INDIRA GANDHI DELHI TECHNICAL UNIVERSITY FOR WOMEN

IT DOCUMENTATION
FOOD RECOMMENDATION SYSTEM

TEAM 53
103_ANANYA N CHOUHAN [CSE-AI 2]
140_SHRUTI CHHIKARA [CSE-AI 2]
141_DIVYA LAMBA [CSE-AI 2]

UNDER GUIDANCE OF:-
Ms. DIMPLE
Ms. GARIMA -

CONTENT

PROBLEM STATEMENT

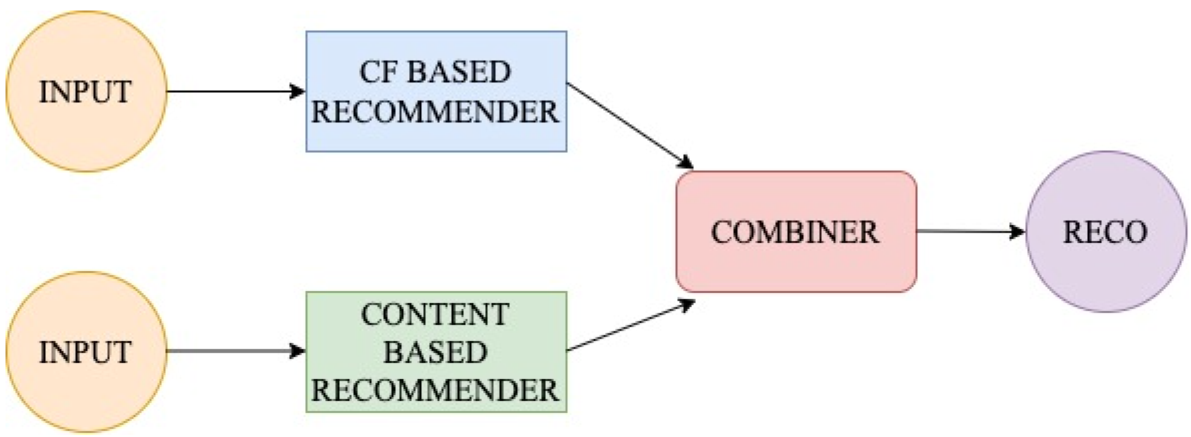
Consumer buying behavior is impacted by article suggestions. Recommendations can be provided in the form of ratings or rankings for specific products. This review proposes a recommendation system that has been trained based on the recommendations of customers who have formerly used the product. The software recommends the product to the client based on the consumer's experience with the same product. Each person has his or her own diet based on the likes and dislikes of the user, demonstrating the significance of individual nutrition to maintain the success and health of the user. The proposed recommendation system uses a deep learning algorithm and a genetic algorithm to provide the best possible advice.

INTRODUCTION

A recommendation system is a subclass of an information filtering system. An information system's main goal is to manage the overload of information. To do this, a user profile is referenced to some characteristics or features. These features may be :-

Content based- based on information of item

Collaborative filtering-User's social environment



Consumer buying behavior is impacted by article suggestions. Recommendations can be provided in the form of ratings or rankings for specific products. This review proposes a recommendation system that has been trained based on the recommendations of customers who have formerly used the product. The software recommends the product to the client based on the consumer's experience with the same product. Each person has his or her own diet based on the likes and dislikes of the user, demonstrating the significance of individual nutrition to maintain the success and health of the user. The proposed recommendation system uses a deep learning algorithm and a genetic algorithm to provide the best possible advice.

There are different types of recommendation systems in the market:-

1. Content-based- these are based on the featuralization of items. This is usually used when known data on items is given, how users interact with the system but user's personal information isn't given.

2. Collaborative filtering-this model is best suited with known data on users i.e. personal information is given but a lack of data for items or it is difficult to do feature extraction for items of interest.

3. Hybrid system- because of the limitation of both content and collaborative models, hybrid models are used to meld both the advantages of the system. It is essentially a model ensemble approach. Tree model, linear regression neural network models etc. can be implemented here.

MOTIVATION

A machine learning algorithm known as a recommender system basically predicts future behavior based on past preferences of a user. Looking back at the history of e-commerce, due to a fixed inventory because of small physical space, retailers were encouraged to stock up on the most well-liked mainstream goods. Online marketplace has revolutionized the retail industry. By narrowing the search space and selecting the most pertinent, high-quality content for consumers, recommender systems reduce choice overload. User feedback fuels recommender systems. As we learn more about a user's preferences, we can create recommendations that are more pertinent and suited to their preferences.
Our personal motivation to build this certain kind of model was if I own a food Delivery App or a Restaurant Web Application and want my customers i.e. Average Cart Value to be more so that whenever he/she is ordering an item or dish he/she can add more, so that the sales are more thus making the overall revenue for the company higher.

