

Understanding user Taste Preferences for Food Recommendation

A. Naresh

Dept. of Computer Science and Engineering,
Jawaharlal Nehru Technological University College of
Engineering
Pulivendula, Andhra Pradesh, India

M.S.S Shaastry

Dept. of Computer Science and Engineering,
Jawaharlal Nehru Technological University College of
Engineering
Pulivendula, Andhra Pradesh, India

B. Purushottam Yadav

Dept. of Computer Science and Engineering,
Jawaharlal Nehru Technological University College of
Engineering
Pulivendula, Andhra Pradesh, India

K Bhaskar

Dept. of Computer Science and Engineering,
Jawaharlal Nehru Technological University College of
Engineering
Pulivendula, Andhra Pradesh, India

Abstract:- Machine learning has led to many technological innovations in several industries like marketing, eCommerce, etc...The benefits of AI and ML systems are experienced by many people without even knowing it, in search and recommendation systems as in google search and the way Netflix recommends movies. Furthermore, ML models can hypothetically be trained to understand the taste and predict taste preferences accurately under specific conditions. This ML model is to understand the taste and recommend food products based on users' tastes. The same process by which a machine learns to identify a users' taste in music applies to a user's taste in food. Based on the previous food orders made by the user we make our ML model understand his taste preference and recommend the food products accordingly. Our study provides a better insight into how an ML model understands the user's taste preferences. In this paper, we provide functionalities to keep track of his previous food orders and taste preferences. Also, we recommend food items for individuals considering their taste preferences.

Keywords— Food Recommendation system, Chatbot, Users' Taste Preference, customised A* Algorithm, customised BFS Algorithm, Manhattan distance, Manhattan cutoff.

I. INTRODUCTION

Food is primitive for human life. It is also an essential part of human life. In the sophisticated world, people have a huge choice concerning the food they eat. The luxury of an abundance of choice offers not only many advantages, but also decision making of what to eat is always complex to the point of being overwhelming. There is an increasing attention for the food recommender systems in recent days due to their relevance to food recommendations based on considering their

taste preferences. Our recommendation studies on the food domain not only recommend food suiting user preferences but also keep track of eating routine. While many food recommender systems focus on recommendations that suggest

food items to a group of members. In such scenarios, the preferences of all group members have to take into account. Also, group recommendation functionalities create many challenges for food recommender systems. This paper discusses all such scenarios and research challenges which are blockers to the development of future food recommendation systems. However, the food domain is a complex domain bringing many challenges for recommendation systems. It is often difficult for consumers to make choices from huge varieties of food items.

For a machine learning model to predict accurately a user's preference or taste, for food is likely to appeal to people, the machine learning model would need some quantifiable data related to food items [1]. Since this data would need to be gathered in the physical world, it will be quite difficult, but not impossible.

Quantifiable factors related to food include:

- Chemical compounds found in food items
- Relative amount(parts per milliliter) of various compounds in the food items
- The color of food
- The type of food

Keeping in the view above all mentioned factors, it is important to consider all the factors into account to suggest food according to the user's taste. In every food item, there will be a combination of different taste may present in it. Some of the tastes we considered here are sweet, salty, sour, umami, astringency and bitter.

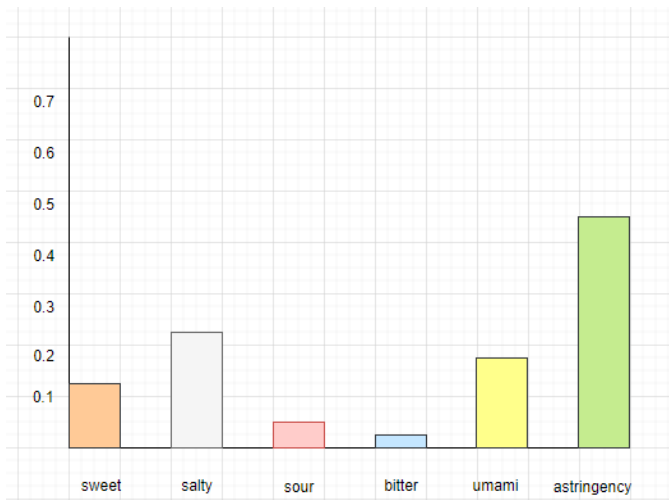


Fig.1. Example of a Food item and different Taste values of it.

