IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 24, Issue 2, Ser. III (Mar. –Apr. 2022), PP 58-65
www.iosrjournals.org

Hybridized Search and Recommendation Propagation: Intensifiers to Agricultural Commercialization

Reetika Bahl
Department of Computer Science and Engineering
United Institute of Technology Prayagraj, India

Sanyam Srivastava
Department of Computer Science and Engineering
United Institute of Technology Prayagraj, India

Abstract
Nutrition is a cardinal factor to every human being. Food is the fuel to the body and hence, our lives depend on the community which feeds us. The question is, the people who feed all, Are they all getting proper nutrition and a good lifestyle? Today in the fast paced scenario, the vendors of different sectors could also use the platform of e-commerce to increase their profitability. As a consequence, the community which staggers behind to find means of commercialization is our Farmers' community. toTheFarmers is a multipurpose cross platform application intended to intensify the commercialization of the Farmers' community. It is an attempt so that the community can collaborate with each other and escalate the opportunities for them, gaining multiple sources which as a result could diminish the amount of manual labour for them and foster their living. The aim is to bridge the entities of the system as close as possible in accordance to their real time physical distance. The same factor is being used to qualify the search results. Moreover, the recommendation pattern issues have been taken to resolve by the inclusion of location and timestamp to the algorithm.

Keywords
Search Space Optimization, Law of Haversines, Bucketing, Collaborative Filtering, Real Time Distance based Sorting

----
Date of Submission: 05-04-2022 Date of Acceptance: 20-04-2022

I. Introduction

The population of the country today also, approximates more than fifty percent of the total in the sector of Agriculture. For the occupation having such a substantial range, it is required that a greater transparency should come into it. Moreover, the conventional methods of marketing and labour needs to be contemplated in order to incorporate the factor of scalability into it. There can be numerous obstructions which could affect a farmer from getting the profit he wants. May be due to inappropriate or limited places to sell, loss of money

in mediation of product, a large field but no workers to work upon, et al. Keeping all of these cases and possibilities, the application is delineated in a manner to be utilized by every farmer, be it a field worker, a seller or, a farmer who pays workers on daily wage so as to work and plough their field. Every entity can use the system to fulfil their needs. As an out-turn a seller gets buyers from the system, a farmer community gets workers from the system, similarly workers find field jobs from the system [See Fig.1]. The objective of the application is to produce search results and recommendations in account to the real time distance between the seller and buyer or the worker or agricultural landlord. Due to this a buyer or worker has to travel least distance to find what he wants matching the order of search results. For the recommendation of items to users, the system considers collaborative filtering algorithm, traditionally being a user - item based filtering. It functions in mapping what ratings a user gives to an item and based on it, suggests items to users having a greater propensity to be bought, effecting the profitability of sellers. The algorithm contains certain setbacks which are taken into account and monitored on the basis of location and timestamping to add an extra feature to increase the effectiveness of the algorithm to the new stimuli.

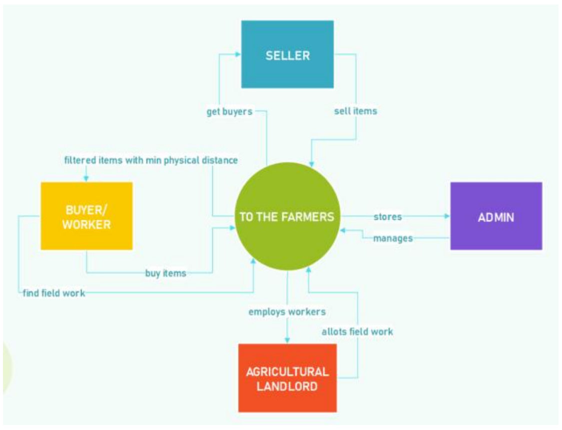


Fig.1. Context Level Data Flow Diagram

II. Search Space Optimization

A search space is defined as the set of all possible elements which match to a predicate entered by the user. In other words, it is the set of all possible situations of the problem that we want to solve could ever be in. Combination of all the possible values for all the variables related with the problem [1]. The generation of data taking place today is quite prodigious. Therefore, for a software dealing with the real world data, it becomes indispensable to keep specific algorithm and criteria to filter the search results concerning to show appropriate and useful results from the application to the user. Coming to the domain of *toTheFarmers* specific search space, the search space filtering takes place in three steps.

a) **Bucketing** of items at the time of addition by sellers
b) **String Matchup** to fetch out items from the respective buckets at the time of search by buyers
c) Using **real time distance based sorting** of items so that buyers could reach sellers as soon as possible

