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

**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.

## For similarity measures

<https://pdfs.semanticscholar.org/943a/e455fafc3d36ae4ce68f1a60ae4f85623e2a.pdf>

© Springer Science+Business Media Singapore 2016 1179

K.J. Kim and N. Joukov (eds.), Information Science and Applications (ICISA) 2016,

Lecture Notes in Electrical Engineering 376,

DOI: 10.1007/978-981-10-0557-2\_112

Recommender Systems: Issues, Challenges,

and Research Opportunities

Shah Khusro, Zafar Ali and Irfan Ullah

© Springer Science+Business Media Singapore 2016 1179

K.J. Kim and N. Joukov (eds.), Information Science and Applications (ICISA) 2016,

Lecture Notes in Electrical Engineering 376,

DOI: 10.1007/978-981-10-0557-2\_112

Recommender Systems: Issues, Challenges,

and Research Opportunities

Shah Khusro, Zafar Ali and Irfan Ullah

© Springer Science+Business Media Singapore 2016 1179

K.J. Kim and N. Joukov (eds.), Information Science and Applications (ICISA) 2016,

Lecture Notes in Electrical Engineering 376,

DOI: 10.1007/978-981-10-0557-2\_112

Recommender Systems: Issues, Challenges,

and Research Opportunities

Shah Khusro, Zafar Ali and Irfan Ullah

© Springer Science+Business Media Singapore 2016 1179 K.J. Kim and N. Joukov (eds.), Information Science and Applications (ICISA) 2016, Lecture Notes in Electrical Engineering 376, DOI: 10.1007/978-981-10-0557-2\_112 Recommender Systems: Issues, Challenges, and Research Opportunities Shah Khusro, Zafar Ali and Irfan Ullah

##### Hybrid type –

<https://www.bluepiit.com/blog/demystifying-hybrid-recommender-systems-and-their-use-cases/>

## BMI DOCS

Rerefences from Paper –

1. Health\_Aware\_Reco – [3]
2. Bringing healthy food – [6] – BMR Formula

<https://www.kbcc.cuny.edu/academicdepartments/physci/Documents/labmanuals/sci70/bmi.pdf>

Impact of food craving and calorie intake on body mass index (BMI) changes during an 18-month behavioral weight loss trial – Check on Springer Link

1. <https://www.cdc.gov/healthyweight/losing_weight/index.html>
2. <https://www.ncbi.nlm.nih.gov/books/NBK56068/table/summarytables.t4/?report=objectonly>
3. <https://health.gov/our-work/food-nutrition/2015-2020-dietary-guidelines/guidelines/appendix-7/>
4. 2002, "Dietary Reference Intakes for Energy, Carbohydrate, Fiber, Fat, Fatty Acids, Cholesterol, Protein, and Amino Acids," Food and Nutrition Board, Institute of Medicine.
5. Ainsworth B.E., 2002, January, "The Compendium of Physical Activities Tracking Guide," Prevention Research Center, Norman J. Arnold School of Public Health, Univ of SC.
6. Actual Formula with source -- <http://www.checkyourhealth.org/eat-healthy/cal_calculator.php>
7. [**http://health.howstuffworks.com/question457.htm**](http://health.howstuffworks.com/question457.htm)
8. [**http://www.bmi-calculator.net/bmr-calculator/bmr-formula.php**](http://www.bmi-calculator.net/bmr-calculator/bmr-formula.php)
9. [**http://www.bmi-calculator.net/bmr-calculator/harris-benedict-equation/**](http://www.bmi-calculator.net/bmr-calculator/harris-benedict-equation/)
10. Bough KJ, Rho JM. [Anticonvulsant mechanisms of the ketogenic diet](https://doi.org/10.1111/j.1528-1167.2007.00915.x). Epilepsia. 2007;48(1):43-58. doi:10.1111/j.1528-1167.2007.00915.x
11. Nordmann AJ, Nordmann A, Briel M, et al. [Effects of low-carbohydrate vs low-fat diets on weight loss and cardiovascular risk factors: a meta-analysis of randomized controlled trials](https://doi.org/10.1001/archinte.166.3.285). Arch Intern Med. 2006;166(3):285-93. doi:10.1001/archinte.166.3.285
12. Harvie M, Wright C, Pegington M, et al. [The effect of intermittent energy and carbohydrate restriction v. daily energy restriction on weight loss and metabolic disease risk markers in overweight women](https://doi.org/10.1017/S0007114513000792). Br J Nutr. 2013;110(8):1534-47. doi:10.1017/S0007114513000792

Source of categorizing diet lables.