II. RELATED WORK

A. USER PREFERENCE AS PREDICTOR

Methods for predicting user’s taste of food items considering user preference using machine learning approaches creates many challenges for food recommender systems since there are some situations where there is a group of members that may sit together for dinner, then preferences of all group members have to be considered appropriately. [2] Besides, the group recommendation scenario, an intelligent approach is proposed to suggest food items to the individual users based on that particular user’s rating on the previous orders. Food ordering chatbot has been developed to collect user preferences for a particular food item. The following table shows how the food item is given a certain value for each of the different tastes according to the food items he ordered earlier.

TABLE I

Taste Values for a particular food item

Item	sweet	salty	bitter	sour	umami	astringency
1	0.121013	0.209636	0.011631	0.013928	0.188598	0.45519304
2	0.103463	0.286349	0.164474	0.098336	0.220877	0.12650166
3	0.242817	0.146773	0.214216	0.097776	0.262706	0.03571207
4	0.290134	0.105297	0.334823	0.138211	0.089056	0.04247905
5	0.229776	0.122944	0.206066	0.189916	0.196159	0.05513877
6	0.305392	0.155347	0.217174	0.116792	0.010316	0.19497922
7	0.046758	0.021164	0.114614	0.240566	0.301171	0.2757258
8	0.223673	0.254988	0.100481	0.059651	0.222374	0.13883349
9	0.193088	0.02481	0.238294	0.275916	0.094305	0.17358736
10	0.06432	0.359743	0.040815	0.331648	0.196371	0.00710203

B. Data Collection

To collect data we have developed a simple chatbot that collects the user taste preferences from his previously ordered food items. We have made a data set with 920 food items with a variety of tastes. Therefore, the user has an abundant choice of food items. The scale would quantify the user judgments on his taste preference based on the taste values as mentioned in the above table. The user is asked to log in to the chatbot and order the food through the chatbot we have developed. We have collected data for the same for the past 11 months with 15 such users. This study provides users with a comprehensive human-computer synergy to collect long-term user preferences in terms of recipe taste values. All the food items are gathered and the taste values of every food item are predicted accordingly. Therefore, predicted food items are recommended to the user based on the taste values of that particular food item.

C. Generating a rank from a correlation Value

Newly, some proposals have been made in food recommender systems, such as *using labels for different clusters of users*. In this model, a rank is generated from a correlation value. After that, the recommender system will estimate the correlation between the user taste preferences and the previously estimated taste value from the food items. If the correlation value is higher than a predefined threshold, the food item is recommended, if not, it gets ignored [2].

D. Nutritional needs of user

In this context, some food recommendation systems have included some personal information of user *age, gender, occupation, physical activities, health problem*, etc... are collected. Based on this information food items that are the best fit for the user’s nutritional needs are recommended. The fundamental steps required to generate the data to be used in recommendation systems are based on :

i. Nutritional information of food profiles.

To make this data fit for recommendation system nutritionist defined appropriate portions for all food according to its kind and characteristics and determines the amount of macro and micronutrients pertaining to each portion.

$$T_k = (Pro_k, Lip_k, Cb_k, Ch_k, Sod_k, Sat_k)$$

- Pro_k = Amount of proteins of food k
- Lip_k = Amount of lipids of food k
- Cb_k = Amount of carbohydrates of food k
- Ch_k = Amount of cholesterol of food k
- Sod_k = Amount of sodium of food k
- Sod_k = Amount of saturated fats of food k
- T_k = Total Amount of macro and micro nutrients pertaining to food k

Carbohydrates, proteins, saturated fats, cholesterol, sodium, lipids are considered as important criteria while recommending the food items to the user[4].

ii. Kind of menu cards to be filled by the food items.

A menu card follows the general scheme of a regular daily meal and it is also created through the assistance of a nutrition specialty expert. This menu card is made of breakfast, lunch, and dinner. Besides, Considering the goal of this scheme is to implement a personalized food recommendation system for users. [4] This scheme is formulated to identify several nutritional-aware user types.

