**Final report on online shopping product recommendation algorithms**

2017103580 Jeongwoon Kim, Sociology

Submitted: 17 June 20211.Design Background

Identifying users' interests and displaying advertisements that are relevant to them can increase the likelihood of purchase. In the past, product advertisements were exposed to users based on their demographic characteristics (gender, age, occupation, etc.). However, recent advances in computer specifications and calculation algorithms have made it possible to accumulate data on a large scale and build mathematical models based on it. In other words, it is possible to predict the interests of individuals using personal information such as search and purchase records as well as demographic characteristics. It is said that personalised advertisements based on this can increase the purchase rate of individuals by 20-50% (pabii blog, 2018.11.30). Therefore, many companies are currently building relevant databases and running targeted advertisements.

However, traditional personalised advertising has several limitations. Firstly, algorithmic performance is poor, showing ads that are not relevant to the user's interests. For example, existing targeting algorithms have difficulty distinguishing between searches for purchases and searches for hobbies. A prime example of this is Naver's adverts, which simply show ads based on blog posts rather than predicting your interests. In addition, the database required for personalised advertising is often not designed efficiently, as it stores unnecessary data other than the data used for analysis (pabii blog, 30.11.2018).

Therefore, this recommendation algorithm uses personal information such as personal search history and product viewing lists, as well as existing demographic characteristics, to predict what products the user is currently interested in and present ads related to them. There are three expected effects of such an algorithm. First, users get the information they need. For advertisers, the ads will be distributed more efficiently and will be more effective. Companies can increase sales by increasing the purchasing power of users and make appropriate decisions by reasonably predicting product demand.

2.Design Goals

1) Requirement analysis

(1) The personal information that users need to enter when registering is ID, password, mobile phone number, gender, and age.

(2) The user's search history and purchase history should be stored, and specific items can be added to the shopping cart. When storing information about search terms, the time of input shall also be recorded.

(3) Shopping mall members can be divided into buyers and sellers, and sellers can additionally enter product information.

(4) Products shall store the name, price, manufacturer, date of manufacture, rating, percentage of purchasers (gender, age), and product information.

(5) Overfitting should not occur when predicting user interests. Overfitting refers to a situation where the variance of the prediction is too large, resulting in poor accuracy.

2) Key considerations during the design process

(1) In many statistical analyses, matrix operations are frequently performed, and data preprocessing (modifying the values of the data for analysis) is often required.

(2) Set functions (mean, variance, covariance, etc.) are frequently used, and information about them should be stored if necessary. This way, when new data is entered, the existing set function only needs to be slightly modified to reflect it, i.e., the values can be updated without reapplying the set function to the entire data. For example, if the average for 100 records is 60, and a record named 69 is added, you can easily update the average to (60\*100+69)/101.

(3) Since there are many operations that produce specific values from records, it is necessary to check for null values and take appropriate action. Otherwise, null values will be contaminated.

(4) It should be possible to identify outliers in the records. It is also important to be able to distinguish between outliers caused by chance and those caused by changes in user interests.

(5) The application has frequent access to browsing history and shopping cart, but very little access to other attributes.

(6) Data related to cart items and search terms are frequently changed, but more changes are made to the latter than the former.

3)System environment to use

(1)DBMS: MySQL\* (2)Language to be used: Python (Numpy, Dassas)

(2) Language to be used: Python (libraries to be used are Numpy, Pandas)

(3) OS: Lubuntu (Ubuntu derivative)\*\* (4) IDEs: Jupyter Notebook, Jupyter Notebook, Jupyter Notebook

(4)IDE: Jupyter Notebook, DBeaver

\*MySQL requires fewer resources, so we wanted to improve its performance.

\*\*Lubuntu is one of the Linux with minimal features, which is why it responds quickly. Therefore, we wanted to improve the performance of the DBMS by using it.

4) Evaluation Criteria

(1) Is the user's search history, purchase history, and shopping cart items properly stored in the database?

(2) Is the user's shopping cart and search history used to identify their interests and serve relevant advertisements?

