms of memory-based collaborative filtering.

2.4.2.1.1 User-user CF:

Similarity between users is calculated based on how similarly they rate several items. It finds other users whose ratings are similar to active user and use their ratings on other items to predict what active user may like. Thus it recommends items to the users that are most preferred by similar users.

Consider example of user and ratings given by users to different recipes. This algorithm will find similarity between each user based on the ratings they have given to the recipes in the past. The prediction of a recipe for a user u is calculated by computing weighted sum of the user ratings given by other users to recipe i.

The prediction for recipe I is given as below:

https://cdn.analyticsvidhya.com/wp-content/uploads/2018/05/Screenshot-from-2018-05-29-20-15-31.png

Where,

Pu,i = prediction of recipe i

Rv,I = rating given by user v to recipe i

Su,v = similarity between users.

The similarity between users can be calculated with the help of several methods described in the section 2.4.3

2.4.1.2 Item-item CF:

* + - 1. Model-Based (item based)

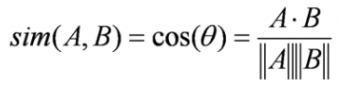
Write something about it

* + 1. Approaches to Find Similarity:

There are several methods available to calculate similarity score.

* + - 1. Cosine Similarity –

In this method, cosine of the angle between profile vector and item vector is calculated. Consider A and B are profile vector and item vector respectively, the similarity between them can be calculated as per below formula:



The value of cosine angle ranges between -1 to 1. Lesser the angle, less distance hence more similarity as cos(0) = 1. Then items are arranged in descending order and recommended to user.

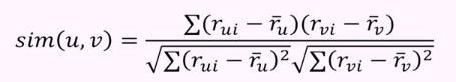
* + - 1. Euclidean Distance

If we plot similar items in n-dimensional space, then they will fall under close proximity. In that case, we can calculate distance between items with Euclidean distance formula which is given by:

https://cdn.analyticsvidhya.com/wp-content/uploads/2018/05/2zjgw1x1.png

* + - 1. Pearson’s Correlation

Person’s correlation helps in finding correlation between similar items. Correlation on higher side implies more similarity. It can be calculated as below:



[All formulas referece – programming collective intelligence book]

**Writeup reference - Health-aware Food Recommender System**

[1] Resnick and Varian. Acm press, recommender systems, volume 40., 1997. URL <http://doi.acm.org/10.1145/245108.245121>.

[bigdata] <https://www.oracle.com/big-data/guide/what-is-big-data.html>

[Khusro2016] <https://link.springer.com/chapter/10.1007%2F978-981-10-0557-2_112> Khusro2016

[recommender\_overview]<https://www.researchgate.net/publication/220604600_Recommender_Systems_An_Overview>

[contentbased]<https://www.researchgate.net/publication/236895069_Content-Based_Recommendation_Systems>

[contentbased\_architecture] <https://www.researchgate.net/publication/331063850_Recommender_Systems_Challenges_and_Solutions_Survey>

[CF] <http://files.grouplens.org/papers/FnT%20CF%20Recsys%20Survey.pdf> Reference for collaborative

filtering

[cf1] <https://link-springer-com.summit.csuci.edu/referenceworkentry/10.1007/978-3-319-32001-4_274-1> -Collaborative filtering by Ashrf (reference for memory based and model based collaborative filtering )

[7]<https://www.researchgate.net/publication/328231954_Comparative_analysis_of_recommender_systems_and_its_enhancements>

\cite{Resnick1997} \cite{CF} \cite{bigdata} \cite{Khusro2016} \cite{recommender\_overview} \cite{figures} \cite{contentbased} \cite{CF} \cite{cf\_figure}

Michael D. Ekstrand, Joseph A Konstan, coursera, Introduction to Recommender Systems: Non-Personalized and Content-Based https://www.coursera.org/learn/recommender-systemsintroduction/ lecture/ZkG45/summary-statistics-i.

### From Indian\_Cusine\_RecipeReco\_BasedonIngredients

#### I might not need point A

A. Web Scraping Web Scraping[6] (Scraping or Web Data Extraction or Web Harvesting[7]) is a system utilized to fetch a lot of information from sites whereby the information is extricated and spared to a nearby record in your PC or to a database in the table (spreadsheet) design. Web scraping is the way toward fetching information from sites. All the activity is completed by a bit of code which is known as a "spider". In the first place, it sends a "GET" question to a particular site. At that point, it parses a HTML record dependent on the got outcome. After it's done, the scrubber looks for the information you need inside the report, and, at long last, changes over it into the predefined format.

B. Data Cleaning Data cleansing or data cleaning[8] is the way toward distinguishing and adjusting (or expelling) degenerate or off base records from a record set, table, or database and alludes to recognizing fragmented, mistaken, off base or unessential parts of the data and afterward supplanting, altering, or erasing the filthy or undesirable data. Data cleansing might be performed intelligently with data wrangling instruments, or as group preparing through scripting.

C. Bags-of-Words The bag-of-words model [9] is a way of representing text data when modeling text with machine learning algorithms. The bags- of-words demonstrate is a rearranging portrayal utilized in natural language processing and information retrieval (IR)In this model, a content, (for example, a sentence or an archive) is spoken to as the sack (multi-set) of its words, dismissing syntax and even word request yet keeping variety.