Further, this recommendation model can be extended to include this additional feature as mentioned in the section II subsection D.

Considering above all, each food item has given some value for each of six different tastes, as shown in the TABLE I.

III. OVERVIEW

User Food Recommendation System is nowadays becoming very popular because of abundant choice food items for the customer and this made the users' choice of choosing a food item complex, among such a vast variety of food items. This Food Recommendation System has predicted food items according to user taste preferences using taste values of every food item as shown in the TABLE I. [3] Based on that taste values of sweet, salty, sour, umami, astringency and bitter an item is recognised which is a 6 dimensional vector having value for each taste which sum up to one are created for each taste.

tastes are:

- SWEET
- SALTY
- SOUR
- UMAMI
- ASTRINGENCY

• BITTER

A. Creation of Graph.

From a menu of food items and six taste values of each food item in that menu. Six-dimensional graphs are drawn in such a way that each food item is represented as a node in the six-dimensional space, where each dimension represents each taste. All such graphs are drawn based on the Manhattan cutoff calculated from the mean and average of taste values of all food items.

$$\text{Manhattan cutoff}(M_x) = \min(\text{Mean}(A_k), \text{Average}(A_k))$$

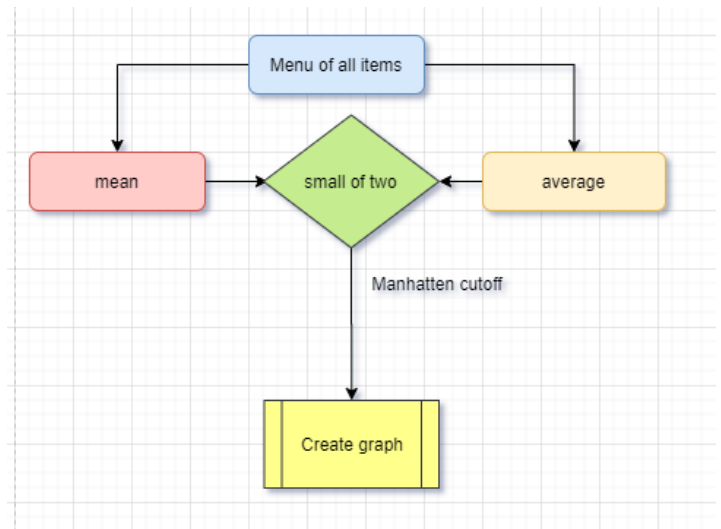


Fig .II Creation of graph from Manhattan Cutoff

B. Calculation of Manhattan distance (M_d)

Every food item is located in a six-dimensional space. All food items are apart from a certain distance between them. Manhattan distance between each node i and i+1 is to be calculated, to find the best path between them. This Manhattan distance is useful to recommend the best relevant food items according to user taste. All the Manhattan distances are calculated and noted separately.

$$\text{Manhattan distance } (M_d) = |h(x)| = \sum_{i=1}^6 (\text{item 1} - \text{item 2})^2$$

item1 = item1 represents the 6 dimension point in the graph
 item2 = item2 represents the 6 dimension point in the graph
 i = 1, 2, 3, 4, 5, 6 represents the dimensions like X, Y, Z

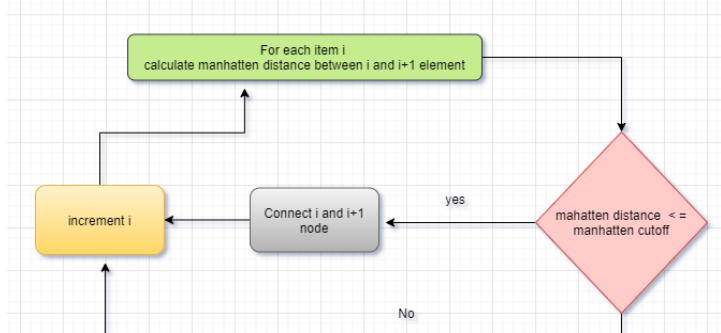


Fig.III Connecting all possible nodes in the graph.