2.2 Bucketing

A bucket data structure is a data structure that uses the key values as the indices of the buckets, and store items of the same key value in the corresponding bucket [2]. Also known as fixed cells that is especially suitable for queries that cover small windows. A dynamic bucketing structure that maintains a two-level director structure and uses a directory doubting and merging technique called extendible hashing. This technique allows directories to expand without sacrificing performance as more objects are inserted into the structure [3]. As of the problem statement, we could create precomputed stored buckets in order to initialize items into a specific cluster. Using bucketing, the overall search space could now be divided among these precomputed sets of products. Now, each item will lie in its specific bucket and the retrieval and search of the item will become much efficient as compared to searching in a heap of unorganized items. At the time, when the seller entity adds an item to the system, there only a dropdown can help him to add the type of product it wants to add in. The items can be ranging from *zaid* crops like cucumber as per quantity to the extent of farming accessories like sprayers, handmade – manures, et al. The diagram is a part from the system's entity relationship diagram which represents the item entity, its key and non – key attributes [See Fig.2]. From the diagram it can be inferred that the item being an object could store its itemType which could be further used for mapping, storing and hashing it into respective pre – built buckets expeditiously.

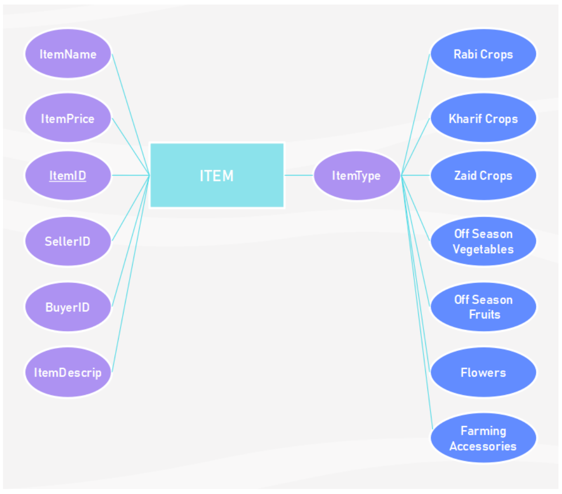


Fig.2. Item entity and its attributes with available buckets for *itemType* attribute

2.2 String Matchup

The system is now filled with multitudinous items which are organized in their respective buckets. Now, sellers have stored items in the system and it is the time when a buyer will come finding an item from the system. For this part, a string matchup algorithm shall be used in order to search the text matching the pattern which the buyer has prompted. Using *RegExp* class of Dart the implementation of string matchup could be easily done. Dart regular expressions have the same syntax and semantics as JavaScript regular expressions. The *firstMatch* method is the main implementation method that applies a regular expression to a string and returns the first *RegExpMatch*. All other methods in *RegExp* can be build from that. The *allMatches* to look for all matches of a regular expression in a string [4].

2.3 Real Time Distance based Sorting

After the phase of string matchup, now the system is having all the products which needs to be displayed to buyer. As the search was bucketed and matched with texts, the probability of anomalous and unrelated results becomes exiguous. Now, comes the part of implementing a distinctive feature of application's search space. Each buyer resides at some location that is it must have its coordinates (*latitude, longitude*) for his place. Similarly, the seller also has its shop set up at some place. Using these coordinates the distance between the seller and buyer could be known. This distance will be measured for each product which the buyer is going to see. After measuring, the products whose physical availability would have the least distance from the buyer will be appearing in as the first search result and in the same manner the product result set would be sorted.

To calculate the real time distance we need some APIs / Plugins, followed by some trigonometric formulae and algorithms.

2.3.1 Application Programming Interface

An application programming interface, or API, enables companies to open up their applications' data and functionality to external third-party developers, business partners, et al. This allows services and products to communicate with each other and leverage each other's data and functionality through a documented interface. Need not to know how an API is implemented, can simply use the interface to communicate with other products and services. API use has surged over the past decade, to the degree that many of the most popular web applications today would not be possible without APIs [5].

2.3.2 Packages

Packages enable the creation of modular code that can be shared easily [6].

2.3.2.1 Dart Packages

General packages written in Dart, for example the path package. Some of these might contain Flutter specific functionality and thus have a dependency on the Flutter framework, restricting their use to Flutter only, for example the *fluro* package [6].

