

推荐算法

丁兆云

内容

- **1**推荐系统模型
- **2**基于内容的推荐
- **3**协同过滤
- **4**评估及实际问题

1推荐系统模型

什么是推荐系统

- 软件/工具/技术
 - 为用户提供**选择项目**的决策支持
 - 购买商品、听歌、新闻
- 大多数推荐系统专注于一种类型的项目
 - **Amazon/当当**
 - **Expedia/携程**
 - **IMDB/豆瓣**

您可能感兴趣的商品



模式识别中的核方法及其应用... ×

¥21.90 ~~¥28.00~~
★★★★★ (33条评论)



机器人学导论 (原书第3版)... ×

¥33.60 ~~¥42.00~~
★★★★★ (268条评论)



数据挖掘实用机器学习技术 (原书第2版)... ×

¥38.40 ~~¥48.00~~
★★★★★ (283条评论)



编程之美——微软技术面试心得 (勤练算法功... ×

¥36.50 ~~¥40.00~~



人工智能 ×

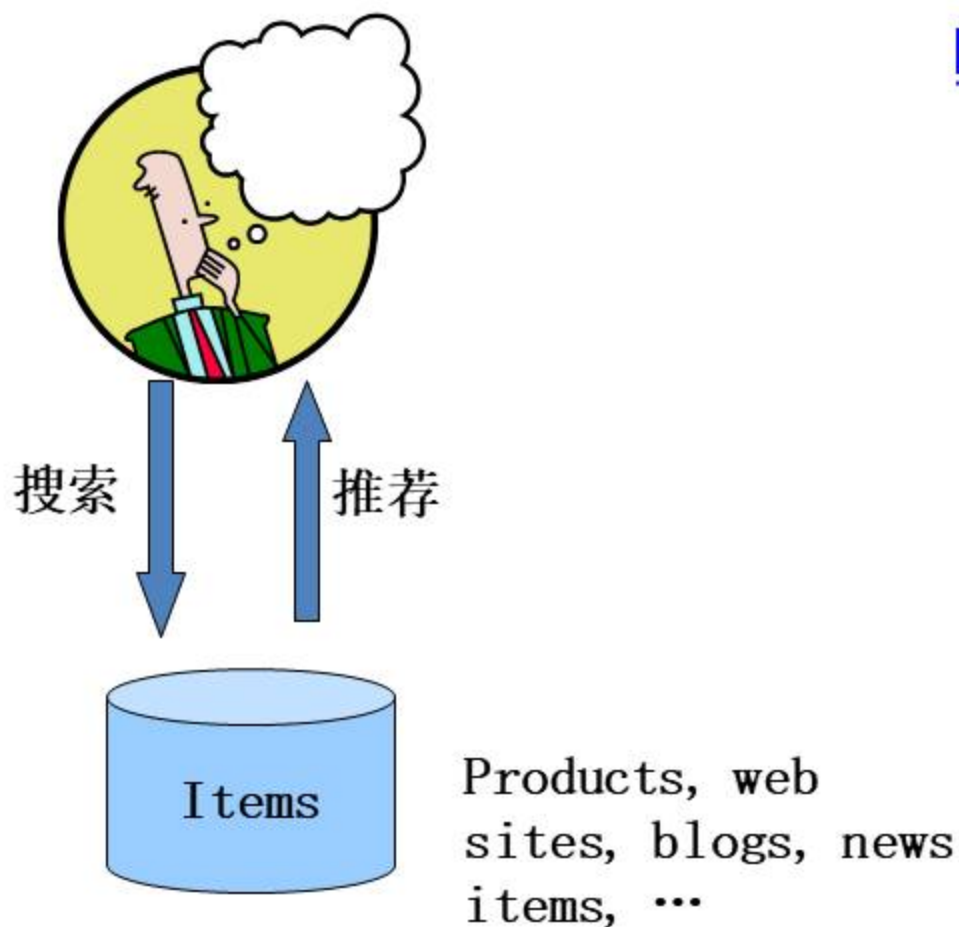
¥24.50 ~~¥30.00~~



数据挖掘：实用机器学习技术及Java实现... ×

¥30.00 ~~¥40.00~~

1.1 推荐系统的特点



Examples:

amazon.com.



StumbleUpon



del.icio.us



movielens
helping you find the *right* movies

last.fm
the social music revolution

Google
News

YouTube

XBOX
LIVE

1.1推荐系统的特点

	推荐系统	信息检索	数据库
用户要求	主动发现用户兴趣	用户使用“自然语言”检索	用户提供格式化的查询
数据要求	系统收集数据	网页等用户产生数据	输入数据
任务要求	(有用) 的项目	最相关的项目	所有精确符合的项目
目标要求	快、准、好	快、准	快

1.2 推荐系统严格模型

- **X** 用户集
- **S** 项目集
- 效用矩阵 **Utility Matrix**
 - 效用函数 **Utility function** u :

	Avatar	LOTR	Matrix	Pirates
Alice	1		0.8	
Bob		0.5		0.3
Carol	0.9		1	0.8
David			1	0.4

- $u: X \times S \rightarrow R$
- **R** 评分集, 完全有序集
- 例如, 0-5 星, $[0,1]$ 之间的实数

1.3推荐系统关键问题

1. 收集已知评分形成 R 矩阵
 - 如何收集效用矩阵中的数据
2. 根据已知的评分推断未知的评分
 - 主要对未知的高评分感兴趣，只关心用户喜欢什么
3. 评估推断方法
 - 如何衡量推荐方法的性能

1.3评分的收集

- 显式评价
 - 要求用户对项目给出评分
 - 实际中不太可行—困扰用户
- 隐式评价
 - 从用户的行为中学习其评分
 - e.g., 购买意味着高评分
 - 什么代表低评分呢?

