Experiment with intelligent recommendation techniques

1. Problem introduction and topic selection background：

With the rapid development of the Internet, the competition in the e-commerce industry is becoming increasingly fierce, and it is often difficult for users to quickly find products that truly meet their needs in the face of massive commodity information. Intelligent recommendation technology has emerged to provide users with a personalized and accurate commodity recommendation service by analyzing user behavior data and mining potential preferences, so as to improve user experience, increase user engagement and platform conversion rate, and become one of the key means for e-commerce enterprises to obtain competitive advantages. This paper focuses on the intelligent recommendation problem in the e-commerce scene, that is, how to accurately predict the product categories that users may be interested in and make personalized recommendations based on the user's historical behavior data (including purchase records and browsing behaviors). Specifically, the core problem is how to effectively mine the potential patterns and associations in user behavior data, and fully consider user multi-dimensional attributes (such as age, gender, income, region, etc.) and product characteristics to construct a recommendation model that can not only accurately reflect user preferences, but also adapt to the complexity of e-commerce environment.

2. Data introduction and preprocessing phase：

1. Introduction to the dataset：

Since the experiment we designed is intelligent push technology, I obtained the dataset from:https://www.kaggle.com/datasets/samps74/e-commerce-customer-behavior-dataset
In addition, I did some simple processing on the dataset. First, we expanded the data to 1000, expanded the categories of products to 6, and merged the ratings into the purchase records. It is easy to implement complex recommendation tasks, and solves the ambiguity that ratings and items are difficult to correspond。

2. Preprocessing of the dataset**:**

First of all, we need to deal with missing values. For annual income, user age and website stay time, we choose to directly take the median instead of them; for residence city and age, we take the mode instead of them; for abnormal values, we only deal with age. In addition, in order to facilitate data processing and suitable for model training, we discretize the data of Location, Annual Income and Time on Site.
Rule: Transform the income scale into four classes Assume that the income range is low, medium, high, and super high.
The classification of browsing time in the website is assumed to be divided into time intervals: short, medium, long, and very long.
The theory of regional division should be combined with specific geographical conditions, but the source data set is not given, so we choose random grouping to achieve.

3. Experimental methods：

1. User-based collaborative filtering：

(1) Technical Ideas：

We find similar users by their preferences for different items or content, and then we recommend the content that similar users are interested in but the current user has not found (waiting to be recommended by us) to the current user.

(2) key technology：

It calculates the similarity of users to find the K most similar neighbors of the current user. According to the preferences of neighboring users, it predicts the preferences of the current user, and then puts the items into a ranked list to recommend to the current user. The k-nearest neighbor algorithm is used here, and we implement our idea based on KNN.

(3) evaluation index：

Precision：

$$\text{Precision} = \frac{\sum_{u \in U} | R(u) \cap T(u) |}{\sum_{u \in U} | R(u) |}$$

Recall:

$$\text{Recall} = \frac{\sum_{u \in U} | R(u) \cap T(u) |}{\sum_{u \in U} | T(u) |}$$

F_1：

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where R(u) is the list of recommendations designed for user u according to the recommendation model, and T(u) is the list of recommendations of real interest on the actual test set.

(4) basic process：

We will create an item-user inverted list, then create a user interaction matrix based on the inverted list (the design of the matrix values will take into account various factors and will be based on our dataset), and then build a similarity matrix by choosing a metric (e.g., cosine similarity, Pearson similarity, Jacard similarity). Finally, the K most similar to the target user u are found from the matrix as the neighbors of u, and the items of the nearest users are extracted and the items that user u has purchased are removed as candidate candidates.
The interaction matrix elements are designed as follows:
If the user did not purchase an item and did not browse, we set it to 0 in the interaction matrix. If the user did not purchase an item, but did browse, then we designed the interaction matrix to be the browsing time (we have 4 levels, according to the level we have 0.5, 1,1.5, 2.5. If the user has purchased the product and there is a history of browsing we will add the user's rating of the item and the value of the history as the matrix element.
When the user makes a purchase:
Then we choose a similarity to measure the similarity relationship. By looking up information and combining my data, I choose cosine similarity as the index. Since the interaction matrix