In the Fig. III, as mentioned if the manhattan distance (M_d) is less than that of the manhattan cutoff (M_x) then only the two nodes are connected, if not ignored. [5] This way all the food items that are of same category are connected.

IV. ARCHITECTURE

User taste prediction and recommending food items based on his/ her taste preference is the motto of this model. From the users' previous ordered food items datasets are collected from different users. These datasets are gathered with the help of chatbot we developed for food ordering. In section III, as mentioned first six-dimensional graph, with six tastes on each dimension, is drawn placing all the food items in that six vector space. Secondly, connecting the food items based on the condition, if and only if Manhattan distance (M_d) is less than or equals to Manhattan cutoff (M_x). After completion of connecting all the nodes (i.e., food items) in the graph, then the model is run into the AI algorithms to make the model intelligent enough to recommend food items to the user.

A. Datasets

As mentioned earlier, users' previously ordered food items are gathered from the chatbot, which is developed for food ordering. These datasets of different users are further processed to understand the users' tastes. [6] Taste values of all previously ordered food items are taken and the mean of individual tastes sweet, sour, umami, astringency, salty and bitter have been calculated. To create data point to run A* algorithm, which is helpful to find a relation between taste values of food items and the users' taste preference among all the six different tastes. This data point is like a base point to run the food recommendation model. Above all, users' taste preference is more important to consider, hence, the dataset plays a vital role in the recommendation system.

B. A* Algorithm

A* algorithm is one of the best searching algorithm for the shortest path in Artificial Intelligence with wide range of applications. [7] A* is an advanced BFS and optimal algorithm for path finding and graph traversals.

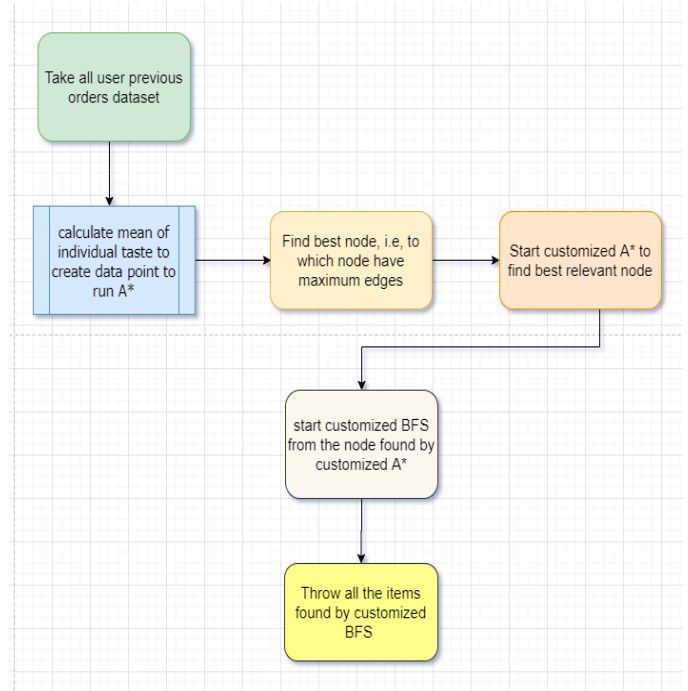


Fig. IV Architecture of Food Recommendation Model

$$g1(x) = h1(x) + h(x)$$

$g1(x)$ represents lowest cost in the neighboring node
 $h1(x)$ actual cost of the path from initial node to any node x
 $h(x)$ represents heuristic estimated cost from node n to target node

Each time A* enters a node, it estimates the cost, $f(n)$ (n being the neighboring node), to travel to all the adjacent nodes, and then enters the node with the lowest value of $f(n)$.

One major shortcoming is its space complexity is $O(b^d)$ as it stores all generated nodes in memory. Hence, to overcome this drawback we have implemented customised A* Algorithm, which subdue this weakness and fabricate this model more efficient and intelligent in recommending food items.

C. BFS Algorithm

Breadth First Search Algorithm is a breadthward motion graph traversal algorithm and it traverse from some arbitrary node to the neighboring node using queue to remember to get to next nodes.