1.3效用的推断

- 关键问题: 效用矩阵 U 稀疏
 - 大多数人没有评价过大多数项目
 - 冷启动
 - 新的项目没有评分
 - 新的用户没有历史

	Avatar	LOTR	Matrix	Pirates
Alice	1		0.8	
Bob		0.5		0.3
Carol	0.9		1	0.8
David			1	0.4

- 3种方法
 - 基于内容 **Content-based**
 - 协同过滤 **Collaborative Filtering**
 - 基于潜在因素（隐变量） **Latent factor based**

2基于内容的推荐系统

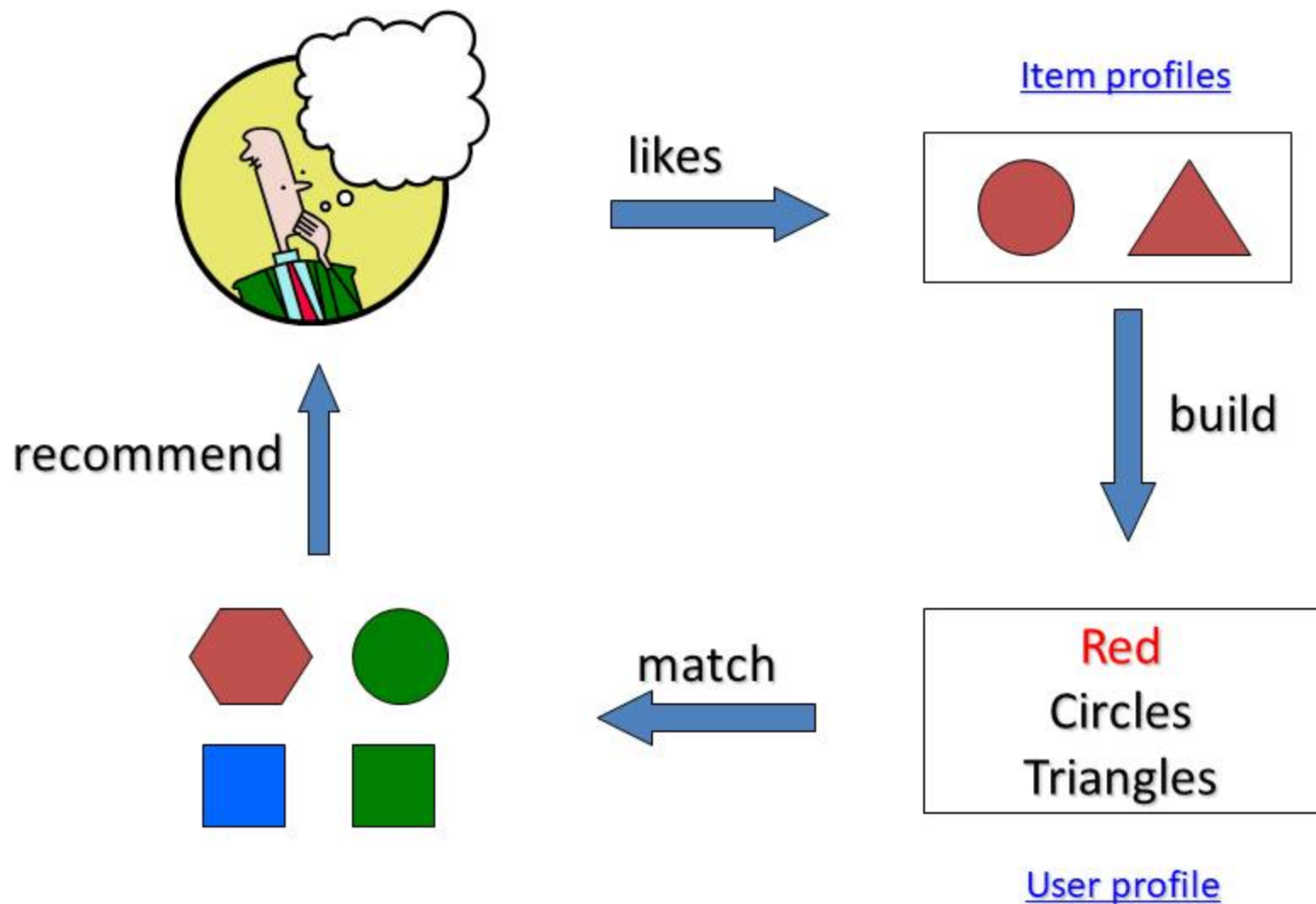
2.1 基于内容的推荐

- 主要思想: 向用户 **c** 推荐与她评分高（喜欢）项目 **相类似的项目**

例子:

- 电影推荐
 - 推荐相同演员、导演、流派 ...
- **Websites, blogs, news**
 - 推荐类似内容的网页

2.2 推荐的过程



2.4项模型 item profile

- 对每个项目建立一份 **item profile**
- **Profile** 是特征 **features** 的集合
 - movies: author, title, actor, director,...
 - text: set of “important” words in document
- 文本特征——关键词
 - 常用的启发式方法是 **TF.IDF** (Term Frequency times Inverse Doc Frequency)
- 非文本项目特征——困难
 - 邀请用户进行标记 **Tag** (词语、短语)

- **Tiananmen square**



- **Sunset at Malibu**



2.5Recap: TF.IDF

f_{ij} 文档 j 中词项 i 出现的频次

$$TF_{ij} = \frac{f_{ij}}{\max_k f_{kj}}$$

Note: we normalize TF
to discount for “longer”
documents

n_i = 包含词项 i 的文档数

N = 文档数

$$IDF_i = \log \frac{N}{n_i}$$

TF.IDF 分值

$$w_{ij} = TF_{ij} \times IDF_i$$

Doc profile = 有最高 TF.IDF 值的词汇及其对应分数的集合

2.6 用户模型 User profiles

- **User profile:**
 - 反映用户的特征偏好
 - 根据项模型统计
 - 用户评过项目的项目 **profile** 加权平均
- 启发式预测
 - 给定用户模型 \mathbf{x} , 项目模型 \mathbf{i} , 估计用户 \mathbf{x} 对于项目 \mathbf{i} 的效用值

$$u(\mathbf{x}, \mathbf{i}) = \cos(\mathbf{x}, \mathbf{i}) = \frac{\mathbf{x} \cdot \mathbf{i}}{||\mathbf{x}|| \cdot ||\mathbf{i}||}$$

2.7 基于内容方法的优点

- 不需要其他用户的数据
 - 没有冷启动或者稀疏性的问题
- 能给品味一致的用户推荐
- 能给新项目或不流行项目推荐
 - 没有第一个评价者的问题
- 能够提供解释
 - 可以对推荐项目给出对应的内容特征描述