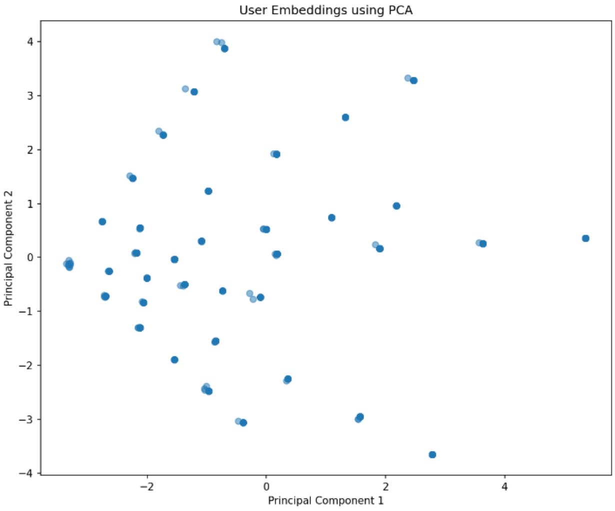
is high-dimensional (user × product category), cosine similarity performs well in high-dimensional space and can effectively capture the similarity of user preference direction. Cosine similarity focuses on the Angle between vectors rather than the length of vectors. This means that it will not be affected by some users having high or low ratings overall.
Here is the link to the reference articlehttps://www.cnblogs.com/chaosimple/p/3160839.html.
Finally, we need to split the dataset into a test set and a training set, so we have a validation set of 0.3 and a training set of 0.7. My idea for how to implement the validation process is to randomly (in my application, I chose 50 percent) leave some purchases in the validation set blank (i.e., no purchases) but not delete the browsing history information. Then, the training set is used to carry out the recommendation task and obtain the recommendation list. For the design of the recommendation list, I adopt the methods of counting, sorting and selecting Topn to optimize the recommendation list, verify it with the actual purchase situation of the validation set, and calculate the performance indicators.
Instead of considering the ratio between the validation and training sets, let's consider the value of K (how many users are we looking for as the nearest neighbors), the proportion of the validation set that excludes purchases, and our value of n in the Topn (how many recommendations are returned). These three parameters will affect the accuracy of our model. Here's what I got when I tested it for different values:

K	The validation set excludes the proportion of purchases	**Topn**	**Precision**	**Recall**	**F1 Score**
5	50%	3	0.3067	0.4801	0.3743
5	50%	5	0.1800	0.4153	0.2512
5	60%	3	0.1544	0.2083	0.1774
5	60%	5	0.1635	0.3833	0.2292
10	50%	3	0.2733	0.3989	0.3244
10	50%	5	0.1958	0.4842	0.2789
10	60%	3	0.1278	0.1799	0.1494
10	60%	5	0.1677	0.4091	0.2379
15	50%	3	0.218	0.3336	0.2643
15	50%	5	0.1930	0.4776	0.2749
15	60%	3	0.2356	0.3608	0.2850
15	60%	5	0.1667	0.4024	0.2358

