A Thesis

For the Degree of Master of Science

A DeepFM model-based personalized Restaurant Recommendation System

Jie Xu

Department of Computer Engineering

Graduate School

Jeju National University

August 2021

A DeepFM model-based personalized Restaurant Recommendation System

Jie Xu

(Supervised by Professor Wang-Cheol Song)

A thesis submitted in partial fulfillment of the requirement for the degree of Master of Science.

2021. 06. 25

This thesis has been examined and approved.

안 기 중

Thesis director, Khi-Jung Ahn, Professor, Jeju National University

김 수 균

Thesis director, Soo Kyun Kim, Professor, Jeju National University

송 왕 철

Thesis Supervisor, Wang-Cheol Song, Professor, Jeju National University

Department of Computer Engineering

GRADUATE SCHOOL

JEJU NATIONAL UNIVERSITY

Dedicated to My dearest parents,

and all Network Convergence Lab members!

Acknowledgment

Foremost, I am grateful to my parents for giving me this opportunity, determination, and courage throughout my study abroad career. I am forever indebted to them for giving me the support and experiences that have promoted my knowledge level. My family has always inspired me into trying my best in every learning phase of my life.

And I would like to express my sincere appreciation to my supervisor, Prof. Wang-Cheol Song, who gave me continuous guidance for completing my thesis work. Without his patient instruction and inspiration, the completion of this thesis would not have been possible. I would like to acknowledge that he played a vital role in shaping my views towards maintaining a correct balance between research work and personal life goals.

Lastly, I appreciate the help and encouragement given by my current lab members Javier, Khizar Abbas, Talha Ahmed Khan, Asif Mehmood, Akbar Waleed, Sarwar Mir Muhammad Suleman, and Dr. Afaq Muhammad. They always provided me a piece of kind advice and help during tough times. Thank you all for your company and growth.

Table of Contents

Acknowledgment..........iv

List of Figures..........vii

List of Tables..........viii

Glossary..........ix

Abstract..........1

Chapter 1: Introduction..........3

Chapter 2: Related Work..........6

2.1 Web Crawling..........6

2.2 MF..........8

2.3 Face recognition algorithm..........12

2.4 Recommendation system..........14

2.5 Limitations of Existing Solutions..........19

Chapter 3: Proposed System design..........20

3.1 Methodology..........20

3.2 CTR prediction based on DFM..23

3.2.1 Data extraction and Web Crawling Technique

3.2.2 Data Cleaning

3.2.3 DFM Parameters and Configuration

3.2.4 Recommendations Module and algorithm

Chapter 4: Recommendation system Experiments and Results........................32

4.1 Tools used...32

4.2 Description of main model..33

4.3 Top-10 recommended Restaurants...35

Chapter 5: Conclusions and future work..38

References...39

List of figures

Figure 1: Workflow of Age-group and gender recognition-based Restaurant Recommendation System..4

Figure 2: Sparrow personal Movie Recommendation example............................9

Figure 3: Proposed recommendation system of MF with implicit feedback.......10

Figure 4: Illustration of the DNN model...13

Figure 5: Recommendation System Overview...15

Figure 6: Types of the recommendation model and industry algorithms 17

Figure 7: Conceptual recommendation program for DFM mechanism.............21

Figure 8: The facial recognition result to use for the DFM model......................22

Figure 9: Recommendation system with DeepFM model of workflow...............23

Figure 10: Architecture of proposed focused web crawler...............................25

Figure 11: The FM model and DNN component share interaction sparse feature which enables DeepFM..27

Figure 12: Based on TF-IDF algorithm with cosine similarity in recommendation system...29

Figure 13: The main procedure to recommend top 10 restaurants……………34

List of Tables

Table 1: CTR-prediction based on ranking..28

Table 2: The system's components setup...33

Table 3: The restaurant recommendation system result.......................................36

Table 4: The industry algorithms approach with application comparison..........37

Glossary

SNS	Social Network Service
MF	Matrix Factorization
DFM	Deep Factorization Machine
ALSWR	alternative least square with weight regularization
SGD	Stochastic Gradient Descent
SVD++	Singular Value Decomposition
CTR	Click-Through Rate
TF-IDF	Term Frequency-Inverse Document Frequency
DNN	Deep Neural Network

Abstract

Due to the continuously growing and overwhelming number of choices on the internet, it is incredibly challenging to extract relevant and meaningful information. So, there is a strong need to efficiently filter, gather, and prioritize the most useful information from the internet automatically that causes to mitigate information overload issues and makes the customers confused. However, the proposed recommendation system solves these issues by filtering a large volume of information and recommending personalized content or products. In this work, a recommendation system technology for food restaurants is applied because there are many choices for the customers to select. Specifically, it becomes challenging for tourists to find a good place to eat. Another significant issue in current systems is that the food is not recommended as per the user's age and gender. To incorporate the factor of age and gender into the recommendation, the proposed system predicts the age and gender of a user before recommending a restaurant. The prediction model predicts gender and age from the user's faces accurately, which supports the process of recommendation to be more accurate and valid. Hence, a DeepFM (Factorization-Machine) based personalized restaurant recommendation mechanism is proposed that recommends restaurants to users according to age and gender more accurately and precisely. Moreover, the technique of web crawling is used to collect restaurant food data, user comments, and ratings from the different websites and trained the DeepFM model.

An open-source Zomato restaurant review data set for training Machine Learning (ML) based DeepFM model is used. This DeepFM-based restaurant recommendation system efficiently recommends top-10 restaurants list to users. Experiments were conducted to generate a list of top 10 recommendations based on Zomato open restaurant reviews data, which were found to be more useful.

Chapter 1: Introduction

Now the Recommendations Systems are becoming ubiquitous in many settings and take many forms, from Netflix-prize recommendation in the e-commerce market, to query keywords in search engines, to friends' recommendations in Social Network Services (SNS) and job-hunting. Currently, the application of Machine Learning Algorithms for recommendation systems is becoming popular. Even so, finding a favorite product per user's case is difficult when considering a massive dataset. Therefore, the development of an effective recommendation system becomes increasingly important.

Firstly, it is using web crawling technology to collect some restaurant reviews. In recent years, the number of information resources on the internet increased exponentially. However, this large data set is not stored locally in any databases for further pre-processing and information extraction. So, users used google, IE, and other search engines to extract information from the internet in a frequent manner, but these search engines provide inaccurate or incomplete results. So, we required a web crawler is an automated program that traverses the always changing, distributed, and dense structure of the HTML pages from the web, store the downloaded pages into a local repository and index them for future usage.

Secondly, we aim to implement a small face recognition system that is dependent on the DNN model. The accuracy for the detection age-gender recognition algorithm is improved. Then I have developed using the DeepFM[1] architecture for ensembles

FM (Factorization Machine) and DNN and to low-order and high-order feature interactions simultaneously from the input raw features from structure user-item interaction as well as to improve recommendation system generalization ability. After, we make the Top-N restaurant recommendation the user's choice.