Steps to implement BFS:

Step 1: Hit the neighboring unvisited vertex. Mark it as toured. Display it. Insert it in a queue.

Step 2: If no neighboring vertex is detected, extract the first vertex from the queue.

Step 3: Repeat Step 1 and Step 2 until the queue is emptied.

V. ALOGORITHMS

A. Customised A* Algorithm

The algorithm is mainly designed to get a path in the created graph using manhattan cutoff graph creation in this project This algorithm has the same implementation of a* but a little customized to get our required path from the start node which helps to recommend the food items from the menu to the user

Formule :

$$f(x) = g(x) + h(x)$$

where heuristic function, h(x)

$$h(x) = \sum_{i=1}^6 (item\ 1 - item\ 2)^2$$

Customized A* algorithm uses PropagateImprovement()

```

PropagateImprovement(m)
Algorithm PropagateImprovement(m)
  neighbours <- Children(m)
  for each s in neighbours
  do newGvalue <- g(m) + k(m,s)
    if newGvalue < g(s)
    then parent(s) <- m
      g(s) <- newGvalue
    if s in closed
    then PropagateImprovement(m)
    
```

```

ListNodesFromBestNodeToStart(closed,startNode)
Algorithm
ListNodesFromBestNodeToStart(closed,startNode)
  closed <- reversedList(closed)
  listOfNodes <- list(startNode)
  parent <- parent of startNode
  for node in closed
    if node is parent
      add node to listOfNodes
  return listOfNodes
    
```

```

Customised_A*(start,end)
Algorithm Customised_A*(start,end):
smallest_distance <- infinity
node_to_return <- NONE
open <- List(start)
f(start) <- h(start)
parent(start) <- NIL
closed<- {}
while open is not EMPTY
do
remove node n
add n to closed
If n == end
then return ListNodesFromBestNodeToStart(closed,n)
neighbours <- Children(n)
for each m in neighbours
do switch
case m not in open AND m not in closed
Add m to open
parent(m) <- n
g(m) <- g(n) + k(n,m)
f(m) <- g(m) + h(m)
case m in open
then parent(m) <- n
g(m) <- g(n) + k(n,m)
f(m) <- g(m) + h(m)
case m in closed
then parent(m) <- n
g(m) <- g(n)+k(n,m)
f(m) <- g(m) + h(m)
PropagateImprovement(m)
node_to_return <- n
return ListNodesFromBestNodeToStart(closed,n)
    
```

B. Customised BFS Algorithm:

The customised bfs is also made from bfs which is customized in order to meet the required requirements of project

Since the food items are connected together as a graph and requirement is to visit the child nodes of the parent node so, bfs is used to visit all nodes connected nodes which is branch factor of a graph, to limit the bfs to visit a limited no of nodes the limit is introduced to the algorithm to quit traversing the graph when reaching the limit[8].

Algorithm Customised BFS:

```

CustomizedBFS(graph_list, v,limit=10)
Algorithm
customizedBFS(graph_list, v,limit=10)
count=0 all = []
Q = []
Q.append(v)
while Q != []:
v = Q.pop(0)
all.append(v)
count+=1
if(count>=limit):
return all
for n in graph[v]:
if n not in Q and n not in all:
Q.append(n)
return all
    
```


C. Created graph algorithm

This algorithm creates a graph for given manhattan distance and nodes in the form of adjacency list a created graph plotted on 2d space can be seen in result

CreateGraph(manhattan cutoff, nodes)

Algorithm

```

createGraph(manhattan_cutoff,nodes)
no_of_nodes <- number of nodes
counter <- 1
counter1 <- counter+1
while counter < no_of_nodes
  while counter2 < no_of_nodes
    calculate manhattan distance between
    current node indexed by counter
and
    next node indexed by counter1
    if(manhattan distance <
      manhattan_cutoff)
      connect nodes and vice versa
      nodes
return graph_created
  
```

D. Predict and Recommend user algorithm

This algorithm is the heart and brain of whole project which takes menu items and user orders as input and provide the resultant items as output which are used to recommend the user

E. Predict and Recommend user algorithm

PredictAndRecommendUser(ItemsMenu, userOrders)