$$u(x, i) = \cos(x, i) = \frac{x \cdot i}{||x|| \cdot ||i||}$$

2.8 基于内容方法的缺点

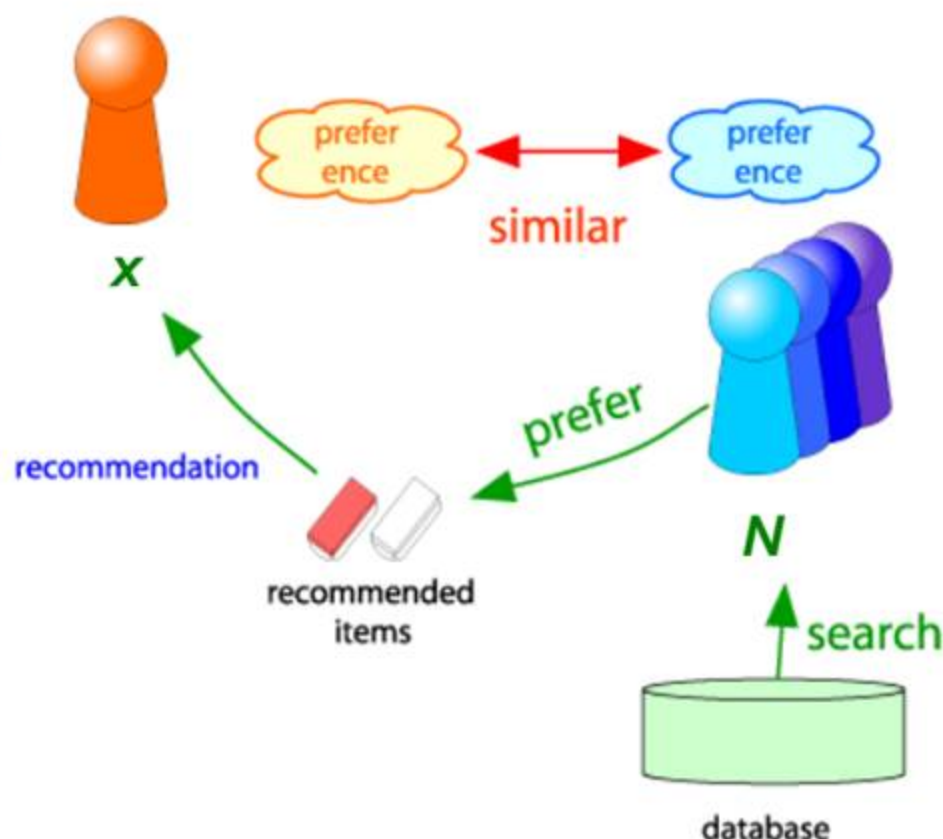
- 找到适当的特征是困难的
 - e.g., images, movies, music
- 过度集中
 - 不会推荐用户内容偏好模型之外的项目
 - 人们可能有多方面的兴趣
 - 不能利用其它用户的优质判断
- 对新用户的推荐
 - 如何给新用户建立模型?

3 协同过滤

COLLABORATIVE FILTERING

3.1 协同过滤

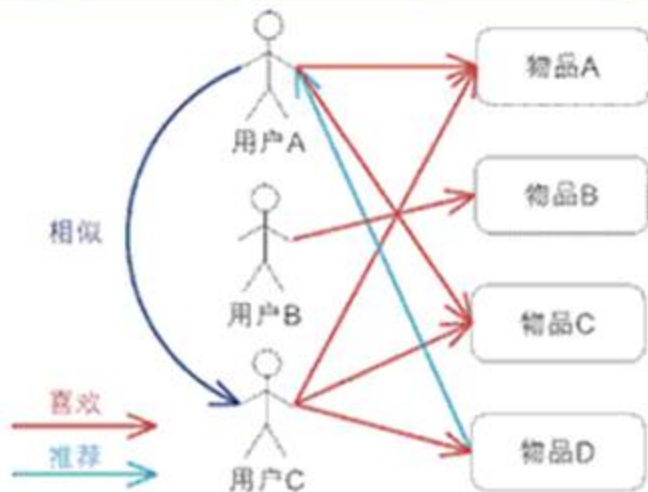
- 考虑用户 x
 - 找到与 x 有相似评分的用户集合 N
 - 根据 N 中用户的评分估计 x 的评分



3.2 基于用户的协同过滤

- 对于用户 **A**，根据用户的历史偏好，这里只计算得到一个邻居 - 用户 **C**，然后将用户 **C** 喜欢的物品 **D** 推荐给用户 **A**。

用户/物品	物品A	物品B	物品C	物品D
用户A	√		√	推荐
用户B		√		
用户C	√		√	√



3.2相似的用户

$$\begin{aligned}r_x &= [*, _, _, *, ***] \\ r_y &= [*, _, **, **, _]\end{aligned}$$

- 令 r_x 为用户 x 的评分矢量
- **Jaccard** 相似度
 - 问题：忽略了评分的分值
- 余弦相似度 **Cosine similarity measure**

r_x, r_y as sets:

$$r_x = \{1, 4, 5\}$$

$$r_y = \{1, 3, 4\}$$

$$\text{sim}(x, y) = \cos(r_x, r_y) = \frac{r_x \cdot r_y}{\|r_x\| \cdot \|r_y\|}$$

r_x, r_y as points:

$$r_x = \{1, 0, 0, 1, 3\}$$

$$r_y = \{1, 0, 2, 2, 0\}$$

- 问题：将缺失项目视为“否定”
- 皮尔森相关系数 **Pearson correlation coefficient**
 - S_{xy} = 用户 x 和用户 y 共同评价过的项目集合

$\bar{r}_x, \bar{r}_y \dots$ avg.
rating of x, y

$$\text{sim}(x, y) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)(r_{ys} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)^2 (r_{ys} - \bar{r}_y)^2}}$$

3.2 缺失 = 否定？

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

- 直觉: $\text{sim}(A, B) > \text{sim}(A, C)$, 但是
 - Jaccard similarity: $1/5 < 2/4$
 - Cosine similarity: $0.386 > 0.322$ (接近)
- 原因: 将缺失分量视为“否定” (取0值, 意味最低评价)
- 解决措施: 减去(行)均值 —— 中心化

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	2/3			5/3	-7/3		
B	1/3	1/3	-2/3				
C				-5/3	1/3	4/3	
D		0					0

sim A,B vs. A,C:
0.092 > -0.559

注意: cosine sim.
在以零为中心时,
就是相关系数

3.2 评分预测

- r_x : 为用户 x 的评分矢量
- N : 为对项目 i 的评分与用户 x 最相似的 k 个用户的集合
- 用户 x 对项目 s 的评分预测

$$r_{xi} = \frac{1}{k} \sum_{y \in N} r_{yi}$$

$$r_{xi} = \frac{\sum_{y \in N} s_{xy} \cdot r_{yi}}{\sum_{y \in N} s_{xy}}$$

$$s_{xy} = \text{sim}(x, y)$$

– 其他方法?