2.3.2.2 Plugin Packages

A specialized Dart package that contains an API written in Dart code combined with one or more platform-specific implementations. Plugin packages can be written for Android (using Kotlin or Java), iOS (using Swift or Objective-C), web, macOS, Windows, or Linux, or any combination thereof [6].

2.3.3 Geolocation

To get the user's position on the globe, the geolocation API is used. It provides the user's latitude and longitude of the place where user stands in. The HTML Geolocation API is used to get the geographical position of a user. Since this can compromise privacy, the position is not available unless the user approves it [See Fig.3]. It is most accurate for devices with GPS, like smartphones [7]. For the system, the flutter geolocation plugin is used. It is a plugin which provides easy access to platform specific location services is a plugin known as *geolocator 8.2.0* [8].

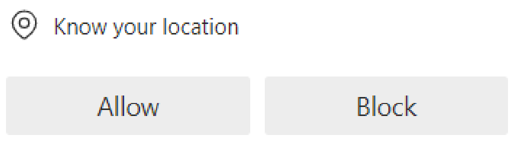


Fig.3. Permission request from user

2.3.4 Haversine Formula

As we know, Earth is a sphere and hence the radians and arcs becomes quite crucial in the calculation of the distance between two points on the earth. After getting the latitude and longitude, we can use the *law of haversines* to plot the distance between points. Given a unit sphere, a "triangle" on the surface of the sphere is defined by the great circles connecting three points *u*, *v*, and *w* on the sphere [See Fig.4]. If the lengths of these three sides are *a* (from *u* to *v*), *b* (from *u* to *w*), and *c* (from *v* to *w*), and the angle of the corner opposite *c* is *C*, then the law of haversines states –

$$hav(c) = hav(a-b) + \sin(a)\sin(b)\, hav(C)$$

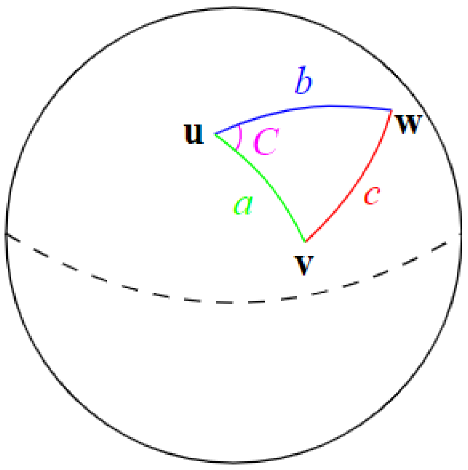


Fig.4. Spherical triangle solved by the law of haversines [9]

The haversine formula determines the great-circle distance between two points on a sphere given their longitudes and latitudes. Important in navigation, it is a special case of a more general formula in spherical trigonometry, the law of haversines, that relates the sides and angles of spherical triangles.
We have –

$$\theta = \frac{d}{r}$$

where *d* is the distance between the two points along a great circle of the sphere, *r* is the radius of the sphere. This formula allows the haversine of *θ* to be computed directly from the latitude (∅) and longitude (λ) of the two points –

$$hav(\theta) = hav(\emptyset_2 - \emptyset_1) + \cos\emptyset_1 \cos\emptyset_2\, hav(\lambda_2 - \lambda_1)$$

To avoid using cosines which cause resolution degradation at small angles –

$hav(\theta) = hav(\emptyset_2 - \emptyset_1) + \left(1 - hav(\emptyset_1 - \emptyset_2).hav(\emptyset_1 + \emptyset_2)\right).hav(\lambda_2 - \lambda_1)$ where $\emptyset_1$, $\emptyset_2$ are the latitude of point 1 and latitude of point 2 and λ_1, λ_2 are the longitude of point 1 and longitude of point 2. Finally, the haversine function hav(θ), applied above to both the central angle θ and the differences in latitude and longitude, is –

$$hav(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1-\cos(\theta)}{2}$$

The haversine function computes half a versine of the angle θ. To solve for the distance d, using the inverse haversine to $h = hav(\theta)$ or use the inverse sine function –

$$d = 2r \sin^{-1}\sqrt{h}$$

or we can also write it as –

$$d = 2r \sin^{-1}\left(\sqrt{\sin^2\left(\frac{\emptyset_2-\emptyset_1}{2}\right) + \cos\emptyset_1 . \cos\emptyset_2 . \sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right)$$

To have this formula implemented on earth, we take $r = 6371.529$ km which stands as an approximate between 6356.752 km at the poles to 6378.137 km at the equator in accordance to the fact that earth is not a perfect sphere [10]. Therefore, by using haversine formula we could get the physical distance knowing the latitudes and longitudes of the two points.