LITERATURE SURVEY

Objective	Parameter	Dataset	Result	Calculation/ Future Scope
Their recommendations are based on the personality types of the user and users/ online recipe reviews with the same personality type. They tested if there was a connection between personality types and recipe categories Author- Adaji et al. Year of Research-2018	Recipes, Personality types of user	Approach: CB,CF Dataset- This study was carried out using data from the popular recipe site allrecipes.com	Type of the study: Experiment 1. concluded that the category of recipes, meat based or vegan, had no influence on the personalities of the reviewers. 2. here is no significant difference between the reviewers who are of personality types *extraversion* and *agreeableness*	Personalized recommendation system that uses the personality type of reviewers in making recipe-based recommendations. (Because people of the same personality tend to have many similarities)
RS that predicts user's preference of restaurants	Restaurants, visual information	Approach: Hybrid-CB+CF	Type of study: Experiment	The results with

by combining metadata, textual and visual information (photos) in blogs. They tested the performance of their RS Author: Chu & Tsai Year of Research-2017		Restaurant metadata, photos from blogs,	1.shown micro-blog topic recommends to the users up to some extent but not full accurate	existing model and which shows 3% - 5% improvements in restaurant and recipe's recommendations
The RS recommends recipes based on user information and contextual factors e.g. the food items available in the kitchen, preparation time of the user, and weather Author:Pratibha & Kaur Year of Research-2019	Recipe	Approach: Context-based	Type of study: Explain model	It implies it is utilized to foresee the question, as per the client's intrigue and if it has comparable content in another protest that is utilized to fulfill the client.
IntelliMeal recommends recipes based on the ingredients the user (dis)likes. The RS continuously learns the user's preferences as the user can give feedback. They evaluated the performance of the RS Author: Skjold, Øynes, Bach & Aamodt Year of Research-2017	Recipe	Approach: Knowledge Based: Case -based Reasoning	Type of study: Experiment	Adaptation of cases increased similarity scores for a given user query in all test cases, and humans had difficulties distinguish computer and human created recipes from one another
Use shopping receipts to keeps track of the nutritional content of foods they have eaten and suggest healthier food alternatives that the user can buy next time Author: Mankoff, Hsieh, Hung, Lee & Nitao Year of Research-2002	Groceries	Approach: Knowledge based	Type of study: Explain model	A set of alternatives is delivered to the user. These are usually presented as a ranked list according to the item utility for the user
QuickyCook recommends recipes that contain (almost) all ingredients that the user	Recipes	Approach: Knowledge based	Type:Experiment. They measured	More control to the user can be provided by letting users add

wants to include. Author: Bushra and Hasan Year of Research: 2019		Recently developed database called Recipe 1M.	the performance of the RS.The processing time compared to the size of the database is quite impressive after the initial run	filters based on meal (appetizer/dessert/entre) type, cuisine type, preparation time, etc.
They compared the performance of their multi-view RS with that of single-view RSs. Author: Zhang,Luo, Chen & Guo Year of Research: 2020	Restaurants	Approach: Hybrid: Collaborative filtering and content based Dataset: multiple images (drink, food, inside & outside of the restaurant). Epinions dataset	Type: Experiment CF-based filtering method can find the user's potential interests but not yet discovered preferences	With the development of deep learning technology, deep learning has gradually been applied in RS due to strong feature representation, and it can learn the latent item association from the user-item rating directly for predictive recommendation without employing a similarity measure
Recommends recipes based on the user's mood (6 aspects: body, mental, taste, time, price, and modification) Author: Ueda, Morishita, Nakamura, Takata & Nakajima Year of research: 2016	Recipes	Approach: Context-based Privately collected data	Type: Explain Model	Students rated recipes on each mood aspect.
The RS bases the restaurant dish recommendations on the user dish ratings, user information, contextual factors, and food attributes. Author: Li et al	Recipes	Approach: Context-based Local Chinese dataset	Type: Experiment The experimental results show that the slope one algorithm based on the	Propose a slope one algorithm based on the fusion of trusted data and user similarity. The algorithm we proposed can applied in many applications, such