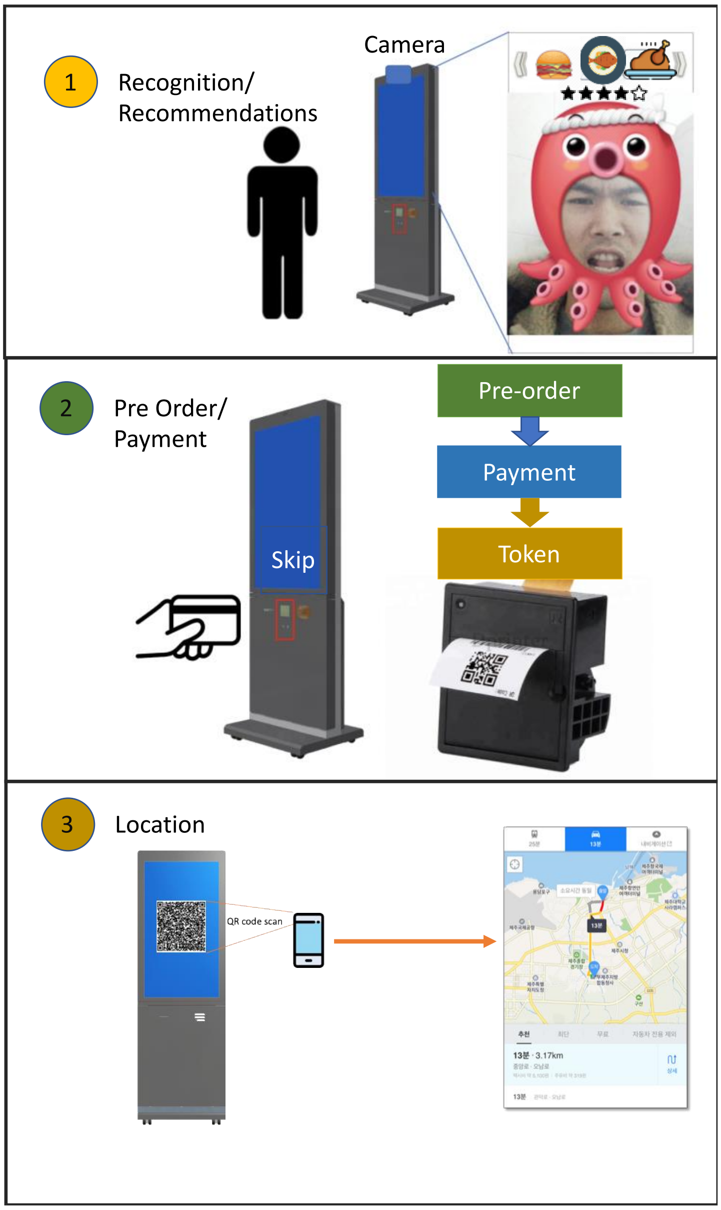


Figure 1: Workflow of Age-group and gender recognition-based Restaurant Recommendation System

In this dissertation, the recommendation application from an open-source data set takes the restaurant reviews such as ratings, votes, and food preference into consideration to recommend similar restaurants. Then an application that can recognize people's faces can recommend a good place to eat basis on the preferences of the user.

Chapter 2: Related Work

In this section, we show the previous related work of the restaurant recommendation systems. As the size of a data set increases, the time for processing the information impacts the performance of the recommendation system. Therefore, it is essential to have an efficient and effective mechanism for filtering useful information.

The related works have been divided into five sub-sections. In section 2.1 we present the works on Web crawling, Matrix Factorization in section 2.2, the face recognition algorithm is shown in section 2.3, and the recommendation system in section 2.4. In section 2.5, we will discuss some present reality defects on the existing solutions. So, given to cold-start and sparsity problems always existed while this research work will be optimization and put forward related DeepFM modeling are investigated.

In this part, background knowledge related to recommendation systems is presented. Here, some recent related research developments are presented, as the knowledge is important for the research in this thesis. Also, a brief introduction to the related work's algorithms and ideas used for this thesis is presented in this section.

2.1 Web Crawling

Web Crawlers' main function is to get data in a faster and more convenient way, to match the corresponding data source from the website. Due to the continuously increasing number of information resources, it is challenging to get semantic information from billions of information resources.

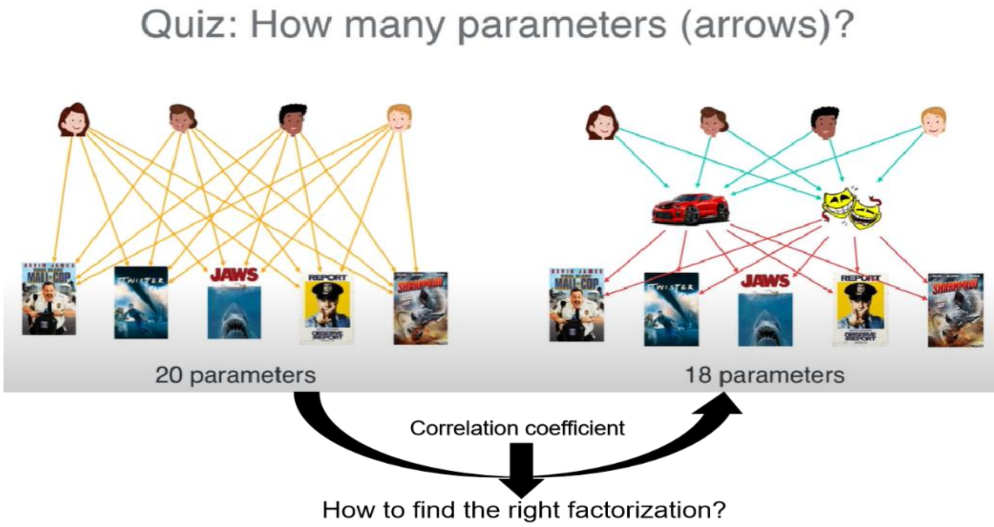
The Internet has an enormous amount of unstructured data known as big data. So, getting desired, relevant, and accurate information from the Internet is complicated. Although several web crawling techniques have been used to extract meaningful data from the Internet, an efficient, robust, and automated web crawling framework is required for extracting accurate content from the web with minimal human effort. Web crawling is divided into four major types: focused, incremental, hidden, and distributed web crawlers.

The CDN (Change Detection and Notification) systems that could automatically detect the changes made to static web pages or dynamic web pages on the Internet with web crawlers [2], notify about the changes, and process and extract the contents related to the user's interesting. For instance, the Dongcheng Peng [3] researchers' work provided a shipping job hunting crawler on a real recruitment platform to get user's personal information. Compare to traditional crawlers they are using a Scrapy architecture that can improve more efficiency and decrease complex workload. Another research work [4], presents a deployment and a workflow chart with a crawling process. The authors propose a parallel crawler by using a multiprocessing technique just like multiple computers operating at the same time for optimizing the parsing of the website information.

The web crawling process is time-consuming, so it needs to be fast as well for extracting a large amount of data in less time. Many developers used python-based open-source libraries such as selenium and Scrapy architecture can help develop the crawler and to improve the process of crawling with the above-mentioned criteria.

2.2 Matrix Factorization

As this part, Explicit and Implicit feedback of the user's preference can be obtained from movies, restaurants, books, or friend's recommendation system. Then reflect this information by embedding it on a user-item matrix list. The explicit feedback e.g., (1–5-star scale, liking a friend's Facebook post, or writing a personal perspective review for a restaurant) is easier to identify. On the contrary, implicit feedback is more subtle and attempts to infer a user's preferences based on their indirect behaviors. Examples of implicit feedback include a user's purchase history or items inside a shopping cart, how many times they replayed or clicked on a video, etc.



Therefore, We make sense of a large amount of information is used to make a user-item matrix. So, when we apply the recommendation system it should separate each item according to the corresponding genre, e.g., (comedy movie, action movie, romance movie). Then, the MF model using ALSWR (alternative least square with weight regularization) will have a minimal loss function with feedback embedding data sparsely. How to find the correlation coefficient with the right factorization is an important job. Normally, SGD (stochastic gradient descent) and ALSWR (alternative least square with weight regularization) are used in the collaborative filtering, also, SVD++ objective function could be used for the minimized loss functions. Sparrow recommendation results are shown in Figure 2.