2.3.5 Using Haversinal Distance

The position of a seller is fixed and the position of buyers can be constant or could change variably. The application could prompt the seller to get the coordinates of his shop. For the buyer, it is possible that the buyer could be moving. For this, the application uses a time interval function which runs every single minute to update the location and sort the result set of items in accordance to different seller locations. For example, considering buyers as B_i having latitudes b_lat_i and longitudes b_lon_i where [$1 \leq i \leq$ Buyers in System] and sellers S_j having latitudes s_lat_j and longitudes s_lon_j [$1 \leq j \leq$ Sellers in System] for instance. After the calculation of haversinal distance for each B_i, the edges B_iD_k could be created which would carry the weight as the distance between buyer and item. At this time, when the search results will appear to buyer then they will be sorted in accordance to their item edge weight D_k where k represents the order of the appearance of items to buyers [See Fig.5]. The same mechanism would be utilized when a worker would search for a field job. He will get all the locations in his reach which were updated into the system by different agricultural landlords.

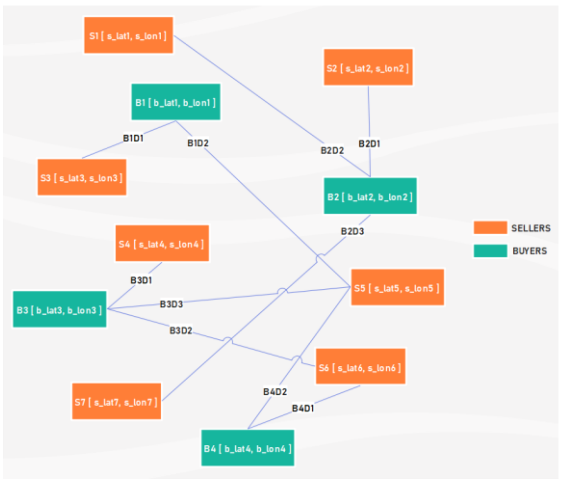


Fig.5. Buyers and Sellers connected with real time distance edge. The D_i part of edge represents the order of appearance of items to buyers.

III. Recommendation System

The explosive growth in the amount of available digital information and the number of visitors to a system have created a potential challenge of information overload which hinders timely access to items of interest on the digital platform. On the application, where the number of choices is overwhelming, there is need to filter, prioritize and efficiently deliver relevant information in order to alleviate the problem of information overload, which has created a potential problem to many users. Recommender systems solve this problem by searching through large volume of dynamically generated information to provide users with personalized content and services [11].

3.2 Collaborative Filtering

Recommender systems help users find information by recommending content that a user might not know about, but will hopefully like. Rating-based collaborative filtering recommender systems do this by finding patterns that are consistent across the ratings of other users. These patterns can be used on their own, or in conjunction with other forms of social information access to identify and recommend content that a user might like [12].

3.2.1 Memory based Collaborative Filtering

Memory-based methods use user rating historical data to compute the similarity between users or items. The idea behind these methods is to define a similarity measure between users or items, and find the most similar to recommend unseen items [13]. Describing some techniques to perform memory based collaborative filtering –

3.2.1.1 KNN

This algorithm needs two tasks **-**

- Find the K-nearest neighbors (KNN) to the user a, using a similarity function w to measure the distance between each pair of users

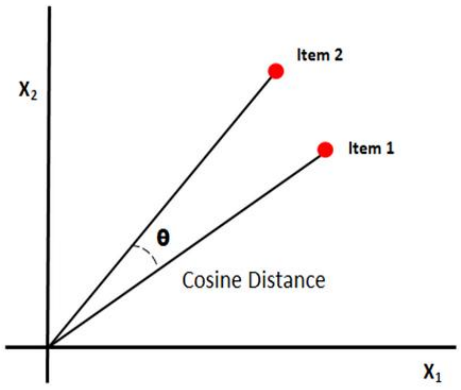
$$Similarity(a,i) = w(a,i), i \in K$$

- Predict the rating that user a will give to all items the k neighbors have consumed but a has not. Analyzing the item j with the best predicted rating.

This technique is also known as User – Based collaborative filtering [14].

3.2.1.2 Cosine Similarity

Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction [15].