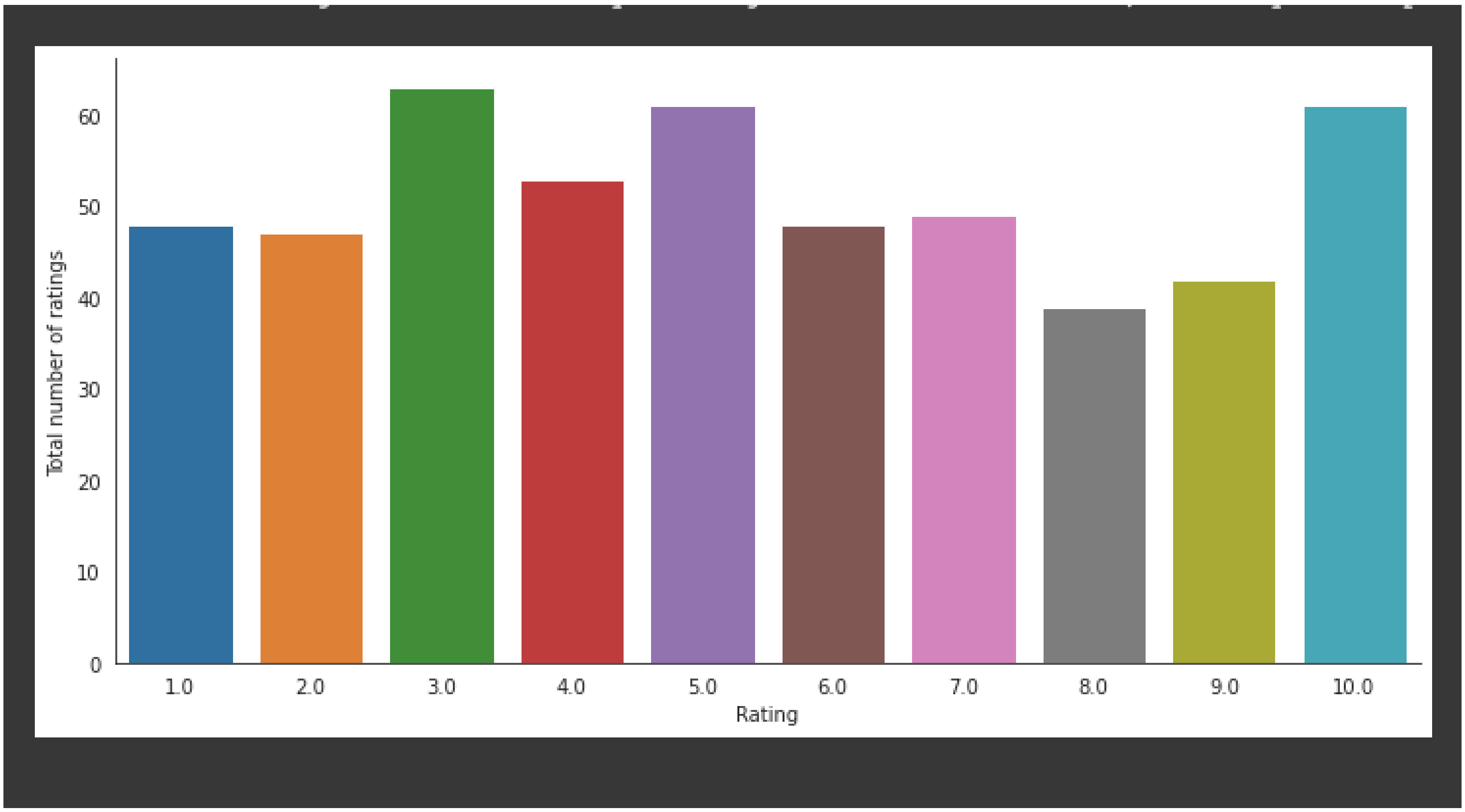
Year of Research:2018			fusion of trusted data and user similarity has greatly improved the prediction accuracy than traditional slope one algorithm.	as the recommendation system for social networks
The RS ranks culinary destinations for tourists based on user positive and negative criteria. Author: FARIED EFFENDY, BARRY NUQOBA, TAUFIK Year of research:2019	Restaurants	Approach:Content based Survey and literature studies are conducted to find out what criteria are considered by tourists when choosing culinary destinations	The decision support system with the Fuzzy TOPSIS method helps users by identifying the essential criteria in the process of selecting culinary destinations.	Decision support systems that are built have a high impact on solving the problem of choosing culinary destinations by providing recommendations on culinary destinations that suit the needs and preferences of users. This feedback can be related to culinary destinations so that this application can be an effective and unbiased platform for people who need culinary destinations recommendation.

METHODOLOGY

Dataset:-
We have used two different datasets - Food.csv and Ratings.csv.
Food.csv has information regarding the recipe- such as its nutrition content (Healthy or unhealthy), ingredients-vegetarian or non-vegetarian.
Ratings dataset has the ratings of every food item. Based on how much every user likes the food item, or how popular it is.
Each food-item has a unique FoodID.

	Food_ID	Name	C_Type	Veg_Non	Describe
0	1	summer squash salad	Healthy Food	veg	white balsamic vinegar, lemon juice, lemon rin...
1	2	chicken minced salad	Healthy Food	non-veg	olive oil, chicken mince, garlic (minced), oni...
2	3	sweet chilli almonds	Snack	veg	almonds whole, egg white, curry leaves, salt, ...
3	4	tricolour salad	Healthy Food	veg	vinegar, honey/sugar, soy sauce, salt, garlic ...
4	5	christmas cake	Dessert	veg	christmas dry fruits (pre-soaked), orange zest...