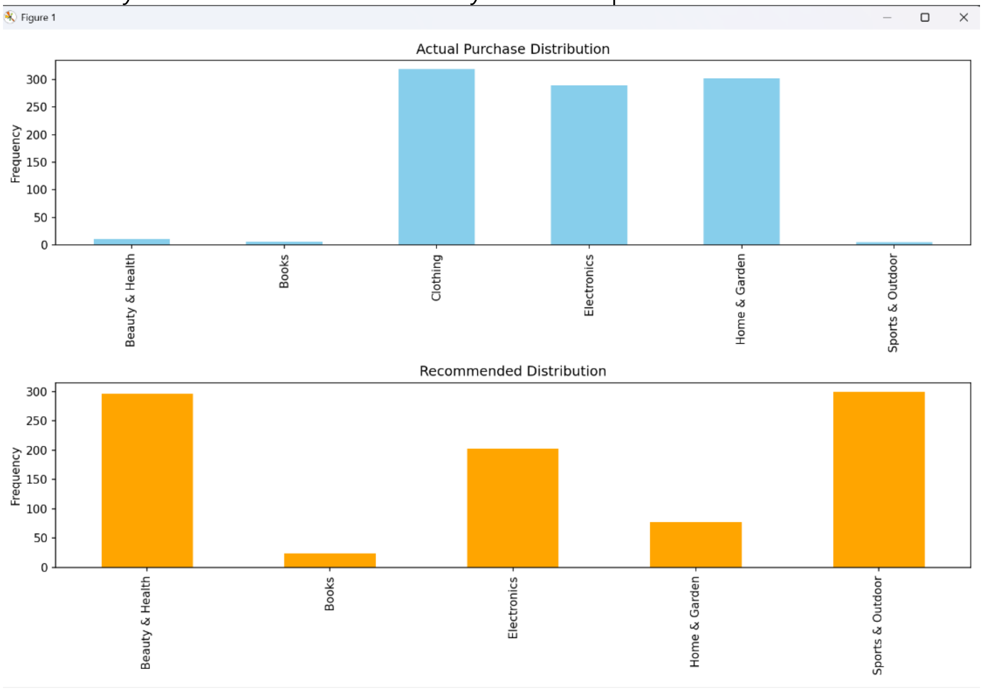
For the experimental results, we analyze: when K=5, the rejection ratio is 50%, and Topn=3, the F1 Score of the model is the highest, reaching 0.3743. This indicates that the model achieves the best balance between precision and recall with this combination of parameters. However, we still need to note that the experimental model seems to be not ideal. We consider that we only consider the purchase and browsing situation in the model, and we are fooling the user location, gender, income, age and other elements, so we need to take these unavoidable factors into account in the next experimental method.

(5) Visualization process：

Firstly, the user-item interaction matrix is visualized by PCA: by projecting the user-item interaction matrix into a two-dimensional space, the distribution of users in the feature space is displayed, which helps to understand the similarity structure between users.



Next design: Comparison plot of recommendation accuracy: The distribution of actual purchase categories and recommended categories are compared, which helps to evaluate the ability of the recommender system to predict the actual interests of users.



2. Association rule analysis：

(1) Technical Ideas：

It combines user attributes (such as gender, income level) with purchase/browsing behavior, constructs A mixed itemset containing attribute items and category items, and mines rules such as "attribute A + attribute B → purchase category C".

(2) key technology：

Firstly, the data is preprocessed: the transaction data set containing attributes is constructed. Each transaction contains "property item + purchase category item".
Frequent itemsets containing attributes and categories are mined using the Apriori

algorithm.

(3) performance criterion:

Support:

Let denote the proportion of transactions that contain both itemsets A and B in the total transactions and measure the importance of the rule.

$$\mathrm{Support}(A \to B) = \frac{|A \cap B|}{N}$$

Confidence:

This is the fraction of transactions with itemset A that also have itemset B. It measures the reliability of the rule (i.e., the probability of "if A, then B").

$$\mathrm{Confidence}(A \to B) = \frac{|A \cap B|}{|A|}$$

Lift:

Indicates the increase in the probability of B when A is included, and measures the actual value of the rule (excluding chance associations). If Lift = 1, then A and B are independent. If Lift>1, A and B are positively correlated; A Lift<1 indicates a negative correlation.

$$\mathrm{Lift}(A \to B) = \frac{\mathrm{Confidence}\ (A \to B)}{\mathrm{Support}\ (B)} = \frac{|A \cap B|\ /\ |A|}{|B|\ /\ N} = \frac{|A \cap B| \cdot N}{|A| \cdot |B|}$$

(4) basic process:

1. Extract user attributes and convert them to "item" numeric attributes (age, income) : Bin age: <30, 30-50, >50 annual income: Low (< 50,000), Medium (50-80,000), High (80-120,000), Very High (> 120,000) (corresponding to Income Tier in the data) Type attribute: directly as an item Gender: Male, Female Region: Region 1-8 Time on Site Tier: Short/Medium/Long/Very Long (keep original tier).