1. <https://www.fda.gov/food/new-nutrition-facts-label/how-understand-and-use-nutrition-facts-label>  
     
   Read More <http://nutritiondata.self.com/tools/calories-burned#ixzz6KsEKVQFh>.
2. Read More <http://nutritiondata.self.com/tools/calories-burned#ixzz6KsEKVQFh>

<https://medium.com/@bond.kirill.alexandrovich/precision-and-recall-in-recommender-systems-and-some-metrics-stuff-ca2ad385c5f8> Recall And Precesion

## Content based references articles for diagrams and descripts –

<https://medium.com/towards-artificial-intelligence/content-based-recommender-system-4db1b3de03e7>

## Content based, Collaborative, Hybrid architecture and flow – youtube video –

<https://www.youtube.com/watch?v=TFi7WXpaIiY>

Converting formula from Cosine similarity to prediction ratings

<https://www.researchgate.net/post/How_would_you_recommend_shifting_from_a_Similarity_Scores_to_Rating_Prediction_system>

### My Writeup References –

1. Evaluating Collaborative Filtering Recommender Algorithms: A Survey - <https://ieeexplore.ieee.org/document/8550639?denied=>
2. Recommendation systems: Principles, methods and evaluation - <https://www.sciencedirect.com/science/article/pii/S1110866515000341#b0380>

### 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>
3. ## Recall Precision

<https://towardsdatascience.com/recommendation-systems-models-and-evaluation-84944a84fb8e>

1. Matrix factorization

<https://cloud.google.com/solutions/machine-learning/recommendation-system-tensorflow-overview>

#### 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.

################# THOMS Review #############################

1. Motivation – www - should we replace it with http? –The main difference between WWW and HTTP is that they refer to different concepts. Simply put, HTTP is the protocol that enables communication online, transferring data from one machine to another. WWW is the set of linked hypertext documents that can be viewed on web browsers (such as Firefox, le Chrome, and more).
2. it did not help in changing user's behaviour towards helathy lifestyle. How healthy behavior would measured by the recommender

########### FLOW #####################

Hello All, First of all I would like to thank you all for joining us today for my thesis defense on short notice. Today I will be talking about FeedMeRight - Recipe recommender system. The aim of this thesis is to recommend healthy recipes to users. To build this recommender system, we have combinined traditional approaches such as content-based filtering and collaborative filtering.

**Contents**

Before diving into it further details, I will talk more about myself and why I chose this topic, what all work is already done in this area, objective of this thesis and what all experiments are performed to come up with varying approach. Then, results of all experiments and conclusion with future work.

**Motivation**

Before coming to Master’s program I was working in Bigdata and Hadoop technology. I learned more about data ceaning and analysis from Advance database class taught by Prof. Thoms. And it inspired me to work in data driven field. That’s the reason I presented my 1st graduate seminar on Bigdata and Hadoop Technology.

**BigData**

Bigdata is any data which generates at high volume, high velocity with variety. This image represents data growth over the period of time. The amount of data generation was very less before internet. With technology growth, the amount of data generation started to grow exponentially and around 1800 TB of data is being generated every second. This data has tremendous value. To find that value we need to build application on top of it and one of the great applications of bigdata is to build recommender systems. Prof. Thoms guided me to work in this area. Information filtering systems that provide a solution for the problem of information overload. Recommender systems can be built for different domains. One of the important domains is food because what we eat plays an important role in an individual’s health.

\*\*Considering current situation where people are warming up to the idea of building immunity and maintaining healthy diets.

**Health factor -**

The research of World Health Organization shows that, globally 39 % adults are overweighed and 13 % adults are obese. Overweight and obesity causes chronic diseases like diabetes, blood pressure. It can be controlled by maintaining a weight. In order to maintain weight, people need to consume the proper amount of calories based on their needs. Most of the people don’t know how much calories they should consume. Also, today’s busy lifestyles don’t give people enough time to think about it. Fortunately, we live in era of internet and information where we get information about health and recipes. But that information is so overwhelming that its difficult to understand which one to choose. Which one is good for me? The result of this situation is, people choose unhealthy and easy options. This problem can be solved by introducing such recipe recommender systems which will consider user’s taste (what user likes) as well as user’s health factor(how much calories user should consume).

**Background –**

There is some work that has been already done in this area to understand what user may like. Or what is good for user’s health.

**Literature Review –**

The research of Freyne and Berkovsky used content base technique to recommend recipes based on ingredients similarity. Content based technique recommends only those recipes whose ingredients are similar. We will see more details about it later. But this technique considers only user’s taste. The same research extended by Morgan Harvey, who considered +ve and –ve weighting factors for ingredients. But In these 2 techniques, only user’s taste has been considered and not the health factor.