	User_ID	Food_ID	Rating
0	1.0	88.0	4.0
1	1.0	46.0	3.0
2	1.0	24.0	5.0
3	1.0	25.0	4.0
4	2.0	49.0	1.0



Website:-

Our website is used for implicit feedback. Users enter their preferences in our questionnaire and on the basis of their feedback, our model recommends them several food items.It is implemented using python library Streamlit(open source app-framework for deploying machine learning models) and hosted using Ngrok.

Model:-

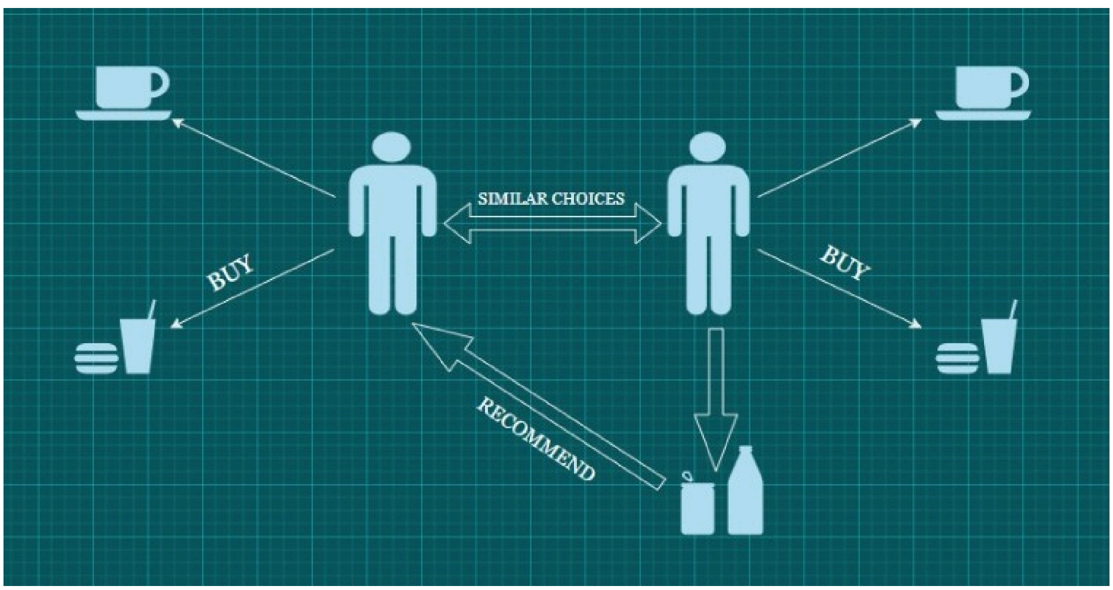
We have used collaborative filtering to make our model. However to combat the cold-start problem, we have also employed content-based filtering.

Collaborative filtering is using the concept of ‘homophily’-similar people like similar things. It is using a user-item matrix- which is populated with how much a user prefers a particular item (food item here).

```
[ ] #main recommendation function
    def food_recommendation(Food_Name):
        n = 10
        FoodList = food[food['Name'].str.contains(Food_Name)]
        if len(FoodList):
            Foodi= FoodList.iloc[0]['Food_ID']
            Foodi = dataset[dataset['Food_ID'] == Foodi].index[0]
            distances , indices = model.kneighbors(csr_dataset[Foodi],n_neighbors=n+1)
            Food_indices = sorted(list(zip(indices.squeeze().tolist(),distances.squeeze().tolist())),key=lambda x: x[1])[:0:
            Recommendations = []
            for val in Food_indices:
                Foodi = dataset.iloc[val[0]]['Food_ID']
                i = food[food['Food_ID'] == Foodi].index
                Recommendations.append({'Name':food.iloc[i]['Name'].values[0],'Distance':val[1]})

            import pandas as pd
            df = pd.DataFrame(Recommendations,index=range(1,n+1))
            return df['Name']
        else:
            return "No Similar Foods."
```

We use a combination of both- explicit and implicit feedback. Explicit feedback is used as the items are recommended based on ratings. Implicit feedback is used as the user also has to give preferences in our web-based questionnaire.