Algorithm

```

PredictAndRecommandUser(ItemsMenu,userOrders)
nodes are each item in the itemsMenu
Ak <- list (sum all manhattan distances between nodes)
manhattan_cutoff <- min( Mean(Ak) , Average(Ak) )
//calling createGraph algorithm
graph <- createGraph(manhattan_cutoff,nodes)

//create a node from all orders of users
for each taste in userOrders
  t_sum <- sum of all values in the
  taste
  sum_column_list <- sum_columns.add(t_sum)
row_sum <- sum of all in sum_columns_list
for each taste in userOrders
  user_mean_cluster <-
  user_mean_cluster.add(
  sum_column_list[taste
  number]/row_sum)
//now user_mean_cluster has the required node
bestnode <- most connected node in the graph
if(user_mean_cluster is in ItemMenu)
  allitems=customizedBFS(graph_list,
  bestnode,limit=10)

return {others:[],recommendations:allitems}
else
  
```

```

others <-
  Customised_A*(bestnode,user_mean_cluster)
allitems <- customizedBFS(graph_list,
  bestnode,limit=10)

return
{others:others,recommendations:allitems}
  
```

The output given by **PredictAndRecommandUser** algorithm are

- recommendations for user
- alternatives

Recommendations for user

The recommendations are to be shown to the user

Alternatives

alternatives are the items which user is never ordered but which have a connection with the present food item so that new recommendations and item can be found

VI. RESULT

The Food Recommendation System, that is proposed in this paper, has become more intelligent and advanced with the benefits of customized A* and customized BFS algorithms. This model can able to understand and predict users' taste preference . The result is more appropriate and well-defined, as shown in the fig below. Food recommendation model, is successfully integrated in a chatbot named "FOODIE", which is used to order food. As user orders food, chatbot stores the data and run the machine learning model with that data for his next food order through that chatbot and recommends food items from the result of that recommendation model.

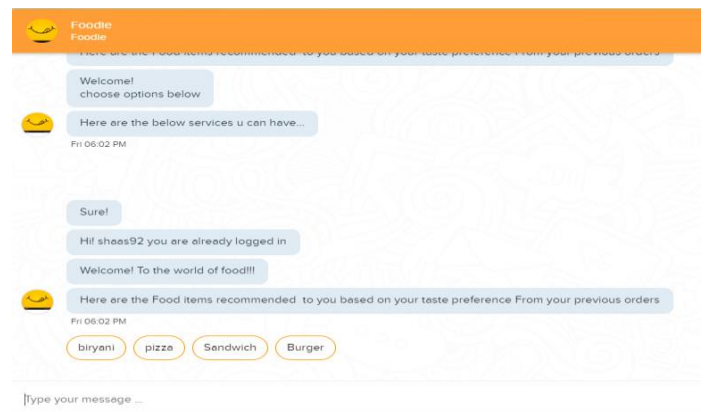


Fig. V Predicted food items

This project uses AI algorithms and shows a way that AI can be implemented to give better and similar results apart from using neural networks and other models [9]