3.Design contents

1)엔티티(Entity)

|  |  |  |
| --- | --- | --- |
| 소비자(Customer) | 판매자(seller) | 상품(product) |
| ID  Password  Phone\_number  Gender  Age  Wishlist(multi-valued dependency)  Searchword(multi-valued dependency)  Buylog(multi-valued dependency) | ID  Password  Phone\_number  Gender  Age | Name  Selle\_ID(외래키 to seller)  Price  Release\_date  Company |

Wishlist, Searchword, and Buylog are attributes that store information about shopping carts, search terms, and purchase history, respectively, and are classified as multi-value attributes because a person can have multiple of them.

2)엔티티간 관계(Relations between Entity)

|  |
| --- |
| 검색어, 검색시간, 방문로그  구매자,상품명,판매자,가격  장바구니  M  M  N  N  구매일, 상품명, 판매자 가격, 배송주소  검색  판매  구매  1  N  N  M  User  Product  Seller |

3)논리적 스키마

(1)Customer: relation for managing customer information

|  |  |  |
| --- | --- | --- |
| 속성(Attritube) | 타입(Type) | 기타 |
| ID | char(16) unique, | primary key |
| Password | char(16) not null |  |
| Phone\_number | char(11) unique |  |
| Gender | char(1) |  |
| Age | tinyint |  |

Tinyint is an 8-bit integer type, and age cannot be more than 4 digits, so it is declared as an 8-bit integer for efficient resource management. Also, although the Customer and Seller releases have the same properties, they are managed separately because the reason for splitting the releases is to store the data in separate parts.Since the algorithm only uses Customer's data, separating the two releases can result in fewer searches than the two combined.

(2)Seller: relation for managing merchant information

|  |  |  |
| --- | --- | --- |
| 속성(Attritube) | 타입(Type) | 기타(Else) |
| ID | char(16) unique, | primary key |
| Password | char(16) not null |  |
| Phone\_number | char(11) unique |  |
| Gender | char(1) |  |
| Age | tinyint |  |

(3)Product: relation for managing selling product information

|  |  |  |
| --- | --- | --- |
| 속성(Attritube) | 타입(Type) | 기타(Else) |
| Name | varchar(255) not null | primary key |
| Seller\_ID | char(16) not null | primary key, foreign key(ON DELETE CASCADE) |
| Price | mediumint |  |
| Release\_date | datetime |  |
| Company | char(20) |  |
| Category | varchar(255) |  |

Since different sellers can sell the same product, Name alone is not enough to distinguish the data, so we took the foreign key Seller\_ID as the primary key to distinguish the data. Also, although varchar is a dynamic type, MySQL sets it to char(255), so the search speed is the same as static search.Although the data is not stored efficiently, we declared it as varchar for usability and convenience because the name and category can be of varying lengths. We also declared the type of Price to be mediumint, which is a large number with only positive numbers. Because prices are only positive, but there are many different price ranges, mediumint allows us to represent the largest possible number with the same amount of data, instead of the usual int.

(4)Wishlist: Relation to manage user-contained cart items

As with the Buylog and Searchword releases, implementing multi-attribute releases not only maintains data consistency to minimise duplication, but also makes it easier and faster for algorithms to get the data they need. The reason we set all attributes as primary and foreign keys is that there can be different sellers for a single item, and multiple consumers can add to cart for the same item (from the same seller), so we need to set all attributes as primary to ensure that the data is sufficiently distinct.

|  |  |  |
| --- | --- | --- |
| 속성(Attritube) | 타입(Type) | 기타(Else) |
| Product\_name | varchar(255) not null | primary key, foreign key( ON UPDATE CASCADE ON DELETE CASCADE,) |
| Customer\_id | char(16) not null | primary key, foreign key(ON DELETE CASCADE) |
| Seller\_id | char(16) not null | primary key, foreign key |

(5) Buylog: relation for managing your buying history

|  |  |  |
| --- | --- | --- |
| 속성(Attritube) | 타입(Type) | 기타(Else) |
| Product\_name | varchar(255) not null | primary key, foreign key( ON UPDATE CASCADE ON DELETE CASCADE,) |
| Customer\_id | char(16) not null | primary key, foreign key(ON DELETE CASCADE) |
| Seller\_id | char(16) not null | primary key, foreign key |
| buy\_time | DATETIME | DEFAULT CURRENT\_TIMESTAMP |

Wishlist와의 차이점은 구매일자를 데이터베이스가 자동적으로 저장한다는 것이다

(6)Searchword: relation for managing your search history

|  |  |  |
| --- | --- | --- |
| 속성(Attritube) | 타입(Type) | 기타(Else) |
| Word | varchar(100) not null | primary key |
| Customer\_id | char(16) not null | primary key, foreign key(ON DELETE CASCADE) |
| search\_time | DATETIME | primary key, foreign key  DEFAULT CURRENT\_TIMESTAMP |

We set Search\_time as the primary key because a particular consumer may search for a word multiple times, and we want to use time to differentiate between them. Also, purchase decisions happen days in advance, which can be seen through outliers in the time series. Therefore, by storing information about time, the algorithm can later analyse the time series.