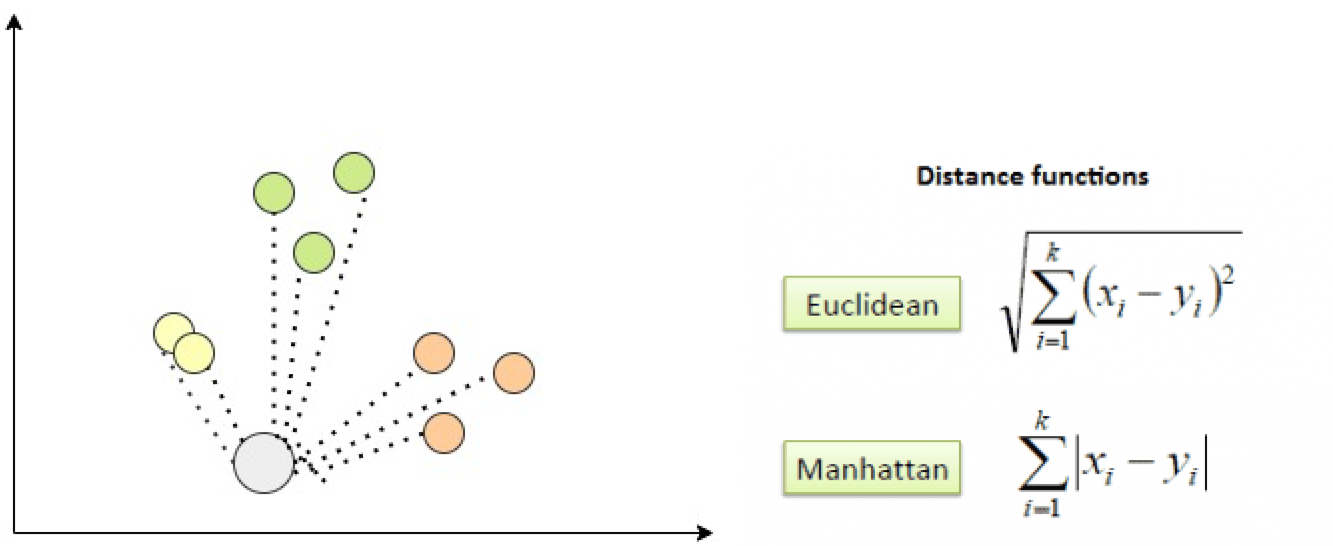
Cold-start problem- a new user, on our website who hasn't interacted with any of the food items, might get random recommendations. To solve this problem, content-based filtering is used- which takes into account the new user's questionnaire and matches it with existing users. To measure similarity, we first need to represent food-items in a n-dimensional space where *n* represents the number of food features. We can then measure the distance between different food items in this n-dimensional space using similarity metrics such as Euclidean distance, cosine similarity, Pearson correlation, and Jaccard distance. The smaller the distance between two movies, the more similar they are to one another.

Here we have used cosine similarity and employed KNN: K-nearest neighbors algorithm.
Distance between the different color data points is a good way to estimate similarity. There are some ways to calculate this distance ; Euclidean, Manhattan, etc. Another way to calculate is to use the Cosine Angular distance. To calculate similarity between angles we need a function that returns a higher similarity or smaller distance for a lower angle and a lower similarity for a higher angle.