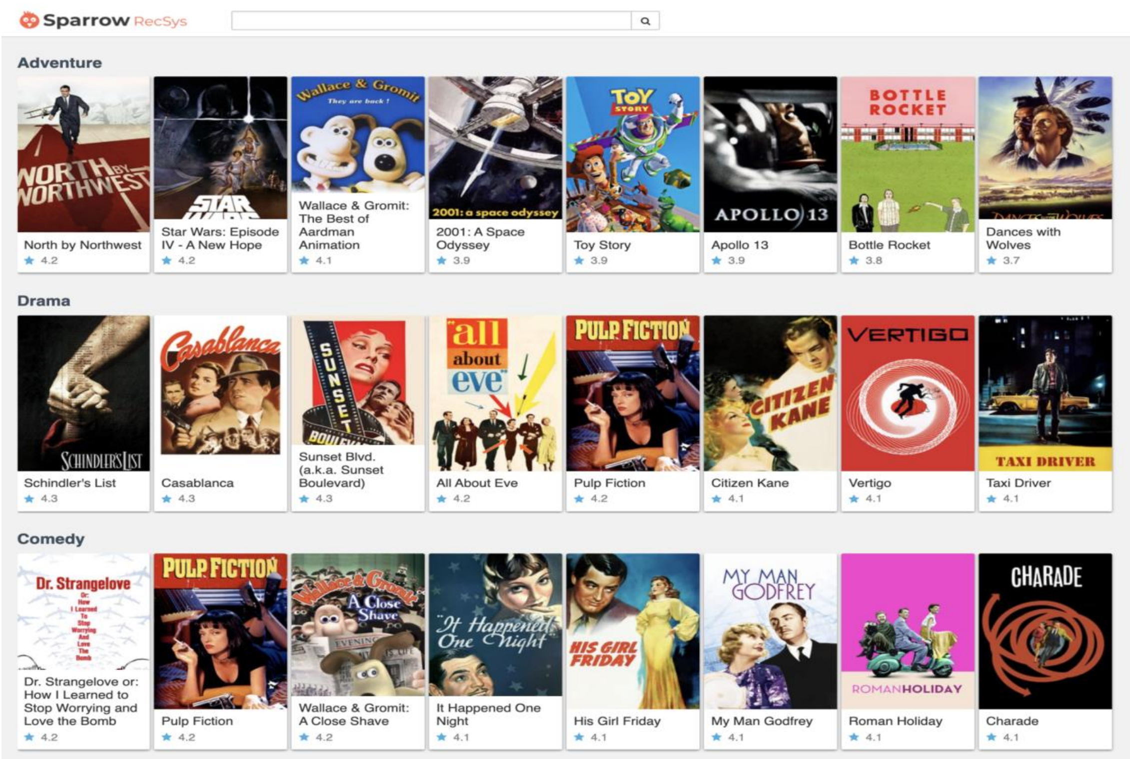


Figure 2: Sparrow personal Movie Recommendation example

The inner product of the user vector and the item vector represents the predicted score of user 'u' for item 'i', the objective function of the matrix:

$$\mathrm{X} = [\chi_1, \chi_2, \ldots, \chi_N]$$

$$\mathrm{Y} = [y_1, y_2, \ldots, y_M]$$

$$\min_{X,Y} \sum_{r_{ui}\neq 0} (r_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u ||x_u||_2^2 + \sum_i ||y_i||_2^2\right)$$

In this function, we define the user embedding matrix with X, and the item embedding matrix with Y, the r_{ui} stand for user 'u 'rating of item 'i.' If $r_{ui} > 0$ indicates that there is a score if $r_{ui} = 0$ indicates that there has not to score to be represented for prediction p_{ui}, λ is used for L2 regular term to ensure the stability of numerical calculation and prevent over-fitting and as usual, determined by cross-entropy validation.

$$\boldsymbol{P_{ui}} = \begin{cases} \mathbf{1}, & \boldsymbol{r_{ui}} > \mathbf{0} \\ \mathbf{0}, & \boldsymbol{r_{ui}} = \mathbf{0} \end{cases}$$

After that, the characteristics of the MF model are used to model the customer's implicit feedback and update the data to make a scoring prediction recommendation system.

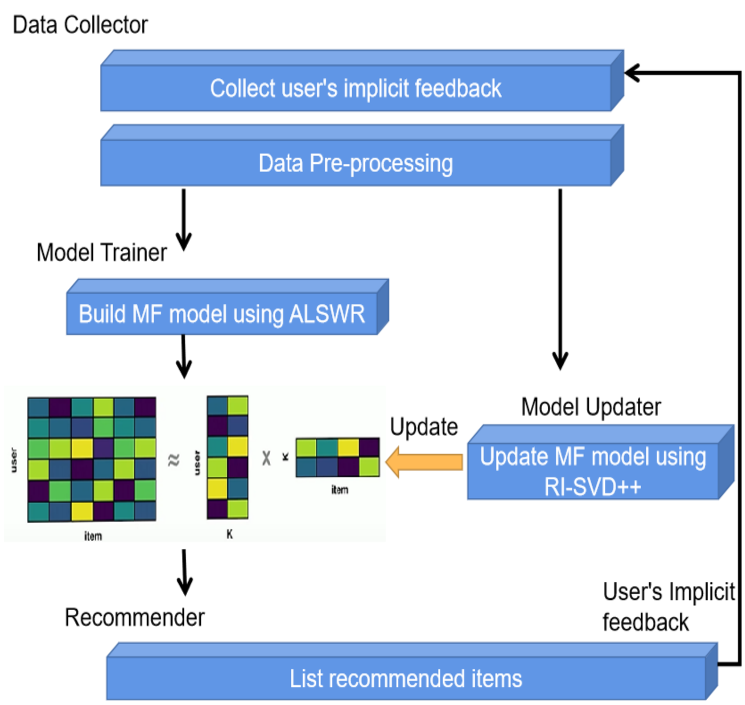


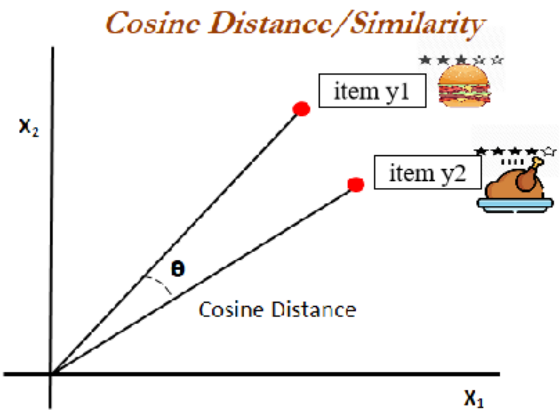
Figure 3: Proposed recommendation system of MF with implicit feedback

In [5], CHIA-YU LIN has proposed a good implicit feedback system as shown in Figure 3. The framework firstly collects the user's implicit feedback to get data and then having the data pre-processing work. This trainer model builds MF methods using ALSWR and is dependent on an objective function. In the last step, it gets the similar items recommended list and feeds the user's implicit feedback to the data collector module such as order, click, and skipped thus behavior.

On the grounds, the authors have summarized four types of k-nearest neighbors-based recommendation methods. Computing two-item vectors such as cosine vector similarity, Pearson correlation coefficient, Euclidean distance similarity, and Jaccard similarity coefficient methods so on. The proposed algorithm uses the deployment of the cosine similarity function [6] in the restaurant recommendation system. The cosine similarity is more familiar with other methods and flexible to apply. Finally, the weighted average is used for each restaurant's average rating with the user's votes.

$$\textbf{Similarity} = \textbf{cos}(\boldsymbol{\theta}) = \frac{y1 \cdot y2}{|y1||y2|} = \frac{\sum_{i=1}^{M} y1_i y2_i}{\sqrt{\sum_{i=1}^{M} y1_i^{\,2}}\sqrt{\sum_{i=1}^{M} y2_i^{\,2}}}$$

As the above diagram shows, this θ angle is the neighborhood restaurant between Item 1 to Item 2 distance. According to the characteristics of the triangle theorem,



normalize this angle difference in the interval [-1,1], where 1 represents the same direction, so it is perfectly similar, -1 represents the opposite direction, so there is no similarity at all.

2.3 Face recognition algorithm

With the growing population in the world, various personal safety test equipment is widely used. In China, the government has a large project to support the detection of facial data, which has played an important auxiliary role in reducing the probability of crime for national security. In addition, some banks will use face detection systems to verify identities. A real-time camera has been implemented on the cv2.dnn module [20] to detect people's age group and gender which is then used for a recommendation system.

The CNN is widely used for applications such as image classification and retrieval, target location detection, target segmentation, face recognition, human pose recognition. Many researchers focus is on making better accuracy using deep learning techniques. A survey has shown the CNN model and GAN model [7] benefit more in the image the process through dimension reduction. There have present the popular typical neural network architectures in object classification Because there has excellently improved the state-of-the-art performance and fostered successful real-world applications. Of course, more researchers want to use a Deep Learning collaborative style development recommendation system like the NCF algorithm. The image process is used to handle loss function with training and testing. Commonly, the full dataset was separate to normally have 80% data for training and 20% data for testing to validate the loss accuracy. So, in recent years deep learning technique develops rapidly. As known as the father of deep learning Hinton [8] proposed completed work using the Back Propagation algorithm to indicate how a machine should change its internal parameters like mini-batch normalization and activation function that is used to compute each layer's weight connected to the layer nodes. In the CNN algorithm, a convolution layer is used to extract image features and the maxi-pooling is used to reduce the dimension and avoid over-fitting which also makes the features more prominent.