Step 2 Generate mixed itemsets (attribute + category) Use the Apriori algorithm to find frequent itemsets that contain both attributes and categories, e.g., frequent 3-itemsets: {Male, Low Income, Home & Garden} (support = number of low-income male users buying Home & Garden/total number of users)

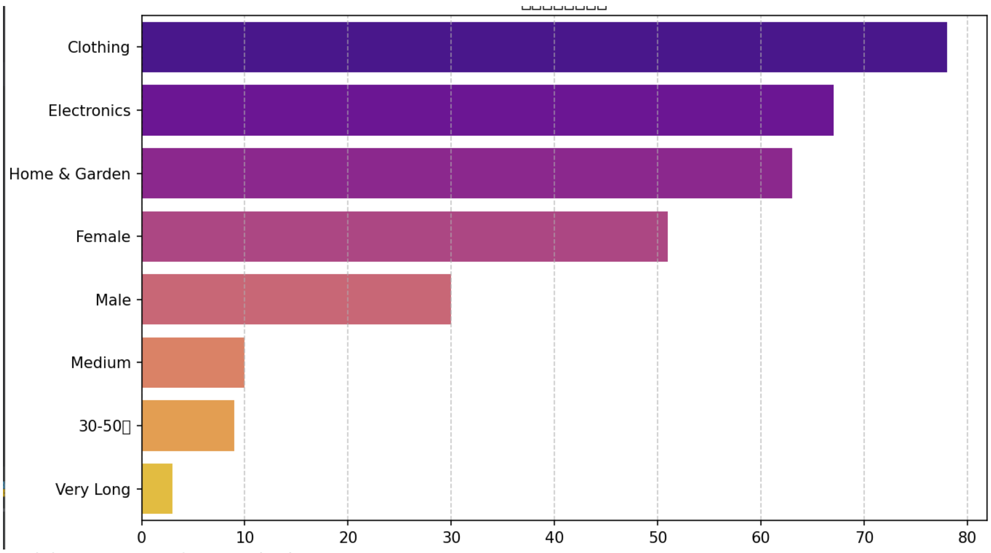
The former includes attributes, and the latter is the form of category rules: Attribute → category: such as Low Income → Electronics (low-income users buy electronic products) attribute + purchased category → Not purchased category: For example, Male ∧ Clothing → Home & Garden (Male users who have bought clothing may buy home & garden products)

Step 3 Rule screening criteria Boost > 1: Attribute + purchased category has a positive impact on the purchase of the target category Confidence ⩾ 50% : rule reliability aftereffect is the category not purchased by the user. Get the ruleset.

In the fourth step, we randomly create a new data set to verify the precision and recall of the rule set.

(5) Visualization process:

Recommend a product category distribution: in the practical application on the basis of high frequency words we can simple recommendations, such as short video hot list is recommended. Statistics after all valid rules in a (recommended product category) occurrences identify high-frequency recommended commodity categories, supporting business decisions (such as inventory management, marketing emphasis)