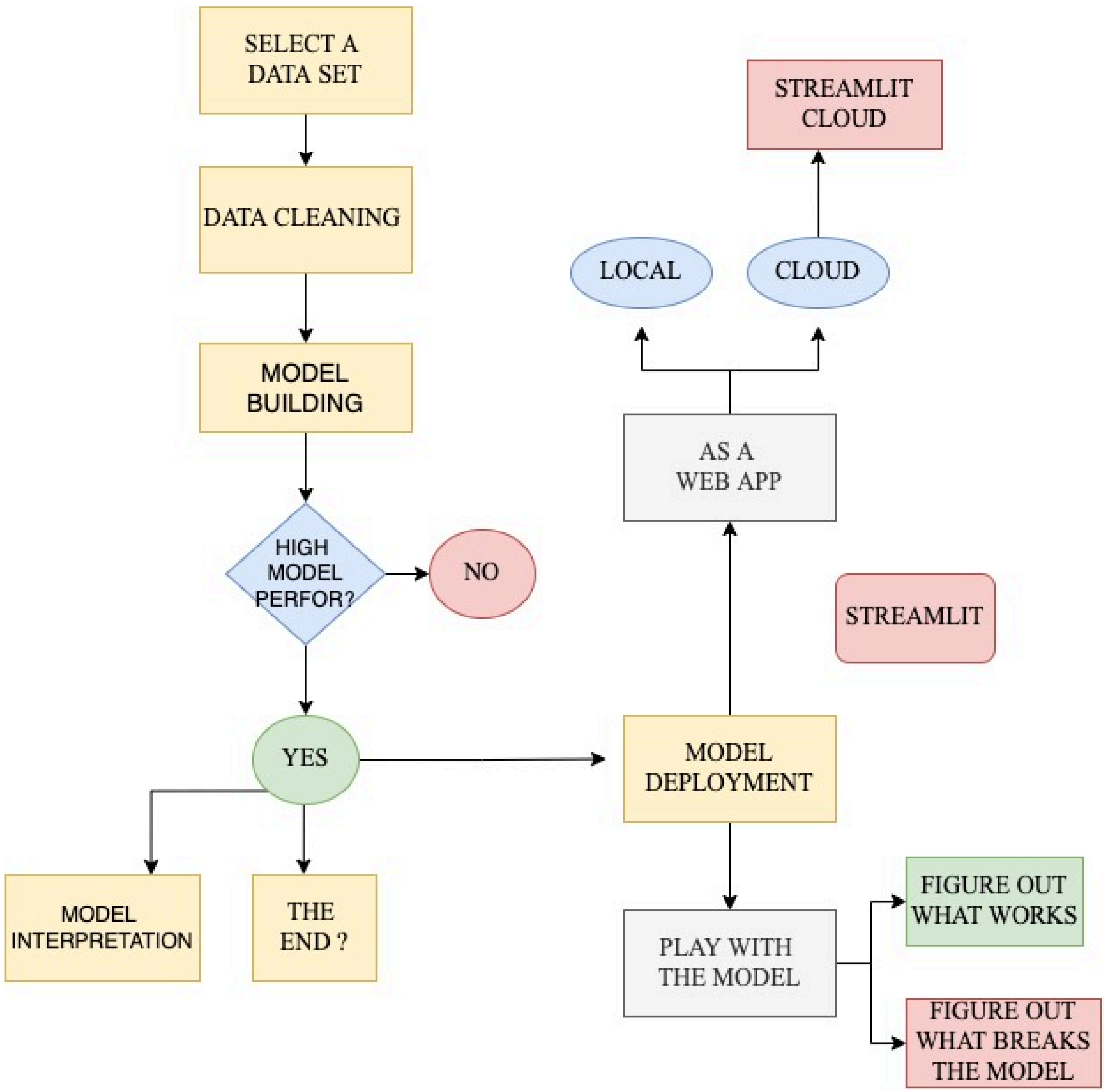
Scipy has a function that calculates the cosine distance of the vectors .
We have used the centered Cosine, where there are a lot of missing values in vectors and we need to place a common value to fill up the missing values.

The distance between u and v, using cosine formula is:

$$1 - \frac{u \cdot v}{\|u\|_2 \|v\|_2}.$$



Libraries Used: (for Recommender Model):-
Pandas, Numpy
Matplotlib: It is used for plotting graphs and data visualization in python. It offers a viable open source alternative to MATLAB.
skLearn: Scikit-learn (Sklearn) is the most useful and robust library for **machine learning in Python**. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python.
Scipy.sparse
Sklearn.neighbours
Libraries Used: (for Deploying the model):-
Streamlit: Streamlit is an open source app framework in Python language. It **is used to create web apps for data science and machine learning in a short time**. It is compatible with major Python libraries such as scikit-learn, Keras, PyTorch, SymPy(latex), NumPy, pandas, Matplotlib etc.
Other services used: -
Ngrok-**utility to expose any locally hosted application over the web. (provides a publicly accessible web URL to any locally hosted application)**



RESULT ANALYSIS

We implemented both SVD and KNN models. Both are extremely popular models for collaborative filtering.
We calculated the Root Mean Squared Error (RMSE) to test the accuracy of both models. RMSE is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed. RMSE values for both the models are: KNN (0.09769) and SVD (0.1039). Based on this, we observed that KNN model proves to be a better fit for our dataset.

```
food_recommendation('dates and nuts ladoo')

1          lamb and chargrilled bell pepper soup
2                                       red rice
3                                    egg paratha
4                           dahi lasooni chicken
5                         andhra crab meat masala
6                white chocolate and lemon pastry
7                                 chicken dragon
8                          lemon poppy seed cake
9                          steam bunny chicken bao
10                             cashew nut cookies
Name: Name, dtype: object
```

```
food_recommendation('andhra crab meat masala')

1        red wine braised mushroom flatbread
2                      eggless coffee cupcakes
3                              sugar free modak
4          watermelon and strawberry smoothie
5                      baked raw banana samosa
6                      steam bunny chicken bao
7                           cashew nut cookies
8                         dates and nuts ladoo
9                               detox haldi tea
10                              tricolour salad
Name: Name, dtype: object
```

Root Mean Square Error of the model:-

```
[77] rmse_df = pd.concat([train_data.mean(), test_data.mean()], axis=1)
     rmse_df.columns = ['Avg_actual_ratings', 'Avg_predicted_ratings']
     print(rmse_df.shape)
     rmse_df['item_index'] = np.arange(0, rmse_df.shape[0], 1)
     rmse_df.head()
```