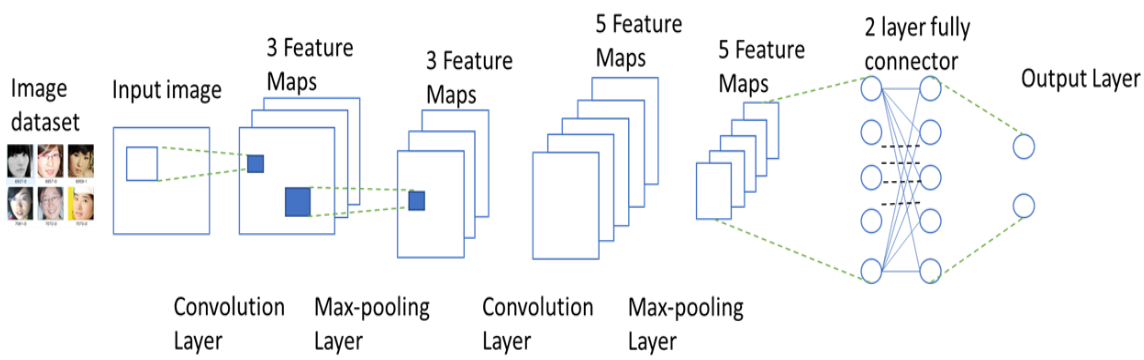


Figure 4: Illustration of the DNN model

Lastly, Mellouk [9] showcases research related to facial emotion recognition system which aims to capture real facial images through a front camera and conduct processing/recognizing functions. To improve precision, 3 hidden layers and 2 fully connected layers are used Each layer uses the Relu activation function to further promote detection of the facial expression by using the pre-trained DNN model shown in Figure 4.

2.4 Recommendation System

With the continuous development of software applications, the recommendation system has been widely used from the initial news prototype to the current in-depth research and diverse application. Some real-life examples of this can be seen in news, movie, book, SNS, restaurants and fashion recommendations, etc. That one way makes it easier for users to experience the best products, R&D on a strong recommendation system is done to help us decide by learning our preferences. The concept is shown in Figure 5.

A variety of techniques including RecSys are summarized on [10] where a recommendation system is based on a popular deep learning algorithm. Traditional types of these algorithms have evolved according to the individual needs of customers, each has its strengths and weakness.

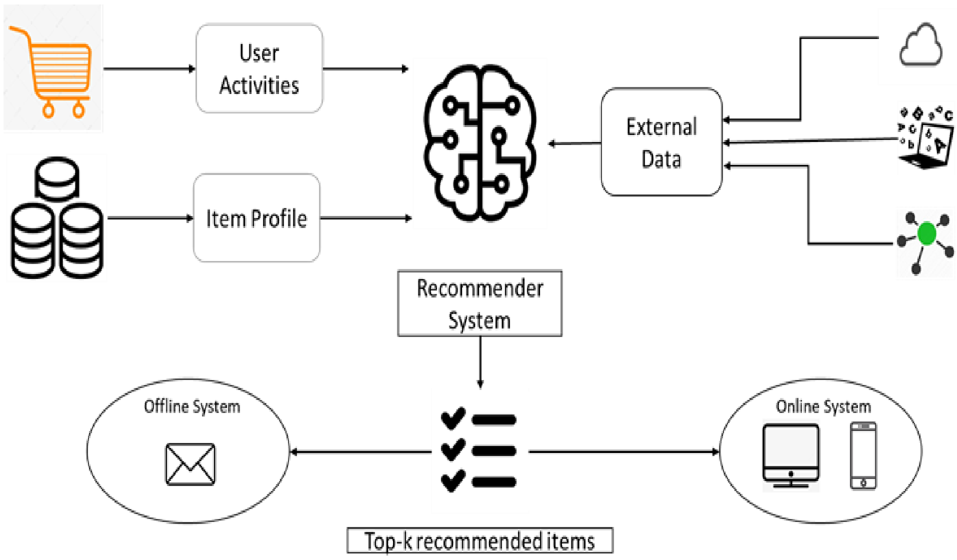


Figure 5: Recommendation System Overview

Therefore, from the RecSys users are suggested items based on their previous behaviors and their similar hobbies relationships. Normally, the recommendation system has two different missions： Rating prediction and Top-N list recommendation. Considering the rating prediction, during the Netflix-prize algorithm competition the research team AT&T-group proposed a solution named BellKor [11] which has 8.43% higher optimized RMSE performance than the second place Cinemamatch. The researcher paper also showcases SGD, ALS, SVD++ basic user-item matrix factorization model. On the other hand, the best example for the Top-N list recommendation is the well-known YouTube platform which is an extremely popular video recommendation site. YouTube users can get the latest news reports and various video content, and the recommendation system can also follow up in real-time and collect the user's usage data for accurate Top-N list recommendations. On [12] the authors have proposed a personalized product recommendation system architecture and a demonstrating into YouTube APP; First, millions of user data is dependent on user preference. The candidate generation model could be capturing more features, more complex models, as well can avoid scoring candidate generators. The different objective functions will lead to different learning results depend on maximum click-through rate and maximizing viewing time. After filtering out hundreds of items to the recalled model; A click-through rate estimation model is used to evaluate the scores of video candidates, and then pushed to a Top-N of best-recommended videos to the user interface.

The personalized and non-personalized recommendation system's properties and their shortcomings are found in this research work. In [13] survey recommendation system methods, massive amounts of data are needed to identify the interest of users and make the information search easier. The recommendation system that seeks to predict the ratings and clicks of a user is primarily used in e-commerce applications between both websites and users.

Various recommendation models are illustrated in Figure 6. There exist three basic categories for a recommendation system, content-based filtering, collaborative filtering, and hybrid filtering. The content-based algorithm is used to predict what you want to find based on what another person liked in the past by using some services. The CB-based shortcoming is the difficult recommendation to other users in contrast.

Secondly, collaborative filtering is using predict what the users like based on what other similar users have liked in the past. This algorithm is divided into two main types: memory-based and model-based.

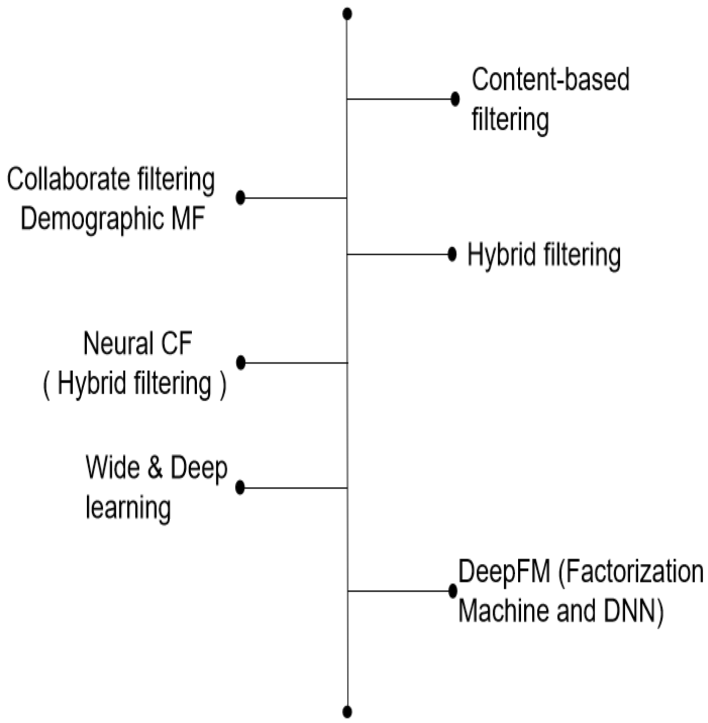


Figure 6: Types of the recommendation model and industry algorithms

The memory-based algorithm recommendation reads all the user data into memory for calculation, this model-based recommendation data using machine learning is divided into training and testing data. Offline training takes a long time, but after the training is completed, the recommendation process is faster. But the CF limitation is cold start [14] for a new user or new products and the sparsity problem. As these two recommendation systems cannot satisfy actual needs, they have evolved into a hybrid filtering algorithm with stronger integration. The hybrid algorithm adopts a way of complimenting each other's strengths to make improvements.