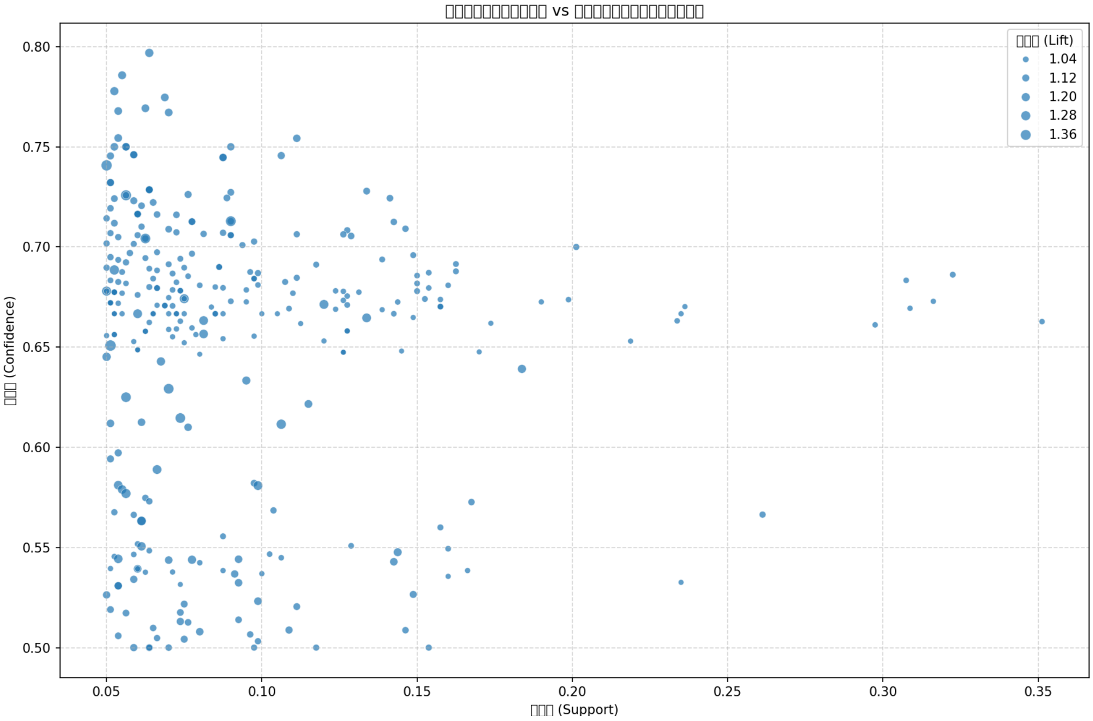
Bubble plots of association rules

Horizontal axis: Support, which is how often a rule appears in the training dataset.

On the vertical axis: Confidence, which is the probability that the rule's successor occurs when its predecessor occurs.

Bubble size: Lift, indicating the validity of the rule (>1 indicates a positive correlation.)

Purpose: Quickly identify high-quality rules with high support, high confidence and high promotion.



4. Summarize:

The experimental methods mainly cover two aspects: user-based collaborative filtering and association rule analysis.

Firstly, in the user-based collaborative filtering experiment, by constructing the item-user inverted list and the user interaction matrix, the cosine similarity is used to calculate the user similarity, and then the k neighboring users most similar to the target user are screened out, and the recommendation list is generated for the current user based on the purchase behavior of these neighboring users. The experimental evaluation indexes include precision, recall and F1 Score. After the combination test of different K values, the rejection ratio of the validation set and the Topn value, it is found that when K = 5, the rejection ratio is 50%, and the Topn value is 3, the F1 score reaches the highest value of 0.3743. It shows that the model achieves a better balance between precision and recall under the parameter combination, but the overall model effect still has room for improvement, because the user location, gender, income, age and other multi-dimensional factors are not fully considered.

The association rule analysis experiment focuses on the combination of user attributes and purchase/browsing behavior, constructs A mixed itemset containing attributes and categories, uses Apriori algorithm to mine frequent itemsets, and generates rules such as "attribute A + attribute B → purchase category C". By setting the screening criteria such as boost > 1 and confidence ⩾ 50%, the rules with high practical value are obtained. Through the visualization of the rules, such as the distribution statistics of the recommended product categories and the bubble chart of association rules, the association patterns between different product categories and the effectiveness of the rules are clearly presented.

User-based collaborative filtering and association rule analysis have different focuses. The former focuses on the similarity of user behavior, while the latter focuses on the correlation between user attributes and product categories. The two complement each other, providing a diversified technical perspective and practical method for e-commerce enterprises to implement personalized recommendation。