$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Fig.6. Similarity using Cosine function [16]

3.2.1.3 Pearson Correlation

Pearson Correlation Coefficient (PCC) is one of the most popular similarity measures for Collaborative filtering recommender system, to evaluate how much two users are correlated [17]. The Pearson correlation coefficient gives us a number (least similar: -1; most similar: 1) that represents the similarity between two items [18].

$$Pearson\ Correlation : Sim(u_i, u_k) = \frac{\sum_j (r_{ij} - r_i)(r_{kj} - r_k)}{\sqrt{\sum_j (r_{ij} - r_i)^2 \sum_j (r_{kj} - r_k)^2}}$$

Fig.7. Pearson Correlation Coefficient Formula

3.1.2 The Cold Start problem

Collaborative Filtering has been widely used in many areas with different techniques. For this algorithm, whichever technique we use, one of the major issue which remains in the algorithm is –

If a new user arrives to the system, then what can be recommended to him? Each technique for giving recommendation needs previous data or mappings such that it could start predictions over an entity. But, what if the user has not rated any item, and has just became a part of the application system?

This problem is known as *Cold Start*. For every application using method of collaborative filtering, it is important that it should have a technique to address the problem of *Cold Start*.

3.1.3 Resolution to Cold Start

In the application, the methods and formulation which are being used for the optimization of the search space will be quite helpful for resolving the problem of *cold start*. For the collaborative filtering, we are extracting specific columns from our database –

- user_id
- item_id
- rating_given
- location
- timestamp

Here, two extra columns *location* and *timestamp* are being considered in alteration to traditional filtering analysis which works upon only *user_id, item_id* and *rating_given*. These two attributes can be utilized specifically in relevance to our problem.
Based upon these, presenting solutions to the *cold start* -

3.1.3.1 Solution 1:
Using Location

When the users are introduced to the system, the application stores a lot of information from the user. Similarly, we store the place of living of the users. For this, we take attributes –

- state
- city
- locality
- pincode

From these attributes, we create a derived attribute *location* which is defined as a composed string for every user where

$$location = state + \text{":"} + \text{city} + \text{":"} + \text{locality} + \text{": "} + pincode$$

Now, for every user this derived attribute will be created and stored. The thought is to use *location* matchup for every buyer this time. To deal with the *cold start* the filtering algorithm will use a specific function –
getItemsMappingFromLocation( )
The above function will use the string matchup to filter out the buyers who have the same location as of the particular user. Whenever the string matchup would become successful, it would fetch out what items have been bought or searched by this one. Now, the items which this user has given the highest rating will be appended into the list of *item_id* which would be returned by the function as the suggestion of items for the new user.
Now, in case for the given pincode, no user exists, then the search range should be increased. The similar thing follows for the *locality* perspective also. Hence, for this whenever the result set would not include any items then, the *location* string will be trimmed from last to the '**:**' character. Doing this, for the new user, the search range for items will be increased.

3.1.3.2 Solution 2:
Using Timestamp

The application is formulated in a manner that every time the user opens the app the stored information in the database gets updated. For the similar thing, we are storing the *timestamp* for the user and updating it whenever the application restarts. By using the *timestamp*, the date can be fetched. From the date, we can analyze that it is what month and hence we can analyze **which crop – season is going on**. The months for the sowing and harvesting of the crops are as follows –

- Kharif Crops –
  - Sown amid July and October
  - Harvested amid September and October
- Rabi Crops –
  - Sown amid October and November
  - Harvested amid February and April
- Zaid Crops –
  - Sown amid March and June
  - Harvested amid August and October

Since, we already have buckets initialized for our crops, we can just select the appropriate bucket and recommend items from it to the users in accordance to what crop – season is going on. The crops have their sale from the time they are harvested till their sowing time comes back. These recommendations can not only help the new buyers to the system, instead it can also be helpful to the sellers, such that they could know that what crops are at sale at the moment. The agricultural landlord and sellers can also know from the application that which crops are to be sown at the particular time.

IV. Conclusion And Future Scope

The categories on which we have proposed our research are *Search Space Optimization with Real Time Distance based sorting* and *Collaborative Filtering robust to Cold Start*, both having its significance in the domain of agricultural application. Because of these techniques the users can save their time to buy a product and the sellers can scale their profitability. For the recommendation part, an existing user can get product recommendation on the basis of his activity whereas every new user can get recommendations from either the item set which is mostly bought and highly rated in his surrounding or on the basis of the crop season going on. All these algorithms and workflow are presented for *toTheFarmers* application which is a mobile application and can be utilized by the farmers' community to leverage their living.