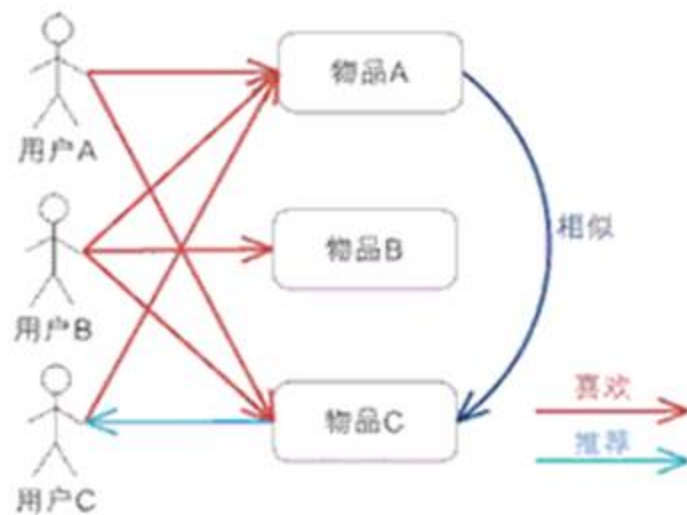
3.3基于项目的协同过滤 Item-Item CF

- 除了 **user-user**, 有另一个角度: **item-item**
 - 对项目*i*, 寻找其他相似的项目
 - 根据相似项目的评分估计项目*i*的评分
 - 可以采用类似 **user-user model**的相似度测度

3.3 基于项目的协同过滤 Item-Item CF

- 对于物品 A，根据所有用户的历史偏好，喜欢物品 A 的用户都喜欢物品 C，得出物品 A 和物品 C 比较相似，而用户 C 喜欢物品 A，那么可以推断出用户 C 可能也喜欢物品 C。

用户/物品	物品A	物品B	物品C
用户A	√		√
用户B	√	√	√
用户C	√		推荐



3.3 基于项目的协同过滤 Item-Item CF

- 除了 **user-user**, 有另一个角度: **item-item**
 - 对项目 i , 寻找其他相似的项目
 - 根据相似项目的评分估计项目 i 的评分
 - 可以采用类似 **user-user model** 的相似度测度

$$r_{xi} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{xj}}{\sum_{j \in N(i;x)} s_{ij}}$$

s_{ij} ... similarity of items i and j

r_{xj} ... rating of user u on item j



$N(i;x)$... set items rated by x similar to i

Item-Item CF ($|N|=2$)

users

movies

	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3			5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

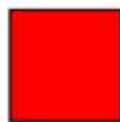
 - unknown rating  - rating between 1 to 5

Item-Item CF ($|N|=2$)

users

	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

movies



- estimate rating of movie 1 by user 5

Item-Item CF ($|N|=2$)

		users												sim(1,m)
		1	2	3	4	5	6	7	8	9	10	11	12	
movies	1	1		3		?	5			5		4		1.00
	2			5	4			4			2	1	3	-0.18
	<u>3</u>	2	4		1	2		3		4	3	5		<u>0.41</u>
	4		2	4		5			4			2		-0.10
	5			4	3	4	2					2	5	-0.31
	<u>6</u>	1		3		3			2			4		<u>0.59</u>

Neighbor selection:

Identify movies similar to movie 1, rated by user 5

Here we use Pearson correlation as similarity:

1) Subtract mean rating m_i from each movie i

$$m_1 = (1+3+5+5+4)/5 = 3.6$$

row 1: $[-2.6, 0, -0.6, 0, 0, 1.4, 0, 0, 1.4, 0, 0.4, 0]$

2) Compute cosine similarities between rows

Item-Item CF ($|N|=2$)

		users												
		1	2	3	4	5	6	7	8	9	10	11	12	$\text{sim}(1,m)$
movies	1	1		3		?	5			5		4		1.00
	2			5	4			4			2	1	3	-0.18
	<u>3</u>	2	4		1	2		3		4	3	5		<u>0.41</u>
	4		2	4		5			4			2		-0.10
	5			4	3	4	2					2	5	-0.31
	<u>6</u>	1		3		3			2			4		<u>0.59</u>

Compute similarity weights:

$$s_{13}=0.41, s_{16}=0.59$$

Item-Item CF ($|N|=2$)

		Users											
		1	2	3	4	5	6	7	8	9	10	11	12
movies	1	1		3		2.6	5			5		4	
	2			5	4			4			2	1	3
	<u>3</u>	2	4		1	2		3		4	3	5	
	4		2	4		5			4			2	
	5			4	3	4	2					2	5
	<u>6</u>	1		3		3			2			4	

Predict by taking weighted average:

$$r_{15} = (0.41 \cdot 2 + 0.59 \cdot 3) / (0.41 + 0.59) = 2.6$$

$$r_{ix} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{jx}}{\sum s_{ij}}$$

3.3CF: 基本操作

- 定义项目*i*和*j*的相似度 s_{ij}
- 选择*k*个最近邻居 $N(i;x)$
 - 用户*x*评价过的最类似*i*的项目
- 以加权平均估计评分 r_{xi}

Before:

$$r_{xi} = \frac{\sum_{j \in N(i;x)} s_{ij} r_{xj}}{\sum_{j \in N(i;x)} s_{ij}}$$

$$r_{xi} = b_{xi} + \frac{\sum_{j \in N(i;x)} s_{ij} \cdot (r_{xj} - b_{xj})}{\sum_{j \in N(i;x)} s_{ij}}$$

baseline estimate for r_{xi}

$$b_{xi} = \mu + b_x + b_i$$

- μ = overall mean movie rating
- b_x = rating deviation of user x
= (avg. rating of user x) - μ
- b_i = rating deviation of movie i

3.3 Item-Item vs User-User

	Avatar	LOTR	Matrix	Pirates
Alice	1		0.8	
Bob		0.5		0.3
Carol	0.9		1	0.8
David			1	0.4

- 实际中，**item-item** 比 **user-user** 的效果好
- 原因？ **Item** 更简单，**user** 往往有多重品味

3.4CF的优缺点

- 适合于任何item
 - 不需要特征选择
- Cold Start:
 - 需要系统中有足够的用户进行匹配
- 稀疏性:
 - ratings 矩阵稀疏，难以发现评价过相同项目的用户
- 第一个评价者
 - 无法推荐一个没有被评价过的项目，新项目, 隐秘项目
- 流行度偏差
 - 无法给只有单一口味的用户推荐项目
 - 倾向于推荐流行项目

3.5混合方法

- 实现两种或多种不同的推荐方法，并组合预测结果
 - 比如用线性组合
- 将基于内容的方法与**CF**相结合
 - 建立**item profile** 解决新**item**问题
 - 利用人口统计信息解决新用户问题

4评估及实际问题

Evaluation

Diagram illustrating a user-movie rating matrix. The vertical axis is labeled **users** and the horizontal axis is labeled **movies**.