Todo

1. Convert project into maven
2. Save each recipe information into json document from edamam API.. Loop through each unique recipe.
3. Will get data for each recipe. And I have data for user and recipe rating data.
4. From that I think I need to prepare user profile. Not sure how to calculate which ingredients each user likes. Check if we can get that info from allrecipes.com WebScrapor tool used in one of the paper was – spider plugin

### link for collaborative filtering.

<https://towardsdatascience.com/how-to-build-a-simple-recommender-system-in-python-375093c3fb7d>

## link to check if we can use it to evaluate recommender engine

<https://github.com/MaurizioFD/RecSys2019_DeepLearning_Evaluation>

Evaluation of recommender systems..

######## How the workflow should be –

1. Enter User Id:
   1. getRecipeFeatures() get called. It creates sparse matrix of all recipes based on it’s ingredients as features. Creates a .csv file for sparse matrix named recipe\_feature\_matrix.csv
   2. Whenever any new recipe will get added to the dataset, we need to update recipe\_feature\_matrix.csv file.
2. build\_user\_profile() will get called. It will read recipe\_feature\_matrix.csv wil get read.

In this method, user profile will get created for provided user\_id based on recipe\_feature\_matrix.csv file.

1. Based on user profile, similarity score will get calculated between user profile and all recipes. Sort the results in descending order.
2. Get the BMI of user and the required calorie range for user. And add 10 results from the sorted resultset such that calories of those recipes would fit in the required calorie range for user.

############ Evaluate the Engine ###############

Prof. thought on Evaluation

Recall -

and precision

we are defining what is good recommendation.

Cold start problem. In that case content based filtering would be more effective.

Full cross validation on train and test set.

To create vnv for kera –

1. Run conda create -n venv\_name and source activate venv\_name, where venv\_name is the name of your virtual environment.
2. Run conda install pip. This will install pip to your venv directory.
3. Find your anaconda directory, and find the actual venv folder. It should be somewhere like /anaconda/envs/venv\_name/.
4. Install new packages by doing /anaconda/envs/venv\_name/bin/pip install package\_name.

This should now successfully install packages using that virtual environment's pip!

Steps followed by me –

conda create –n conda\_venv

created environment location - C:\Users\jpall\Anaconda3\envs\conda\_venv

to activate virtual env in windows - C:\Users\jpall\Anaconda3\Scripts\activate.bat

files will be created at new place - C:\Users\jpall\Anaconda3\envs\conda\_venv\conda-meta

to install new packages –

C:\Users\jpall\Anaconda3\envs\conda\_venv>pip install tensorflow

### Links of Articles –

1. Theory - Collaborative filtering nearest neighborhood and matrix factorization - <https://towardsdatascience.com/intro-to-recommender-system-collaborative-filtering-64a238194a26>
2. Theory –Content based and collaborative based and matrix factorization and Evaluating Algortihms - <https://towardsdatascience.com/recommendation-systems-models-and-evaluation-84944a84fb8e>
3. Theory – Evaluating Recommender Systems –Actually everything – Content, Collaborative, matrix <https://medium.com/fnplus/evaluating-recommender-systems-with-python-code-ae0c370c90be>
4. Understanding Matrix Factorization - <http://nicolas-hug.com/blog/>
5. Evaluation code using surprise - <https://towardsdatascience.com/evaluating-a-real-life-recommender-system-error-based-and-ranking-based-84708e3285b>
6. <https://www.offerzen.com/blog/how-to-build-a-content-based-recommender-system-for-your-product>
7. Collaborative with evaluation by simple code - <https://www.ethanrosenthal.com/2015/11/02/intro-to-collaborative-filtering/>
8. Surprise basic good example of collaborative filtering- <https://realpython.com/build-recommendation-engine-collaborative-filtering/>
9. Tensorrrec all details.. huge blog with code -<https://towardsdatascience.com/getting-started-with-recommender-systems-and-tensorrec-8f50a9943eef>
10. Surprise all algorithm code in notebook –

<https://github.com/susanli2016/Machine-Learning-with-Python/blob/master/Movielens%20Recommender%20Metrics.ipynb>

<https://nbviewer.jupyter.org/github/NicolasHug/Surprise/blob/master/examples/notebooks/KNNBasic_analysis.ipynb>

1. <https://towardsdatascience.com/building-and-testing-recommender-systems-with-surprise-step-by-step-d4ba702ef80b>
2. Hybrid Implementations –

<https://www.kaggle.com/rounakbanik/movie-recommender-systems>

<https://www.kaggle.com/robottums/hybrid-recommender-systems-with-surprise>

1. Writeup reference pointwise - <https://github.com/jalajthanaki/Movie_recommendation_engine/blob/master/Intro_recommendation_system.ipynb>
2. # ## Source - <https://towardsdatascience.com/evaluating-a-real-life-recommender-system-error-based-and-ranking-based-84708e3285b>

#### Command to convert Jupyter ntebook to python file –

jupyter nbconvert --to python allrecipes\_collaborative\_item\_item.ipynb

###### Tools to use to link references in the report

1. Mendeley
2. Jabref

############## Papers with summary ===🡺 what Should I follow –

1. Health Aware Food Recommender System – Extends Matrix Factorization by adding user’s BMI factor. Algo details in 2.
2. Using Tags and Latent factor in a Food Recommender System – matrix factorization without adding BMI factor. Best explanation about offline evaluation.