In Xiangnan He research work [15], the DNN has positive deployment on speech recognition, computer vision, and NLP (natural language processing), then from a user-profile produced explicit feedback and implicit feedback. The NCF model with non-linearity function to user-item inner vector on the implicit feedback of their latent factors. There is also mention of neural network architecture to simulate latent factors of user-item and design a general neural network-based collaborative filtering framework of NCF. A wide & deep learning approach is proposed in [16] to define the DNN that can easily generalize features interaction on low-dimensional dense embedding learned from the sparse features. Their program also proposes some improvements using Relu activation against the NCF model’s inability to make an effective online recommendation. Lastly, the DeepFM model is introduced in [17,22], This model can further refine the user’s behaviors from the implicit feedback on the CTR prediction models. Two architectures are proposed: FM and DNN. The embedding method is the way a numeric vector “represents” an object. It is a typical method of dealing with sparse features is also an effective method to fuse large sparse features for generating dense feature vectors. The researcher paper compares the performance results of various algorithms.

2.5 Limitations of Existing Solutions

Despite the many applications of recommendations algorithms that currently exist, there are still conditions that need to be satisfied for optimal use cased. For example, in commercial deployments, the recommendation system should consider server performance. A deficient performance in both academia and industry is still present in recommendation systems [11,18], mostly due to a strategy that can block more than 90% of the duplication of recommendation requests. Many recommendation architectures cannot tackle this point well. Also, the cold-start problem needs to be considered [19, 25] when the platform has new users or newly added items. In addition, in many papers, the real-time performance optimization services were not considered for the recommendation to customers. As well, due to the increased user data related to these recommendation services, individual data representation becomes sparse, and the recommendation result becomes less precise.

Chapter 3: Proposed System design

In this chapter, the description, and details of the DeepFM approach are presented. An FM (factorization machine) model to generate more interactive features and the DNN approach to capture non-linear characteristics at latent relationship is proposed. Also, a one face recognition system for detection of age-group and gender with the cv2.dnn python open library is applied. Commonly, the OpenCV provides two functions to facilitate image processing for deep learning classification. The open data set for the Top-N restaurant recommendation has been used for experiments and results.

In section 3.1, a simple working mechanism is proposed. In part 3.2 the design of the DeepFM model is presented which covers data collection and data cleaning work, the proposed model methods are analyzed step by step. And some redundant or useless data columns are deleted. Then finally the recommendation of the product is based on the cosine similarity and the user profile.

3.1 Methodology

The proposed system is for a restaurant recommendation that uses OpenCV and DeepFM model. In the field of deep learning and image classification, pre-processing usually includes mean subtraction and scaling by some factor. The 'blob' parameter [20] is used, which refers to a group of linked pixels that share common properties in an image.

The dark connected areas in the image are a blob and the goal of blob detection is to mark these areas. OpenCV's new neural network module DNN contains two pre-processing functions for classification through pre-trained deep learning models. A detailed methodology already used to develop the application is described in the following subsections.

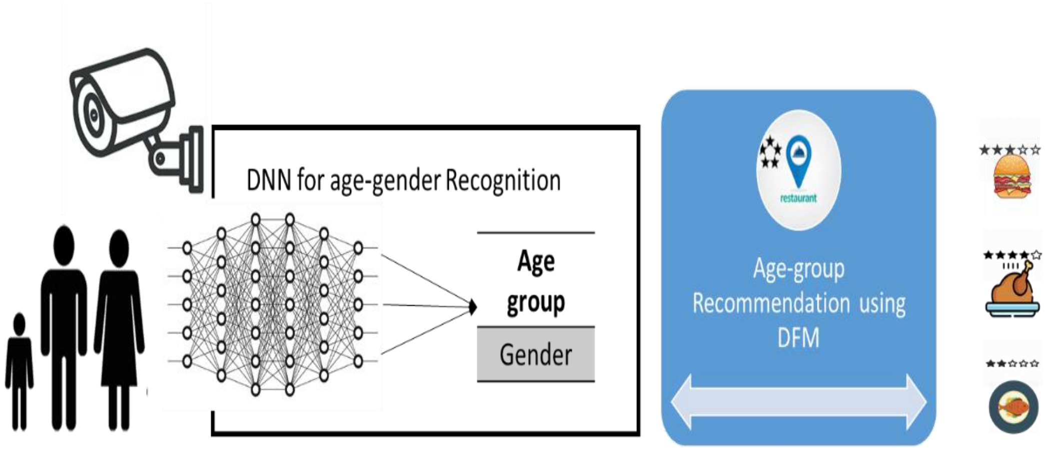


Figure 7: Conceptual recommendation program for DFM mechanism

As shown in Figure 7, this recommendation system is based on diverse user votes related to the restaurant for the recommendation mechanism deployment. The same method could be used for a movie, book, e-commence product recommendation, or services like google scholar. Due to the individual recommendation attributes, there recommend the inside mechanism features also differ.

The proposed recommendation system is separated into several modules. (a) restaurant reviews analysis module which depends on MF and embedding (b) A deep learning model technique used for face detection and a DFM model used as the neural network to simulate the MF (c)cosine similarity for the recommendation module.

We try to frame a gender and age detector that can around guess the gender and age of the person's face in a picture or through the front camera video. The theoretical framework of our proposed system can be seen in Figure 8. It uses the cv2.dnn function and a camera to deal with images or frames directly and quickly.

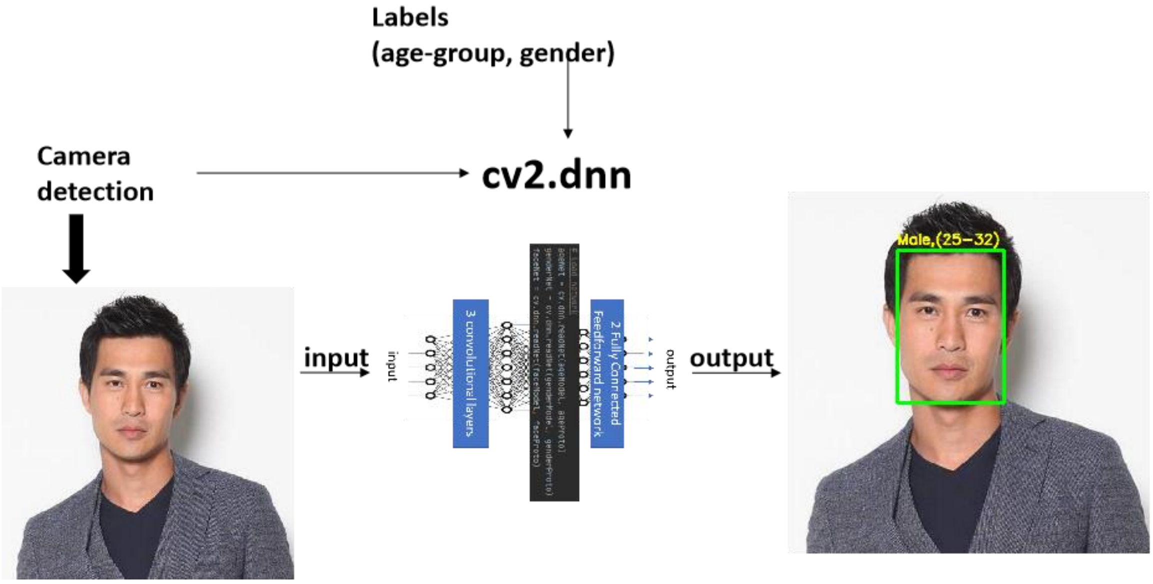


Figure 8: The facial recognition result to use for the DFM model

Normally, every image has three channels B, G, R in the OpenCV. So before training the DNN model can calculate the image average performance with μ_B,μ_G,μ_R value as shown in equation (1), (2). However, in some cases, the mean value of R, G, B is calculated per channel rather than per pixel, and the result is a mean of the M*N matrix. In this case when the input image is being training or test.

$$B = B - \mu_B; \quad G = G - \mu_G; \quad R = R - \mu_R \quad (1)$$

And have a scaling factor addition in a normalization: The value of σ may be a standard deviation of the entire training set.

$$B = (B - \mu_B)/\sigma\ ; G = (G - \mu_G)/\sigma\ ; R = (R - \mu_R)/\sigma \quad (2)$$

Three networks have been loaded to the separated program by python language to achievement: ageNet, genderNet, faceNet for detection of user's face. Although the accuracy rate is not incredibly good, a more sophisticated recommendation system is proposed given the relationship recognition [21] network. Customers can be represented as family, friends, non-Intimate, or teacher-student relationships. Therefore, we get the age and gender recognition results to be good at this research to cooperations with personalized recommendations based on mutual trust and a set menu that suits multiple customers are proposed.