1	3	4			
	3	5			5
		4	5		5
		3			
		3			
2			2		2
				5	
	2	1			1
	3			3	
1					

Evaluation

Diagram illustrating a user-movie rating matrix for evaluation. The matrix is labeled "users" (vertical axis) and "movies" (horizontal axis).

The matrix is divided into two sections:

- Training Data (Yellow cells):** Contains known ratings.
- Test Data Set (Gray cells):** Contains unknown ratings, indicated by question marks.

The matrix structure is as follows:

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6
User 1	1	3	4			
User 2		3	5			5
User 3			4	5		5
User 4			3			
User 5			3			
User 6	2			?		?
User 7					?	
User 8		2	1			?
User 9		3			?	
User 10	1					

4.1 评估预测性能

- 对比预测值与已知的评分
 - Root-mean-square error (RMSE)
 - Precision at top 10
 - Rank correlation
- 另一种方法: **0/1 model**
 - 覆盖度
 - 系统能够预测的items/users 数量
 - 精确度
 - 预测的精度

4.2 错误测度的问题

- 有时狭隘地关注精度没有意义
 - **Prediction Diversity** 预测多样性
 - **Prediction Context** 预测情境
 - **Order of predictions** 预测顺序

总结

- 效用矩阵 **Rating**
- 推荐系统
 - **content based** : 项目模型; 用户模型; 相似度
 - **CF**: 相似度; **User-based CF**; **Item-based CF**; knn
- 推荐性能评估
 - **RSME, 0/1 model**; 精度及其他

数据挖掘竞赛案例1

<地点推荐系统>



竞赛背景

移动数据

基于地点推荐技术

熟悉周遭环境

提升地点的影响力

参赛数据

地图信息

A	B	C	D	E	F	G
地点ID	纬度	经度	所在城市	粗类别	细类别	
107780	22.29743	114.1726	overseas	公共机构	公寓/小区/里弄	
70990	31.13684	121.4226	shanghai			
132379	31.23218	121.3976	shanghai			
38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆	
104522	27.7	85.33333	overseas			
91784	13.73705	100.5604	overseas	交通/住宿	地铁站/轻轨站	
97543	34.52023	112.9788	zhengzhou			
2996	31.18514	121.428	shanghai	商店/生活	时尚服饰	
96184	31.17635	121.5073	shanghai			
33986	30.71455	121.3366	shanghai			
84982	1.29695	103.8523	overseas	学校/教育	大学/研究所/专科院	
41797	31.27782	121.3654	shanghai			
60801	31.1731	121.4908	shanghai			

用户信息

	A	B	C	D
1	用户ID	地点ID	前往次数	
2	7263	112417	1	
3	7263	112416	1	
4	7262	112413	1	
5	7262	112412	1	
6	7262	112411	1	
7	7262	112410	1	
8	7261	112408	1	
9	7261	112407	1	



参赛要求

就训练集数据中的每一位用户，
各推荐50个不同的用户感兴趣的地点。

评分标准

平均截断召回率

$$\text{Recall} = \frac{1}{M} \sum_u \frac{|V_u \cap S_u|}{V_u}$$



协同过滤算法

一、基于用户的协同过滤算法

二、基于物品的协同过滤算法

实验过程

1

数据预处理

用户信息

	A	B	C	D
1	用户ID	地点ID	前往次数	
2	7263	112417	1	
3	7263	112416	1	
4	7262	112418	1	
5	7262	112412	1	
6	7262	112411	1	
7	7262	112410	1	
8	7261	112408	1	
9	7261	112407	1	

地图信息

根据经纬度聚类，将连续数据离散化

A	B	C	D	E	F	G
地点ID	纬度	经度	所在城市	粗类别	细类别	
107780	22.29743	114.1726	overseas	公共机构	公寓/小区/里弄	
70990	31.13684	121.4226	shanghai			
132379	31.23218	121.3976	shanghai			
38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆	
104522	27.7	85.33333	overseas			
91784	13.73705	100.5604	overseas	交通/住宿	地铁站/轻轨站	
97543	34.52023	112.9788	zhengzhou			
2996	31.18514	121.428	shanghai	商店/生活	时尚服饰	
96184	31.17635	121.5073	shanghai			
33986	30.71455	121.3366	shanghai			
84982	1.29695	103.8523	overseas	学校/教育	大学/研究所/专科院	
41797	31.27782	121.3654	shanghai			
60801	31.1731	121.4908	shanghai			

实验过程

1 数据预处理

用户ID	地点ID	纬度	经度	地点	粗类别	细类别
592	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆
2761	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆
4266	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆
4608	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆
6598	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆
7531	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆
13255	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆
13693	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆
17482	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆
23743	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆
25023	38132	34.26635	117.1878	xuzhou	餐饮	咖啡馆



实验过程

2

计算相关度

皮尔逊相关系数

(Pearson correlation coefficient)

$$r(X, Y) = \frac{Cov(X, Y)}{\sqrt{Var[X] Var[Y]}}$$

$Cov(X, Y)$ 为X与Y的协方差
 $Var[X]$ 为X的方差, $Var[Y]$ 为Y的方差

实验过程

3 设置参数

```
class recommender:
```

```
    # data: 数据集, 这里指users
```

```
    # k: 表示得出最相近的k的近邻
```

```
    # metric: 表示使用计算相似度的方法
```

```
    # n: 表示推荐place的个数
```

```
    def __init__(self, data, k=10, metric='pearson', n=50):
```

```
        #数据集data (用user), pearson矩阵, 推荐数为i+1, 最近邻为3
```

```
        #如果推荐数少于50, 可能是邻居数不够
```

```
        self.k = k
```

```
        self.n = n
```

```
        self.username2id = {}
```

```
        self.userid2name = {}
```

```
        self.productid2name = {}
```

```
# 推荐算法的主体函数
```

```
def recommend(self, user_id):
```

```
    # 定义一个字典, 用来存储推荐的地点和分数
```

```
    recommendations = {}
```

```
    # 计算出user与其他所有用户的相似度, 返回一个list
```

```
    nearest = self.computeNearestNeighbor(user_id)
```

```
    # print nearest
```

```
    userRatings = self.data[user_id] # 打分=数据中的userid
```

```
    # print userRatings
```

```
    totalDistance = 0.0
```

```
    # 得住最近的k个近邻的总距离
```

```
    for i in range(self.k):
```

```
        totalDistance += nearest[i][1]
```

```
    if totalDistance == 0.0:
```

```
        totalDistance = 1.0
```