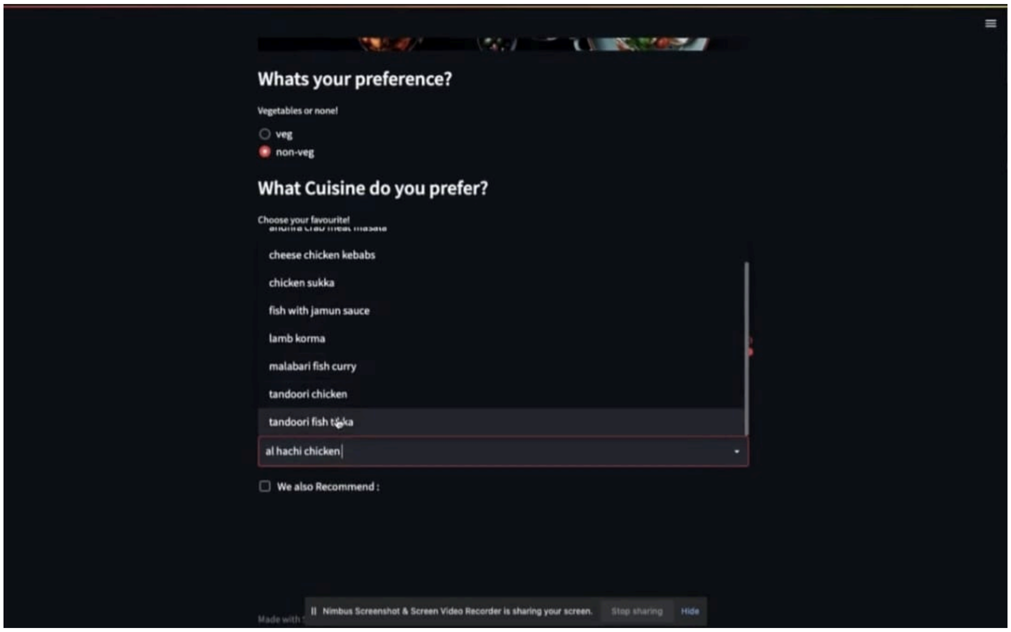
```
(100, 2)
```

User_ID	Avg_actual_ratings	Avg_predicted_ratings	item_index
1.0	0.060185	0.032258	0
2.0	0.120370	0.000000	1
3.0	0.166667	0.129032	2
4.0	0.125000	0.010753	3
5.0	0.129630	0.150538	4

```
RMSE = round((((rmse_df.Avg_actual_ratings - rmse_df.Avg_predicted_ratings) ** 2).mean() ** 0.5), 5)
print('\nRMSE KNN Model = {} \n'.format(RMSE))
```

```
RMSE KNN Model = 0.09769
```

We have deployed the model on local host using Streamlit Library provided by Python. We have used ngrox to create a secure tunnel to locally host our web application using a Reverse Proxy.



CONCLUSION

Overall, through our project we have presented a food recommendation approach focused on generating daily personalized meal plans for the users, according to their nutritional necessities and food preferences. The most recent related works prove that although there are several researches focused on developing computational tools for food intake advice, most of them do not directly manage both nutritional information and user preferences.

However, the final web application created in this project presents a general architecture for food recommendation, composed of an information gathering layer which asks the user about their food-type preferences and the health and quality factor measured on the basis of the rating of the food item. After that, the web application recommends a dish which satisfies the needs of the user along with some extra recommendations with the same rating as the recommended item.

FUTURE SCOPE

Our future research will be focused on three main directions:

- The use of long-term information for menu generation: Our project only considers physical information for the nutritional requirement calculation. In this future direction, the goal will also use the previous food logs as input for the nutritional calculation, in order to guarantee an adequate weekly-monthly food intake balance.
- The incorporation of recipe recommendations into the daily generated meal plan: Recipe recommendation has recently been studied by some authors, and therefore it is necessary to integrate it into our approach which is focused on the simultaneous management of preference-based and nutritional information.
- The exploration of the presented approach in a group recommendation scenario. Group recommendation(which has a direct application in food recommendation) has recently been a very active

research area. Therefore, it is necessary to extend our current approach to be used in the group recommendation context.

We will also focus on the recommendations breakdown for different timings of the day, such as breakfast, lunch, and dinner. Moreover, we will consider the amount of nutrition in different food items as per timing and daily needs of the users.

In this day and age, we have so many choices when it comes to making online purchases, watching a TV show or movie using a video streaming service, or finding songs to listen to online. Currently we are using matrix factorization and collaborative filtering to build recommendation systems. But the future is in Deep Learning and Neural Networks. Recommender systems can be a very powerful tool in a company's arsenal, and future developments will add even more value to the business.