3.2 CTR prediction based on DeepFM

The DeepFM model for user-item recommends preference is described and illustrated in Figure 9. In section 3.2.1 the data collection is presented, section 3.2.2 describes the Data Cleaning process. In section 3.2.3 the DeepFM model configurations and details of the CTR prediction that is used are thoroughly described. Finally, the proposed recommendation module and algorithms design is presented in section 3.2.4.

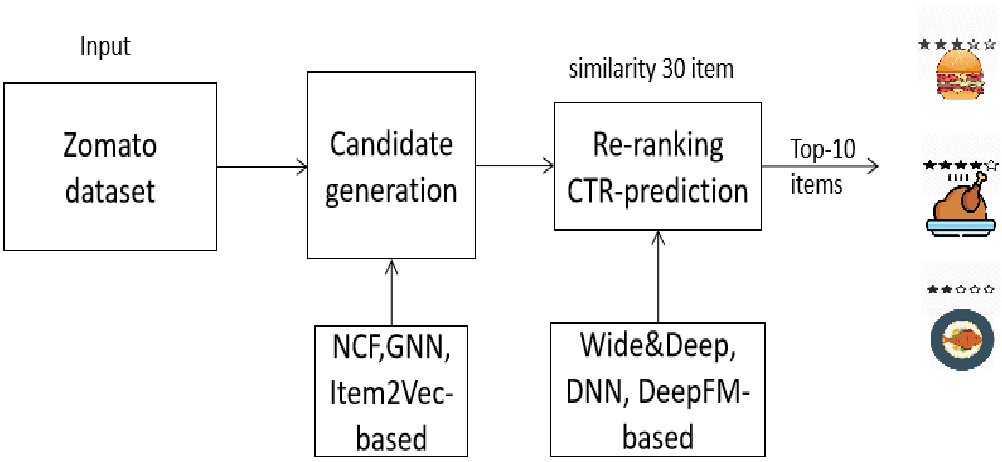


Figure 9: Recommendation system with DeepFM model of workflow

3.2.1 Data extraction and Web Crawling Technique

Firstly, a collection of desired data required for Web Crawling is obtained from the internet. For example, if a user wants to get information from the internet, he/she should know the exact address of that webpage or move from link to link patiently to seek the desired information. Getting a massive amount of relevant information from the internet through search engines is a very time-consuming process [23]. To overcome these issues, an efficient and automatic way is used to get more accurate, relevant, and complete information from the internet, known as a web crawler. A web crawler is an automated program that traverses through the ever-changing, distributed, and dense structure of the HTML pages from the web, stores the downloaded pages into a local repository and indexes them for future usage.

The web crawler is one of the essential components of the search engines, also known as an automatic indexer or web spider. It is an automatic program that can browse the WWW systematically. The growth of the WWW caused the increase of web crawlers exponentially. Web crawlers maintain the list of URLs, and each URL is named as a seed. Firstly, it starts the crawling process from a seed URL and visits the pages in the list, then it downloads the pages and retrieves the hidden URLs in the pages, and stores the discovered hyperlinks into a queue. The crawler repeats this process and extracts meaningful information from web page to web page [24].

The general web crawling process is shown in Figure 10. Also, the used BeautifulSoup4, XPath, Request, and selenium tools have been used for the implementation of the focused web crawling framework.

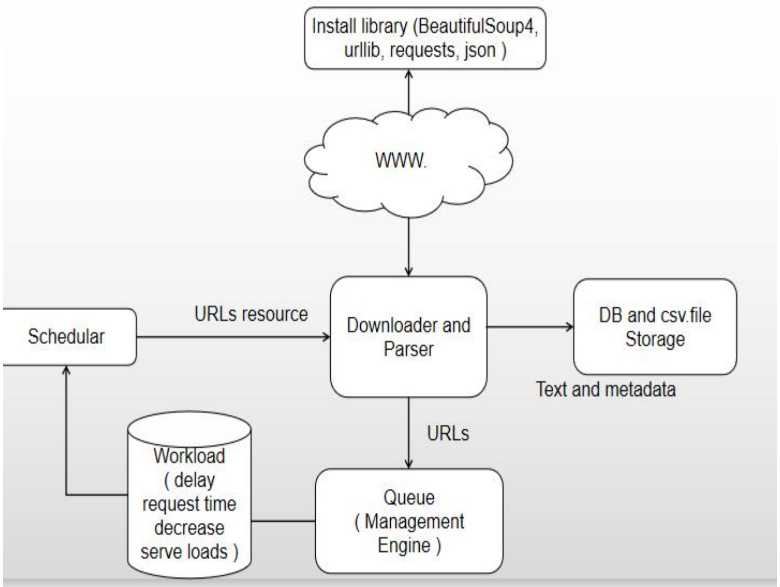


Figure 10: Architecture of proposed focused web crawler

In our program, a virtual customer data set needs to be created by using the open-source zomato.csv data set. Before, there were many incomplete data sets. Some of them have been expanded to get the explanation of the raw data set, for this, the dataset is separated into two parts: a feature similar group and CTR-prediction score similar group. So, this data is used to provide the votes as input in the restaurant recommendation process.

3.2.2 Data Cleaning

The data pre-processing is necessary work for all machine learning scholars before training the model. So, it is important to clean any data duplication and empty data before applying an algorithm. This program is responsible for multiple python functions that delete unnecessary columns, URLs from the raw data set and change some of the column names. This pre-processing work is used for obtaining a meaningful label to make a random sample. After getting the vital data columns to the recommender.

$$W = \left(\frac{v}{v+m} * R \right) + \left(\frac{m}{v+m} * C \right)$$

We are using a weighted average for each restaurant's average rating. The "W" is weighted rating, "R" is average for the restaurant services as a number from 0 to 5, "V" is several votes for the restaurant, "m" is the minimum votes required to be listed in the top 10, and "C" is the mean vote across the whole report.

3.2.3 DFM Parameters and Configurations

In this sub-section, the details of the working mechanism are described. Figure 11 shows the configuration and parameters.

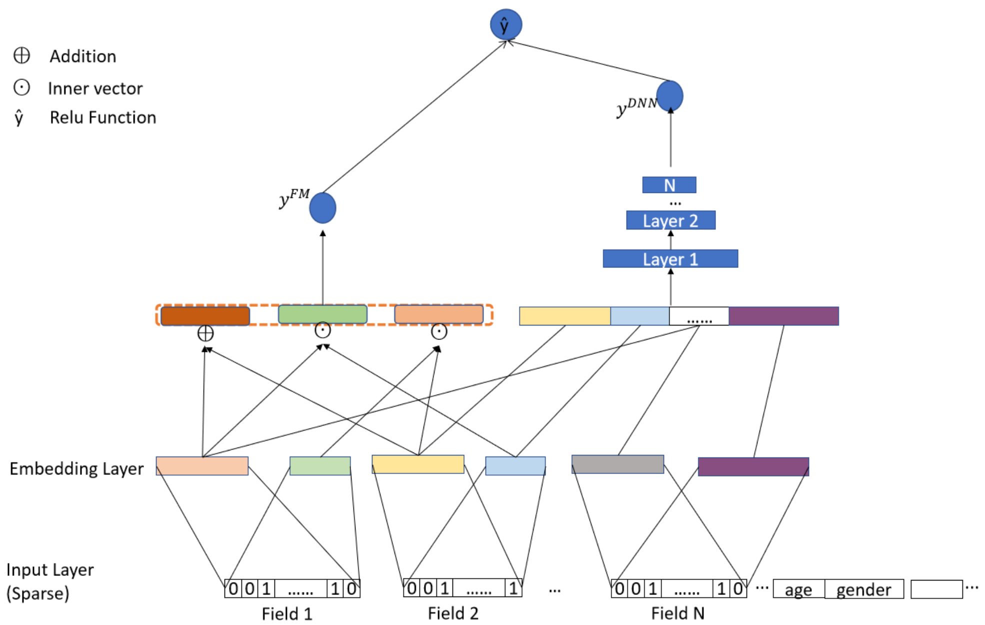


Figure 11: The FM model and DNN component share interaction sparse feature which enables DeepFM

As the user preference profile in the input layer supports many sparse features, a one-hot is used to define each item to make a dense interaction model. Firstly, x consists of N fields and each field contains various user profiles. For example, the user field contains information related to the user's id, gender, and age; The items field contains information about the purchase of a specific item. Each field can be used as a one-hot vector in the case of a category feature, and the corresponding value could be continuously used for overriding. In general, x is very sparse and high-dimensional in this case.