设置K近邻的相关参数

对相似度进行排序计算

实验过程

4 输出结果

id= 22242

placeid_list: ['1354', '800', '14975', '55000', '17763', '1367', '57514', '72703', '120387', '10155', '29

near_list: [('6231', 1.0000000000000475), ('9673', 1.0000000000000002), ('37061', 1.0000000000000002), ('

id= 24848

placeid_list: ['4878', '11787', '36328', '36332', '116644', '42053', '510', '55444', '52522', '55883', '4









near_list: [('48227', 1.0000000000000002), ('47316', 1.0), ('42469', 0.9999999999999999), ('30514', 0.99

id= 26802

placeid_list: ['4418', '3151', '1478', '11445', '25095', '30981', '1498', '20279', '1529', '2242', '525',

near_list: [('8147', 1.0000000002665401), ('20203', 1.0000000002665401), ('20444', 1.0000000002665401),

实验总结

4	-		lhtlovewzx	0	1	2016-11-26 21:50
5	-		148	0	2	2016-10-16 10:03
6	-		rw_personal	0	3	2016-10-22 13:11
7	-		huaming	0.00026	1	2016-11-22 14:30
8	-		beautiful	0.00082	1	2016-11-08 23:06
9	-		testmm	0.00092	1	2016-11-19 20:19
10	-		NUDT丁兆云DM	0.00166	9	2017-11-27 06:34
11	-		yshbjut	0.04306	6	2016-11-07 18:17

数据挖掘竞赛案例2

<重复购买预测>

01 赛题介绍

03 数据处理

05 模型训练

02 数据描述

04 特征提取

06 模型结果

商家有时会在特定日期（例如“Boxing-day”，“黑色星期五”或“双11”）进行大促销（例如折扣或现金券），以吸引大量新买家。许多吸引的买家都是一次性交易猎人，这些促销可能对销售产生很小的长期影响。为了缓解这个问题，商家必须确定谁可以转换为重复买家。通过瞄准这些潜力忠诚的客户，商家可以大大降低促销成本，提高投资回报率（ROI）。

题目提供了一套商家及其在“双11”日促销期间获得的相应新买家。任务是预测对于指定商家的新买家将来是否会成为忠实客户。即预测这些新买家在6个月内再次从同一商家购买商品概率。一个包含大约20万用户的数据集用于训练，还有一个类似大小的数据集用于测试。

数据格式

官方给了数据：data_format1

data_format1: user_log_format1, user_info_format1, test_format1, train_format1

用户行为日志：包含用户ID、商品ID、商品类别、商户ID、商品品牌、时间和用户行为类别7个特征。

用户信息：包含用户ID、用户年龄段和用户性别信息。

训练集和测试集：分别包含用户ID、商户ID和是否为重复买家标签，其中训练集标签为0-1，测试集标签为空，需要预测。

数据量

Name ▲	Type	Size	Value
data1	DataFrame	(54925330, 7)	Column names: user_id, item_id, cat_id, seller_id, brand_id, time_stam ...



步骤2：数据清洗

进行brand_id缺失值(91015)填充，并使用pickle模块进行序列化，加快速度读写

步骤1：数据压缩

压缩csv中的数据，通过改变扫描每列的dtype，转换成适合的大小。

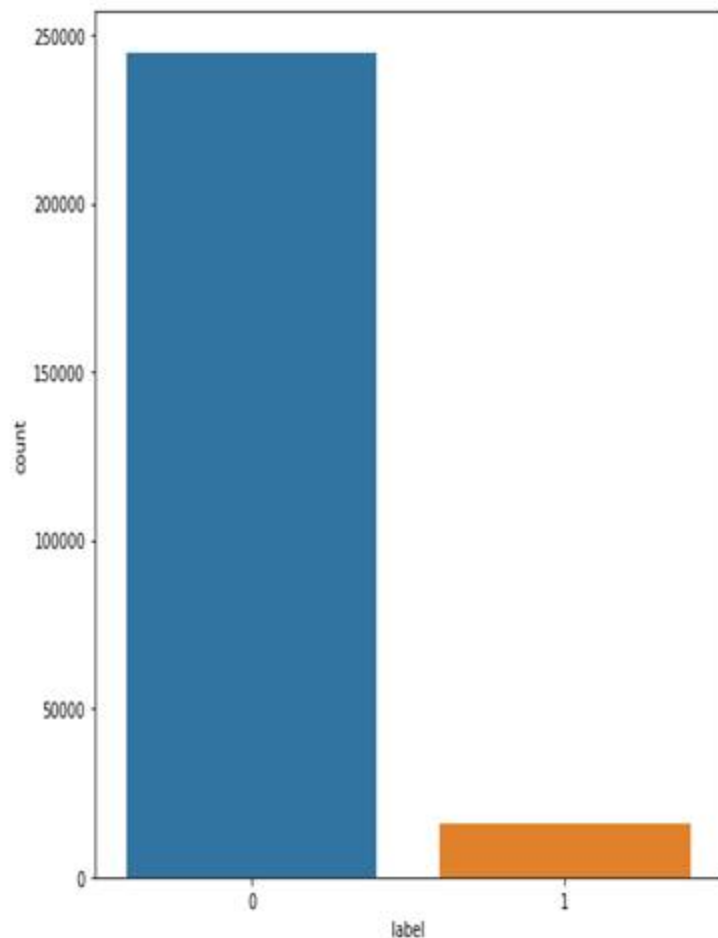
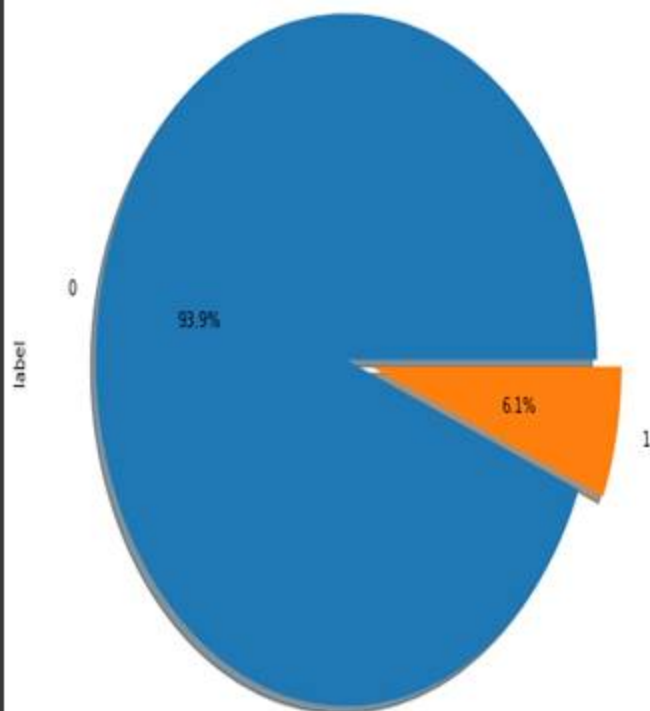
步骤3：数据可视化