The future scope is to analyze and collect more data from the users so as to give better and more appropriate results to them. The itemized collaborative filtering can be enhanced with the model based collaborative filtering. The bucketing process can be more robust and the concept of clustering can be used for it. We are constantly working on acquiring more sturdy methods to build a better service to the community.

Acknowledgement

We extend our gratitude to the college faculties and professors for their support. Our sincere thanks to our guide **Professor Abhishek Malviya Sir** for his timely support and inputs.

References

[1] Gestal, Marcos, and Julián Dorado. "Genetic Algorithms in Multimodal Search Space." In Encyclopedia of Information Science and Technology, Second Edition. edited by Khosrow-Pour, D.B.A., Mehdi, 1621-1629. Hershey, PA: IGI Global, 2009. https://doi.org/10.4018/978-1-60566-026-4.ch256

[2] "What is a bucket or double-bucket data structure?", Internet – https://stackoverflow.com/questions/42399355/what-is-a-bucket-or-double-bucket-data-structure, [Apr 26, 2017]

[3] Y. S. Kuo, S. . -Y. Hwang and H. F. Hu, "A data structure for fast region searches," in IEEE Design & Test of Computers, vol. 6, no. 5, pp. 20-28, Oct. 1989, doi: 10.1109/54.43076

[4] "RegExp class", Internet – https://api.flutter.dev/flutter/dart-core/RegExp-class.html

[5] "Application Programming Interface (API)", Internet – https://www.ibm.com/cloud/learn/api, [19 Aug, 2020]

[6] "Package introduction", Internet – https://github.com/flutter/website/blob/main/src/development/packages-and-plugins/developing-packages.md

[7] "HTML Geolocation API", Internet – https://www.w3schools.com/html/html5_geolocation.asp

[8] "geolocator 8.2.0", Internet –https://pub.dev/packages/geolocator, [14 Feb, 2022]

[9] Image Source - "Spherical triangle solved by the law of haversines", Internet – https://en.wikipedia.org/wiki/Haversine_formula#/media/File:Law-of-haversines.svg, [Last Edit: 05 Mar, 2022]

[10] "Haversine formula", Internet – https://en.wikipedia.org/wiki/Haversine_formula#:~:text=The%20haversine%20formula%20determines%20the,and%20angles%20of%20spherical%20triangles, [Last Edit: 05 Mar, 2022]

[11] F.O. Isinkaye, Y.O. Folajimi, B.A. Ojokoh, "Recommendation systems: Principles, methods and evaluation", Egyptian Informatics Journal, Volume 16, Issue 3, 2015, Pages 261-273, ISSN 1110-8665, https://doi.org/10.1016/j.eij.2015.06.005, https://www.sciencedirect.com/science/article/pii/S1110866515000341)

[12] Kluver D., Ekstrand M.D., Konstan J.A. (2018) Rating-Based Collaborative Filtering: Algorithms and Evaluation. In: Brusilovsky P., He D. (eds) Social Information Access. Lecture Notes in Computer Science, vol 10100. Springer, Cham. https://doi.org/10.1007/978-3-319-90092-6_10

[13] "Types of Collaborative Filtering Methods", Internet - https://towardsdatascience.com/how-does-collaborative-filtering-work-da56ea94e331, [Oct 29, 2020]

[14] "Collaborative Filtering Using k-Nearest Neighbors (kNN)", Internet – https://towardsdatascience.com/how-did-we-build-book-recommender-systems-in-an-hour-part-2-k-nearest-neighbors-and-matrix-c04b3c2ef55c#:~:text=Collaborative%20Filtering%20Using%20k%2DNearest,of%20top%2Dk%20nearest%20neighbors., [Sep 20, 2017]

[15] "2.4.7 Cosine Similarity", Internet – https://www.sciencedirect.com/topics/computer-science/cosine-similarity#:~:text=Cosine%20similarity%20measures%20the%20similarity,document%20similarity%20in%20text%20analysis, [2012]

[16] Image Source – "MiWORD of the Day Is…Cosine Distance!", Internet - https://www.tyrrell4innovation.ca/miword-of-the-day-iscosine-distance/, [Jun 30, 2021]

[17] L. Sheugh and S. H. Alizadeh, "A note on pearson correlation coefficient as a metric of similarity in recommender system," *2015 AI & Robotics (IRANOPEN)*, 2015, pp. 1-6, doi: 10.1109/RIOS.2015.7270736.

[18] "Introduction to Data Analysis: Movie Recommendations", Internet - https://betterprogramming.pub/building-a-movie-recommendation-engine-for-beginners-7161430e35b9, [Jun 15, 2020]