The CTR prediction model is based on the values "0-1" the means exactly click behavior that item how much probability. This is used to make the ranking number for the prediction of the next user, which is shown in the example of Table 1. The ranking item29 is number 1 and item 22 is number 6 according to the sequence of recommendations.

Table 1: CTR-prediction based on ranking

recommended items	CTR-prediction	rank
item 22	0.1	6
item 29	0.9	1
item 36	0.3	4
item 7	0.6	2
item 199	0.2	5
item 1000	0.5	3

In our proposed flowchart, we have explained the FM and DNN model to collaborate work theory in the following function.

$$\hat{y} = \boldsymbol{CTR\_model(x)} = \boldsymbol{DeepFM(x)} = \boldsymbol{y^{FM}} + \boldsymbol{y^{DNN}}$$

The basic input layer has many sparse fields after with embedding layer uses the embedding of items to directly obtain their similarity or as an important feature input recommendation model for training. Consequently, from the embedding raw feature to FM model and DNN based model for generating the prediction result. In the embedding layer, some of the original user-item features are directly used for input into the DNN model, and some will form interactions such as attribute additions and mutual inner product to generalize new attributes in the FM model; According to the research and observation about this Thesis, there are still many companies using artificial methods to extract the feature labels, but the emergence of the FM model has reduced this human workload. Lastly, the Relu function is used to output the layer in CTR prediction.

3.2.4 Recommendations module and algorithm

The primary highlight of this section is the restaurant recommendation module based on the TF-IDF algorithm with a cosine similarity function. At purpose used in the workflow is shown in Figure 12.

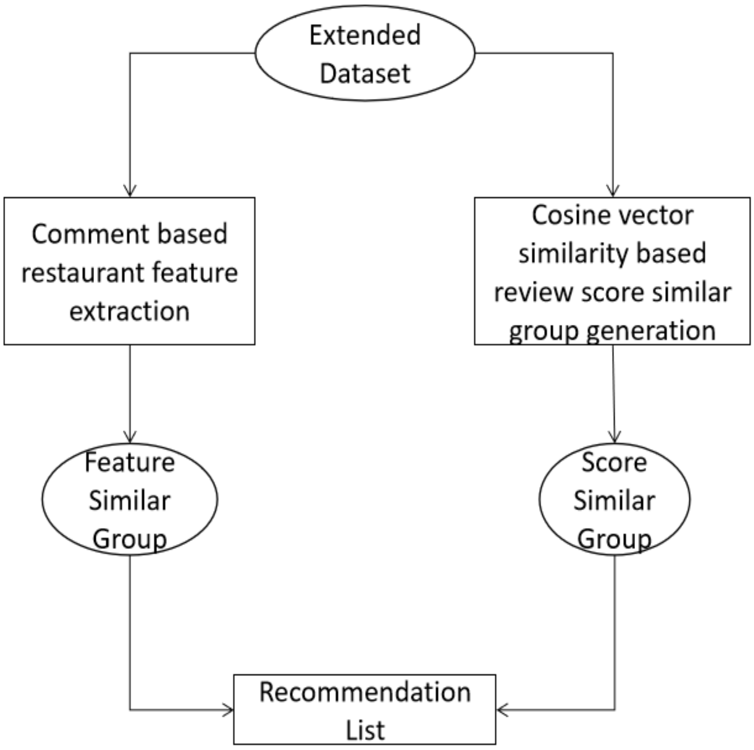


Figure 12: Based on TF-IDF algorithm with cosine similarity in recommendation system

The TF-IDF (Term Frequency-Inverse document frequency) is a commonly used extension technique for information retrieval and data mining. From this point of view, as the name implies, it is often used to mine keywords in articles, and the algorithm is simple and efficient. It is used to obtain how much probability exists for the consumer's good reviews or "no comment" reviews. It is often used in the industry for data cleaning then filter the upper reviews list and cuisines.

So, depending on the data set reviews lists and columns, we use the TF-IDF character to make a matrix with a cosine similarity function to calculate the distance of each item. A new data set with comment-based restaurant features and scores or votes to show similar restaurants is obtained. The user's votes can provide convincing evidence of the recommendation module. Finally, in the recall progress, 30 restaurant indexes are extracted by using the best similar cosine value. The highest scoring restaurant is presented to the user, ranked by their score with votes.

Algorithm 1: restaurant recommender system using DFM to generation features and CTR-prediction
1. Input: zomato.csv data set with user-item, rating tables, votes tables 2. Output: Top-10 recommended restaurants list 3. Procedure: 4. Create the relationships between user table 5. depend on user-item matrix and user profile we have using Factorization Machine to prediction model is created 6. Cosine similarity are matching the closed-group for new user 7. Count restaurants rating and votes 8. recommended restaurants list

In this dissertation, we use the DFM model and user profile analysis by using personalized generation feature methods and propose an effective restaurant recommendation system. This provides the best CTR prediction about the list of restaurants that the user may have an interest in. The designer algorithm of this proposed method is shown in <Algorithm. 1>

Chapter 4: Recommendation System Experiments and Results

In this chapter, all sections provide a detailed discussion on the performed testing and those corresponding results. The testbed uses a python programming language to execute the algorithms. Section 4.1 presents the system's components and specifications. Section 4.2 showcases the main recommendation model summary for the application methods. Lastly, the public data set for testing the Top-10 restaurant recommendation results have presented in section 4.3.

4.1 Tools used

The performance component of the actual operating system is presented in this sub-section. The process starts by choosing the source of data and making a public data set, then cleaning and running the implemented module, also the testing is performed with the use of many auxiliary tools. A recommendation mechanism is built upon these CTR-prediction that is also capable of optimizations. It contains the preferences of users and can recommend items with trust relation levels accordingly.

Table 2 presents the composed system components and specifications. the MS Excel was used for saving to the data set. The Scikit-learn API is a simple and efficient data mining development that uses data analysis tools. It encapsulated commonly used normal Machine-Learning methods, including regression, dimensional reduction, classification, clustering, and other algorithms.

Table 2: The system’s components setup

System Details	Characteristics
OS (Operating System)	Windows 10
Primary Programming Language	Python
System Memory	16Gb
CPU	Intel (R) Core (TM) i9-9900 CPU @ 3.10GHz
Python Version	3.6.8
Library	Scikit-learn

4.2 Description of the main model

One of the major python classes used in the implementation of the application is the sub-class that calculates the similarity between the user-item using the Similarity(cosine) function. The cosine function is used to calculate the similarity between the items based on the users who rate the items. Then the recommendation system scores the predicted likeliness on a top-10 list.

A separate module with the python function to produce this recommendation system is defined using the main model workflow shown in Figure 13. In the end, the two theme restaurant names from the raw data set for recommendation have been randomly selected.

```
procedure DEF RECOMMEND(name, cosine_similarities =
cosine_similarities)
    recommend_restaurant = []
    score_series = pd.Series(cosine_similarity[idx]).
    sort_values(ascending = False)  ▷ Calculate cosine
similarity score
    top30_indexes = list(score_series.iloc[0 : 31].index)
▷ recall to get 30 restaurants
    foreachintop30_indexes :
    recommend_restaurant.append(list(df_percent.index)[each])
    df_new = pd.DataFrame(columns = ['suisines',
    'MeanRating','votes'])
    foreachinrecommend_restaurant :
    returndf_new
    recommend('AmoebaSportsBar')
    recommend('GrandVillage')
end procedure
```

Figure 13: The procedure to recommend top 10 restaurants

4.3 Top-10 recommended Restaurants

Existing collaborative filtering models suffer from the sparse and cold-start problem. An interaction feature to make a personalized recommendation system is proposed. After a series of systematic extraction schemes, we set out to proposed algorithms to further arrive at this goal. This recommendation result is illustrated in Figure 13. We have promoted mean ratings accuracy and more votes the means others many favorites to this restaurant would be proposed in our system.