4) Physical schema In MYSQL,

Indexes are set for all primary keys and a B+ tree structure is used.Therefore, the indexes of each table are ID (Customer and Seller), Product name, and Time (Search word only). The shopping mall recommendation algorithm uses only the data corresponding to the logged-in user's ID. Therefore, you need to be able to quickly find the data corresponding to the user's ID in the DB. Also, if you get a specific word in the analysis results, you need to advertise the corresponding products to the user. This means that you need to be able to quickly find products that contain a specific word.

As mentioned earlier, algorithms often perform time series analysis to determine a user's likelihood to buy, and often only use data from a specific time period. Therefore, an index on the property Searchword allows applications to quickly obtain data that corresponds to a specific time period.

However, SearchWord is frequently added and deleted, which causes overhead associated with updating the index. To avoid this, the application should create a local file for SearchWord to read and write data, and only store some data in the DB according to certain criteria.

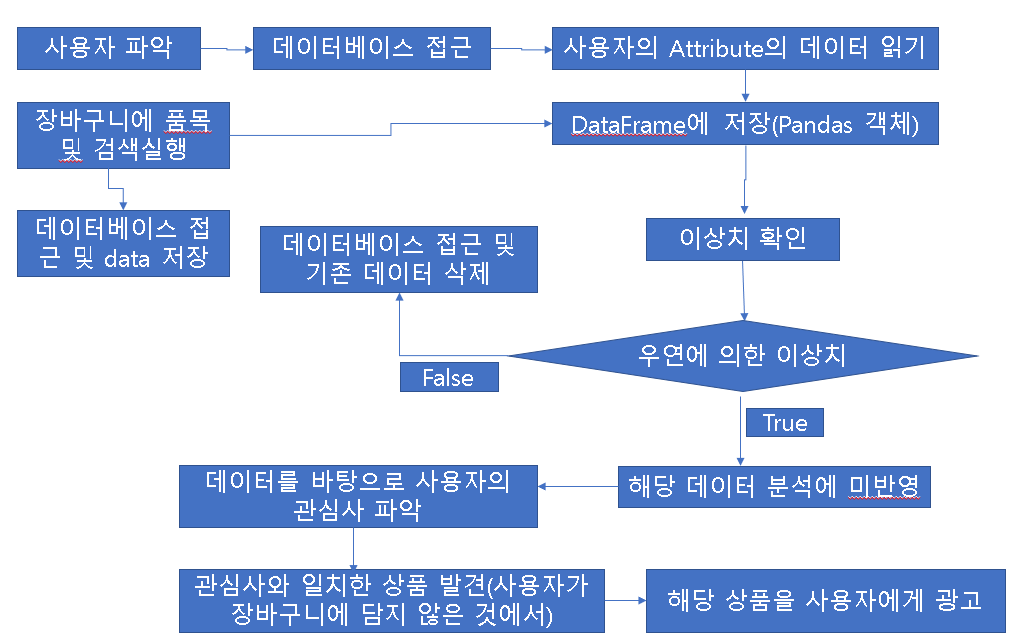
5) Application Design

(1) Function

: It statistically identifies the association between the user's search history, visit history, and shopping cart (wishlist). At this time, the stored time information is used to give high weight to products that have recently increased the user's interest.

: Recommend products that match the user's interests among the products they have not viewed based on the information obtained in step 21.

: When the recommendations based on the existing data are not statistically significant, i.e., when the user's interest in different products increases, the existing records (wishlist, searchword) stored in the database are deleted. At this time, data that is not related to the recently interested products is deleted, and purchase records for uninteresting products are not reflected in the recommendation algorithm.

(2) Application operation flow

(3) Test results

In our tests, all data input and output worked fine.