There was one more research done by Chun-Yuen Tend, Yyu-Ru Lin, and La-da adamic’s where health factor was considered by substituting ingredients.

Elahi proposed a system with matrix factorization and active learning algorithm whose performance was very good. But his approach only considered user’s taste.

But the same research was extended by Mouzhi Ge, Francesco Ricci, and David Massimo by incorporating calorie count. Where along with user’s preferences, health factors are also considered.

**Objective –**

This work inspired me and I am extending this work using different approach to build varying recommender system that considers user’s preferences as well as user’s health. The objective of this thesis is to perform comparative analysis on different recommender techniques in recipe domain, and design and develop varying recipe recommender system considering user’s preferences and health.

**Implementation –**

To achieve this goal, few experiments have performed. And to perform those experiments, some data preprocessing steps were necessary.

**Data Source –**

The data is recieved from Kaggle platform. Kaggle is wellknown platform to find and publish datasets. The data consists of recipes and user interactions from Allrecipes.com website. The recipes file has 6 coolumns as shown in this image such as recipe\_id – its unique id given to the recipe., recipe\_name, image\_url, ingredients – it has all ingredients required for each recipe., cooking\_directions have all instructions about how to cook that recipe. and nutritions has calories and other nutrition information for that recipe..

There were 2 different files available for ratings. I merged those 2 files and removed duplicate interactions and considered only those recipes which are rated by users. This ratings file denotes each user interaction with recipes. How much rating user has given to the recipe. After applying all basic data cleaning process we got around 44k recipes, 20k users 655 thousand interactions.

**VM on GCP –**

When we are bulding models with such huge amount of data, it requires high memory for computation and to fit that model in memory. In order to reduce execution time and to perform high computational operations, I chose to run the application on Virtual Machine. I created VM for FeedMeRight project on GCP. It uses 32 GB memory, 8 core cpu, and 100gb ssd harddisk.

**Features Extraction –**

Next part is to extract some features which are required for content-based filtering.

Content-based filtering relies on contents or features of the items – in our caseitems are recipes. There are 4 features we have considered - ingredients, cook-method, diet laabels and calories.

**Ingredients Extraction –**

First feature is Ingredients. To extract ingredients from recipes, nltk library has been used. It tokenizes sentences into words. To remove irrelevant words and to make ingredients very specific, recipe stopwords are applied. These recipe stopwords are customized set of words. For example, white from white eggs, thawed from thawed rotis. These white, thawed, frozen words are stores in one set and removed from ingredients as they are not relevant to ingredients. If we compare first 2 images, from 1st row, frozen and thawed have been removed. Next, on these clean ingredients, lemmatization has been applied. Lemmatization is a technique that converts word into its existing form. For example, potatoes will get convert into potato. Here we can see, eggs has converted into egg.

So, that’s how ingredients are extrcted.

**Cook Method Extraction –**

Next feature is cook method. To extract cook methods, we have used cooking direction column. It has all instructions about cooking specific to recipe. Its a combination of ingredients and cooking methods. Again, using nltk library these instructions converted into words. We have predefined glossary of cook methods. Common words from predefined glossary of cook methods and tokenized words are considered as cook methods.

**Diet Labels Extraction –**

Next feature is diet label. For diet labels, we need to understand how nutrients can be categorized. Broadly they are divided into high-protien, high-fiber, lowfat, lowcarb, lowsodium and balanced categories. But these categories are based on %DV which is called percentage daily value for each nutrient. If this %dv value is 5% or less than 5% then we categorize that nutrient as low-nutrient. And if this %DV value is 20% or more then we categorize it as high nutrient. For example, in this image percentdaily value for carbihydrates and sodium is 3 which is less than 5% hence this recipe fall under lowcarb and low sodium category.

**Calorie Extraction –**

Calorie extraction of a recipe was a simple process. In nutrition column, everything is stored in nested dictionsry. So, we extracted calorie from this column and stored in newly created calories column as shown in this figure. We need these calories of all recipes to compare with user’s calorie requirements.

**User Information –**

Now to calculate calories required for a user, we have used Harris Benedict equation. It calculates Basal Metabolic Rate of a user based on height, weight, age and gender. Once we get BMR, we can get calorie intake required for a user based on his activity level. This table represents calorie requirement needed for a user based on his activity lifestyle.

That was all about data that we need to use for further experiments and analysis.