Due to time constraints, we have not been able to completely crawl the required information from some diner's APP. Because there are many open-source data on the Internet, we analyze this part of the public data to provide our recommendation algorithm requirements. Then such a useful list from the Zomato data set is filtered out. Also, the user-item matrix could be built by using this main data.

0	name	rate	cuisines	votes
1	Jalsa	4.1/5	North Indian, Mughlai, Chinese	775
2	Spice Elephant	4.1/5	Chinese, North Indian, Thai	787
3	San Churro Cafe	3.8/5	Cafe, Mexican, Italian	918
4	Addhuri Udupi Bhojana	3.7/5	South Indian, North Indian	88
5	Grand Village	3.8/5	North Indian, Rajasthani	166

Table 3: The restaurant recommendation system result

	cuisines	Mean Rating	votes
The Black Pearl	North India, European, Mediterranean	4.78	10413
The Black Pearl	North Indian, European, Mediterranean, BBQ	4.78	7023
Deja vu Resto Bar	North India, Italian	4.35	2493
Amritsari Kulcha Land	North Indian	3..86	260
Village – The Soul of India	North Indian, South Indian	3.85	569
Real Fresh Dosa Corner	South Indian	3.84	267
Real Fresh Dosa Corner	South Indian	3.84	31
Curry With A ‘K’ – St. Mark’S Hotel	North Indian, Mughlai, Hyderabadi	3.71	90
Café @ Elanza	Chinese, North Indian, Café	3.45	145
Desi Dhaba	Chinese, North Indian	3.19	303

The real world has the cold-start problem. But the item-based model can provide a solution. If a new customer wants to order in this restaurant, by following this source, similar items like the best Top-10 restaurant recommendations can be found.

Table 4: The industry algorithms approach with application comparison

Method	CB filtering	CF Demographic MF	Hybrid filtering	Neural CF	DeepFM
Wang et al. [26]	✗	✓	✗	✗	✗
Fu et al. [27]	✗	✗	✗	✗	✓
Karthik et al. [28]	✗	✓	✗	✓	✗
Poonam B. Thorat et al. [29]	✓	✓	✓	✗	✗
CHIA-YU LIN et al. [5]	✗	✓	✗	✗	✗

We also survey which paper using the recommendation algorithm to deploy in their system. At the same time, a comparison of the performance of various algorithms is also made shown in table 4. Be after the many practicality recommendation platforms that have developed their mechanism. The performance of diversity is also reflected there.

Chapter 5: Conclusions and future work

With the rapid development of smart devices and the exponential increase in the amount of data distributed on the internet, it is difficult for users to find the content they want through cuisine services restaurants, or by searching in huge shopping malls. So, the application of the recommendation system has rapidly developed to help users for convenient selection of preferred products. In many researchers' cases, the feedback user-items matrix that is using various algorithms for the development of recommendation services. In the work presented in this Thesis, the DFM model-based was used.

Recently, with the economic development and progress of society, interpersonal communication has become common. People that go together to a restaurant must have trust relationships, so a user-relationship recognition for recommendation is proposed, like friends, family, couples, professional, commercial, or unrelated at all. Although the accuracy of identifying the gender and age-group detection results of each person is not remarkably high, further analysis and improvement based on this feature are considered in future work.

The Top-10 restaurant means rating has shown the best performance. Therefore, in future research, we propose to try adjusting the precision and recall between, because the recall value is high it was means precision is not good on the contrary. By reducing the recall rate to a certain extent, the recommendation effect will also be significantly enhanced.

Reference

[1]. Guo, Huifeng, et al. "DeepFM: a factorization-machine based neural network for CTR prediction." *arXiv preprint arXiv:1703.04247* (2017).

[2]. Mallawaarachchi, Vijini, et al. "Change detection and notification of web pages: A survey." *ACM Computing Surveys (CSUR)* 53.1 (2020): 1-35.

[3]. Peng, Dongcheng, et al. "Research on information collection method of shipping job hunting based on web crawler." *2018 Eighth international conference on information science and technology (ICIST)*. IEEE, 2018.

[4]. Kausar, Md Abu, V. S. Dhaka, and Sanjeev Kumar Singh. "Web crawler: a review." *International Journal of Computer Applications* 63.2 (2013).

[5]. Lin, Chia-Yu, Li-Chun Wang, and Kun-Hung Tsai. "Hybrid real-time matrix factorization for implicit feedback recommendation systems." *IEEE Access* 6 (2018): 21369-21380.

[6]. Mahara, Tripti. "A new similarity measure based on mean measure of divergence for collaborative filtering in sparse environment." *Procedia Computer Science* 89 (2016): 450-456.

[7]. Masi, Iacopo, et al. "Deep face recognition: A survey." *2018 31st SIBGRAPI conference on graphics, patterns, and images (SIBGRAPI)*. IEEE, 2018.

[8]. LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." *nature* 521.7553 (2015): 436-444.

[9]. Mellouk, Wafa, and Wahida Handouzi. "Facial emotion recognition using deep learning: review and insights." *Procedia Computer Science* 175 (2020): 689-694.

[10]. Zhang, Shuai, et al. "Deep learning-based recommender system: A survey and new perspectives." *ACM Computing Surveys (CSUR)* 52.1 (2019): 1-38.

[11]. Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." *Computer* 42.8 (2009): 30-37.

[12]. Covington, Paul, Jay Adams, and Emre Sargin. "Deep neural networks for youtube recommendations." *Proceedings of the 10th ACM conference on recommender systems*. 2016.

[13]. Nagarnaik, Paritosh, and A. Thomas. "Survey on recommendation system methods." *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*. IEEE, 2015.

[14]. Wei, Jian, et al. "Collaborative filtering and deep learning-based recommendation system for cold start items." *Expert Systems with Applications* 69 (2017): 29-39.

[15]. He, Xiangnan, et al. "Neural collaborative filtering." *Proceedings of the 26th international conference on world wide web*. 2017.

[16]. Cheng, Heng-Tze, et al. "Wide & deep learning for recommender systems." *Proceedings of the 1st workshop on deep learning for recommender systems*. 2016.

[17]. Guo, Huifeng, et al. "DeepFM: a factorization-machine based neural network for CTR prediction." *arXiv preprint arXiv:1703.04247* (2017).

[18]. Zhao, Ruidong, Jishuai Li, and Sujuan Qin. "Weighted DeepFM: Modeling Multiple Features Interaction for Recommendation System." *2019 4th International Conference on Computational Intelligence and Applications (ICCIA)*. IEEE, 2019.

[19]. Kula, Maciej. "Metadata embeddings for user and item cold-start recommendations." *arXiv preprint arXiv:1507.08439* (2015).

[20]. https://www.pyimagesearch.com/2017/11/06/deep-learning-opencvs-blobfromimage-works/

[21]. Li, Junnan, et al. "Dual-glance model for deciphering social relationships." *Proceedings of the IEEE International Conference on Computer Vision*. 2017.

[22]. Guo, Huifeng, et al. "Deepfm: An end-to-end wide & deep learning framework for CTR prediction." *arXiv preprint arXiv:1804.04950* (2018).

[23]. Mallawaarachchi, Vijini, et al. "Change detection and notification of web pages: A survey." *ACM Computing Surveys (CSUR)* 53.1 (2020): 1-35.

[24]. Mallawaarachchi, V., Meegahapola, L., Alwis, R., Nimalarathna, E., Meedeniya, D., & Jayarathna, S. (2019). Change Detection and Notification of Webpages: A Survey. arXiv preprint arXiv:1901.02660.

[25]. Benkaddour, Fatima Zohra, et al. "An Adapted Approach for User Profiling in a Recommendation System: Application to Industrial Diagnosis." IJIMAI 5.3 (2018): 118-130.

[26]. Wang B, Ye F, Xu J. A personalized recommendation algorithm based on the user's implicit feedback in e-commerce[J]. Future Internet, 2018, 10(12): 117.

[27]. Fu W, Peng Z, Wang S, et al. Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33(01): 94-101.

[28]. Bhaskar K R K, Kundur D, Lawryshyn Y. Implicit Feedback Deep Collaborative Filtering Product Recommendation System[J]. arXiv preprint arXiv:2009.08950, 2020.

[29]. Thorat P B, Goudar R M, Barve S. Survey on collaborative filtering, content-based filtering and hybrid recommendation system[J]. International Journal of Computer Applications, 2015, 110(4): 31-36.