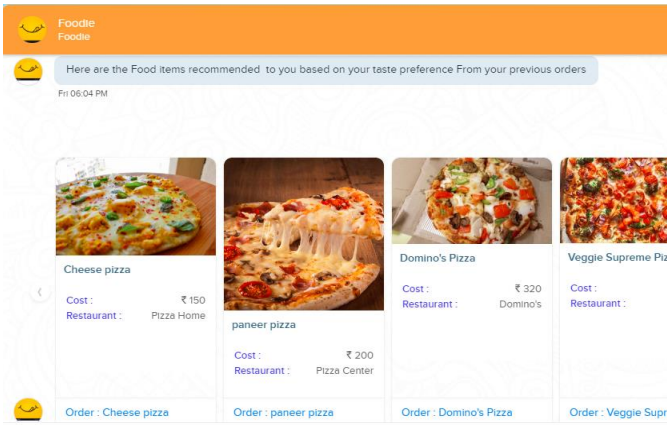


Fig. VI Food items recommended based on users' preference

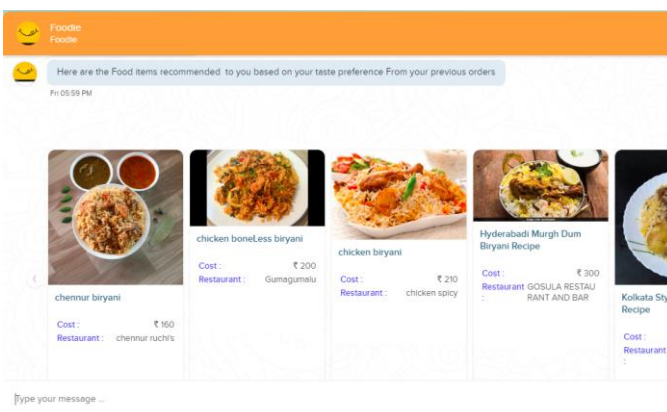


Fig.VII Food items that are recommended to a particular user

VII. CONCLUSION

This food recommendation system, is built intelligently by considering all the factors into account and is helpful for the users to come over the chaos of huge choice of food items. Most importantly, the problem of considering nutritional needs can also be solved with this model. This makes the sense of advanced and intelligent food recommendation system. However, it should be agreed that the dataset considered is small and food items considered is hundred, yet the model gives the more accurate predictions because of the advantage of customised A* algorithm and customized BFS algorithm. Accurate predictions, here, in the sense the food recommendation model that is designed has been implemented in the chatbot and can provide intended food recommendations to the user. The users have rated the recommendation model, if satisfied, with thumbs up and if not satisfied, with thumbs down symbol. The pie chart is shown below for the users' ratings.

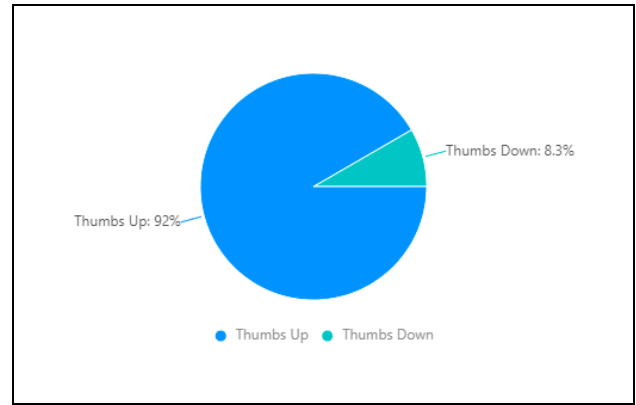


Fig VIII Pie chart of users rating

Here, in the fig VIII users rating for the chatbot where 92% of the users rated as thumbs up which means satisfied with the food recommended by the chatbot.

REFERENCES

- [1] Christoph Trattner and David Elswiler, Food Recommender Systems: Important Contributions, Challenges, and Future Research Directions
- [2] Xinke Li, Wenyan Jia, Zhaofang Yang, Yuecheng Li, Ding Yuan, Zhang, Mingui Sun Application of Intelligent Recommendation Techniques for Consumers' Food Choices in Restaurants
- [3] Asanov, D. (2011). Algorithms and methods in recommender systems Berlin Institute of Technology. Germany: Berlin
- [4] Thi Ngoc Trang Tran, Müslüm Atas, Alexander Felfernig & Martin Stettinger An overview of recommender systems in the healthy food domain
- [5] Drownowski A, Taste preferences and food intake.
- [6] Longqi Yang, Cheng-Kang Hsieh, Hongjian Yang, John P. Pollak, Nicola Dell, Serge Belongie, Curtis Cole, And Deborah Estrin, Yum-Me: A Personalized Nutrient-Based Meal Recommender System
- [7] A* algorithm by Peter Hart, Nils Nilsson and Bertram Raphael
- [8] Artificial search methods for problem solving course by professor deepak khemani
- [9] Faisal Rehman, Osman Khalid, Nuhman Haq, Atta Ur Rehman Khan Diet-Right: A Smart Food Recommendation System.
- [10] Lei Zhou ,Chu Zhang, Fei Liu, Zhengjun Qiu ,Yong He Application of Deep Learning in Food: A Review