We can even get our model patented and sell it to the companies which are working on similar applications so that they can use it for their further growth as well.

REFERENCES

[1]https://medium.com/fnplus/everything-you-need-to-know-about-recommendation-systems-348cdfccbaf0: Everything You Need to Know About Recommendation Systems — Part 1

[2]https://realpython.com/build-recommendation-engine-collaborative-filtering/#:~:text=Collaborative%20filtering%20is%20a%20family,type%20of%20collaborative%20filtering%20approach.: Build a recommendation system-realPython.com

[3]RUI CHEN 1,2, QINGYI HUA1 , YAN-SHUO CHANG3 , BO WANG1,4, LEI ZHANG5 , AND XIANGJIE KONG 6 , (Senior Member, IEEE)-A Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based on Social Networks

[4]QuicklyCook: A User-Friendly Recipe Recommender: Naila Bushra and Mehedi Hasa

[5]Adomavicius, G., Bockstedt, J., Curley, S., & Zhang, J. (2011). *Recommender systems, consumer preferences, and anchoring effects*, *811*, 35–42.

[6]https://www.researchgate.net/publication/334628838_A_SURVEY_ON_COMPREHENSIVE_TRENDS_IN_RECOMMENDATION_SYSTEMS_APPLICATIONS

[7]Asanov, D. (2011). *Algorithms and methods in recommender systems Berlin Institute of Technology*. Germany: Berlin.

[8]Bokde, D.K., Girase, S., & Mukhopadhyay, D. (2015). Role of matrix factorization model in collaborative filtering algorithm: A survey. CoRR abs/1503.07475.

[9]Burke, R., Felfernig, A., & Göker, M.H. (2011). Recommender systems: an overview. *AI Magazine*, *32*, 13–18.

[10]Elahi, M., Ge, M., Ricci, F., Massimo, D., & Berkovsky, S. (2014). Interactive food recommendation for groups, *RECSYS*, Vol. 1247.

[11]J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, Recommender systems survey. Knowledge-Based Systems, 46, p109-132,2013

.http://dx.doi.org/10.1016/j.knosys.2013.03.012.

[12]Wikipedia, Location-based recommendation. Retrieved on January 28, 2018, from https://en.wikipedia.org/wiki/Location-based_recommendation#Background

[13]J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, Recommender system application developments: A survey. Decision Support Systems, 74, 12-32, 2015. http://dx.doi.org/10.1016/j.dss.2015.03.008.

[14]G. Adomavicius, and A. Tuzhilin, Context-aware recommender systems. In F. Ricci, L. Rokach, B. Shapira (eds.). Recommender Systems Handbook (p191-226). Springer, Boston, MA, 2015. http://dx.doi.org/10.1007/978-1-4899-7637-6_6.

[15]M. Sipper, R. S. Olson, and J. H. Moore, Evolutionary computation: The next major transition of artificial intelligence? Retrieved on August 30, 2018, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5534026/

[16]P. Cremonesi, A. Tripodi, and R. Turrin, Cross-domain recommender systems. IEEE 11th International Conference on Data Mining Workshops, p496–503, 2011. http://dx.doi.org/10.1109/ICDMW.2011.57.

[17]Felfernig, A., Zehentner, C., Ninaus, G., Grabner, H., Maalej, W., Pagano, D., Weninger, L., & Reinfrank, F. (2012). Advances in user modeling, Springer Berlin Heidelberg, chapter: Group Decision Support for Requirements Negotiation, 105–116.

[18]Freyne, J., Berkovsky, S., & Smith, G. (2011). Recipe recommendation: Accuracy and reasoning, *19th international conference, UMAP 2011, Girona, Spain, July 11-15, 2011., Springer Berlin Heidelberg* (pp. 99–110).

[19]Jameson, A. (2004). More than the sum of its members: Challenges for group recommender systems, *Proceedings of the working conference on advanced visual interfaces, ACM, New York, NY, USA, AVI '04* (pp. 48–54).

[20]Masthoff, J. (2011). Group recommender systems: Combining individual models, *Recommender systems handbook, Springer* (pp. 677–702).

[21]J. Sobecki, E. Babiak, and M. Słanina, "Application of Hybrid Recommendation in Web-Based Cooking Assistant," Springer, Berlin, Heidelberg, 2006, pp. 797–804

[22]B. Lika, K. Kolomvatsos, and S. Hadji Efthymiades, "Facing the cold start problem in recommender systems," Expert Systems with Applications, vol. 41, no. 4, pp. 2065–2073, Mar. 2014

[23]N. B. Silva, I.-R. Tsang, G. D. C. Cavalcanti, and I.-J. Tsang, "A graph-based friend recommendation system using Genetic Algorithm," in IEEE Congress on Evolutionary Computation, 2010, pp. 1–7.