**Content-Based Filtering Workflow –**

The 1st part of the experiments is to undertsand content-based filtering. In Content-based filtering we create two vectors. One is for user and another is for an item. In our case an item is a recipe. Once we get user profile and recipe profile, we can find similarity score between these 2 vectors using cosine similarity. This similarity score helps us to find the most preferable recipe for a user. We will have more clear idea once we go through this flowchart and how this flowchart works for a single user.

This user ratings contains all users and their ratings for different recipes. From this dataset we can filter recipes which are rated by single user.

Similarly we have all recipes here. And for all recipes document-term matrix is formed. Document-term matrix is basically tf-idf values of terms for all recipes. And that term can be any term from our features. For example, if our feature is ingredients, then columns of this matrix will be all unique ingredients and recipes will be rows. And the value of cell will be tf-idf value of the unique ingredient for that recipe. Now, this document-term matrix actually represents all recipe profiles. For our targated user, we can filter out rated recipe’s profiles. User-recipes profiles represents recipe profiles for those recipes which are rated by a user. Next, to contruct a user profile, we calculated weighted average all these recipe profiles by the ratings of a user. We normalized that weighted average and calculated user profile.

Now, we have user profile and recipe profiles. Next, the similarity score is calculated between this user profile and all recipe profiles from this set. User-recipe similarity stored in descending order of similarity score. After that calorie filter was applied. Here we filter only those recipes whose calories are <= user’s calorie intake requirement. And the resultant recipes are considered as recommended recipes for our targated user.

There are 3 experiements are performed based on content-based filtering using features.

**Content-Based using ingredients –**

From which 1st experiment is perfomred using ingredients feature. It’s the exact same workflow but the difference here is document- term matrix is formed using ingredinets features. So, here we are getting all user-recipe profiles based on ingredients feature. And from all these recipe profiles we are constructing user profile by weightage average. And the generated similarity score is also based on ingredients feature. On which we apply calory filter and present resultant set as recommended recipes. So here, we are considering only one feature and that is ingredients.

In next experiments, we are considering 2 feature together – Ingredients and cook methods.

**Content-Based using ingredients and Cook Methods –**

Cook methods are helpful to aggregate recipes together. The way we build vector space model for ingredinets feature, similarly one more vector space model is built for cook methods feature..Where document-term matrix is formed beased on cook methods feature. Constructed user profile is based on cook methods feature. And finally we have user-recipe similarity based on cook-methods feature. Now, the similarity generated from vector space model for cook methods and similarity generated from vector space model for ingredients are added together to calculate average similarity score. On this average similarity score, calorie filter was applied and resultant set was recommended as prefered recipes.

**Content-Based using ingredients and Cook Methods and Diet labels –**

Exactly in same we have incorporated another feature - diet lables. The new vector space model created for it and the similarity score generated from this model was added to the score generated by vsm for ingredients and vsm for cook methods. And all the scores added together to calculate the average score. On this average similarity score, calorie filter was applied and resultant set was recommended as prefered recipes.

**Results comparison of content-based models**

Now, if we compare the results of these 3 experirmts, we can see that the recall, precision and accuracy of the last model - which has all three attributes is highest. As we keep adding more relecvant content to this model, content based performs really well. So, the last model of content based with all three attributes is considered for next experiment.

**Collaborative Filtering (CF) using SVD**

Our 4th experiment was done usign Collaborative filtering using SVD. In collaborative filtering iteself so many different algorithms exists from which we used SVD which is based on matrix factorization. For Collaborative filtering we don’t need to use any features or attributes of an item like we need it in content based. The user-rating matrix is cnverted into sparse matrix. We send it as an input to SVD algorithm. SVD algorithm runs Principle Component Analysis (PCA) and it returns factors of rating matrix . Dot product of these factors gives us rating matrix with predicted ratings. Further normalization was done on these raings and calorie filter was applied. And resultant set of recipes were offered as recommended recipes.

**Hybrid Approach**

**Application Demonstration**

This application is console based application. But I really want to show how it works. And I am planning to move it into a mobile application. This is my Vm on Google Cloud Platform where I am running my applciation. At this point I am going to run it for very small amount of data because to run it on larger dataset takes a lot of time. On running, application prompts for these command promt options. 1. If we opt for 1st option to create recommendation models, the application will create recommendation models for all algorithms. Content-based, collaborative and hybrid. And it saves those models using .mdl file. I already have built models. So here I will opt for 2nd option which is getting recipe recommendations for existing users.

**Results Comparison**

**Conclusion**

**Application Demonstration**

**Future Work**

My Questions for myself?

* 1. How to calculate calorie intake per meal?
  2. Range for adults height and weight?
  3. result set – replace slight increase with percentage.