读取训练集，对正负样本、正负样本与性别比例、正负样本与年龄段的比例进行可视化。

数据可视化

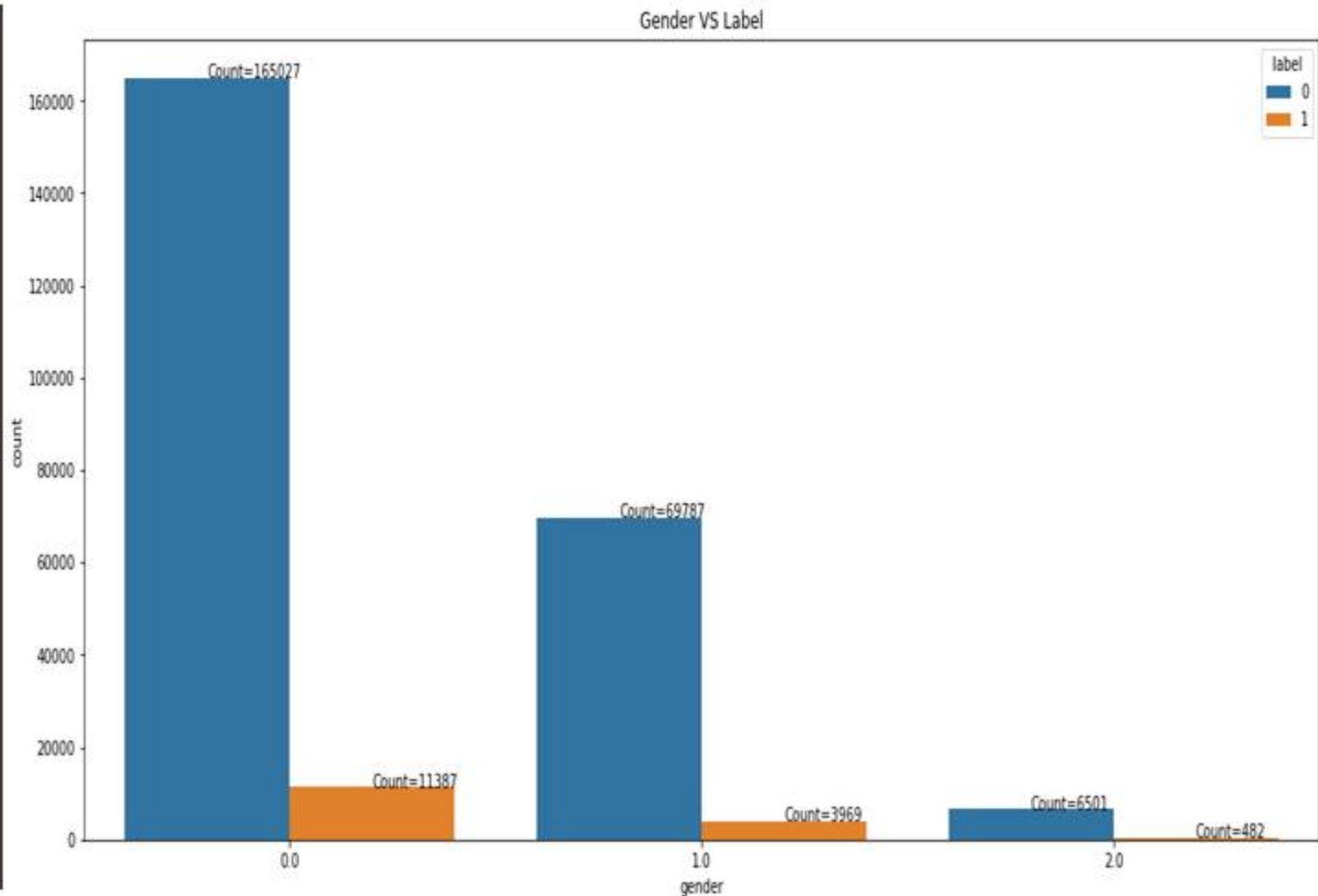
训练集正负样本可
视化：

训练集中label取值
范围 {0, 1}, 1表示
重复购买, 0 表示
非重复购买。



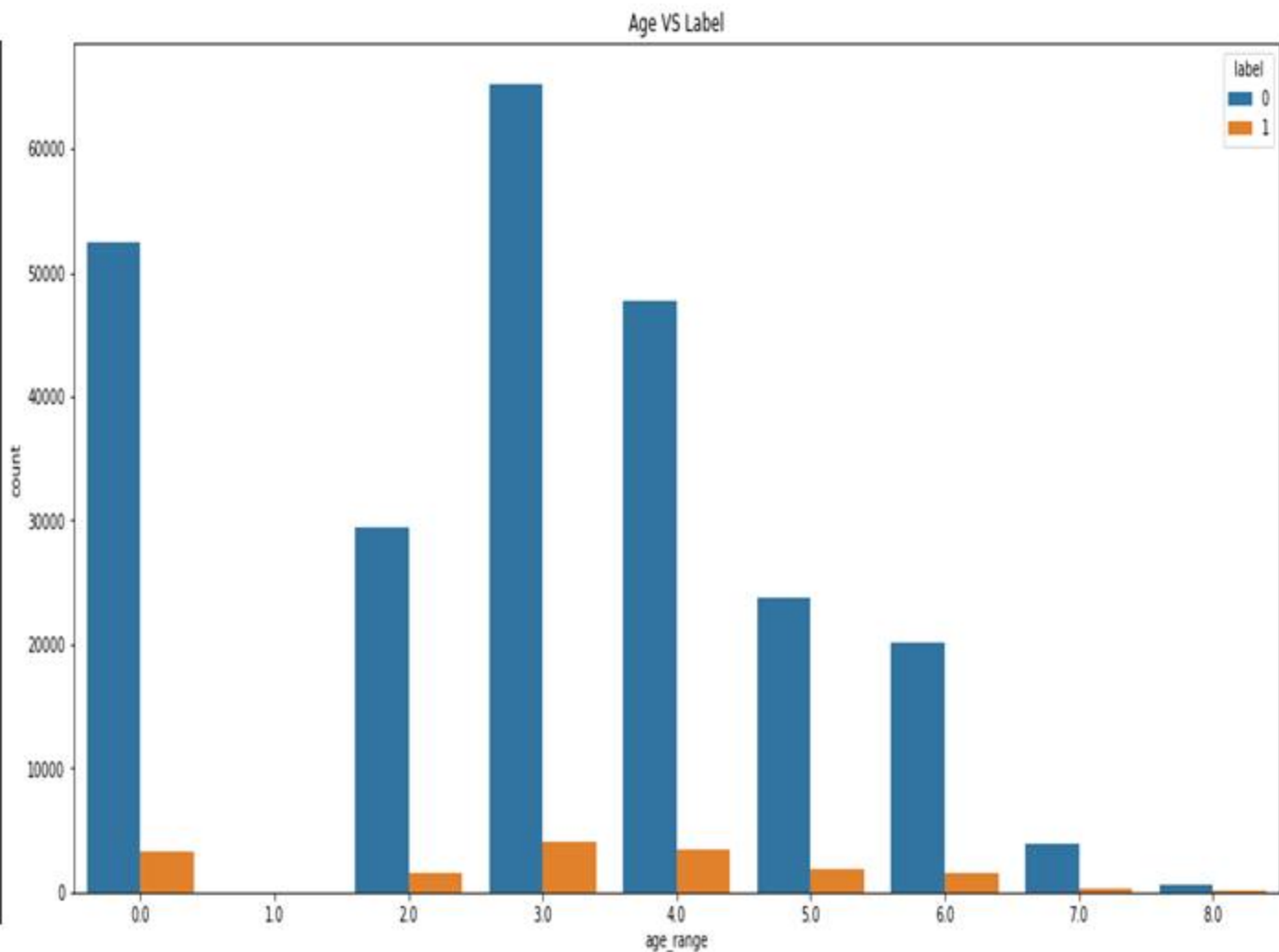
数据可视化

读取用户信息数据，
并与训练集数据进行合并；
展示正负样本与用户
性别比例；
顾客性别：0 表示
女性，1 表示男
性，2 and NULL 表
示未知。



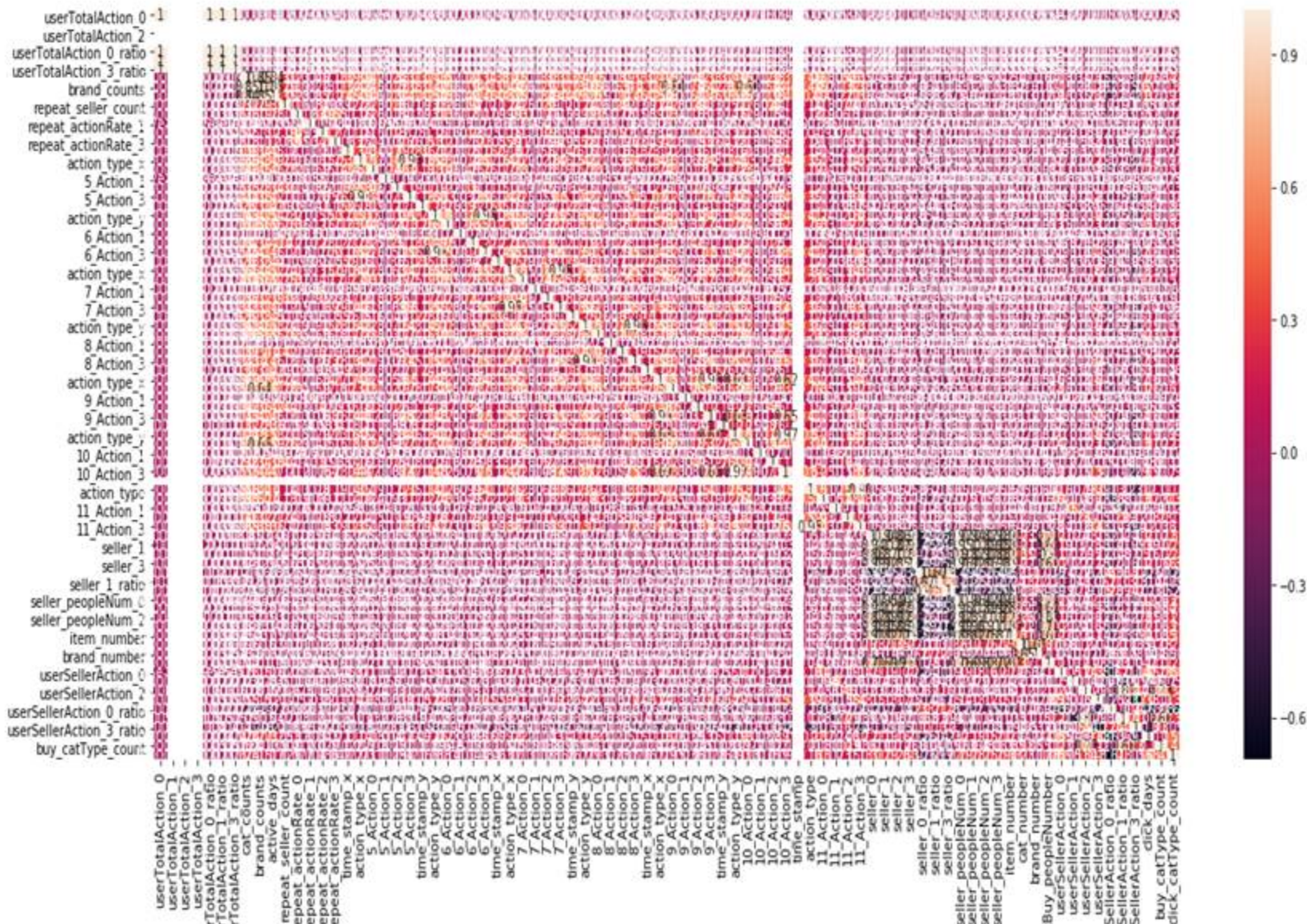
数据可视化

展示正负样本与用户年龄段的比例；
顾客年龄范围：1 表示<18; 2 表示[18,24]; 3 表示[25,29]; 4 表示[30,34]; 5 表示[35,39]; 6 表示[40,49]; 7 and 8 表示 ≥ 50 ;
0 and NULL 表示未知





0	age_0.0	q	111	save_days	q
1	age_1.0	q	112	item_click_count	q
2	age_2.0	q	113	item_add_count	q
3	age_3.0	q	114	item_buy_count	q
4	age_4.0	q	115	item_save_count	q
5	age_5.0	q	116	cat_click_count	q
6	age_6.0	q	117	cat_add_count	q
7	age_7.0	q	118	cat_buy_count	q
8	age_8.0	q	119	cat_save_count	q
9	female	q	120	brand_click_count	q
10	male	q	121	brand_add_count	q
11	unknown	q	122	brand_buy_count	q
12	userTotalAction_0	q	123	brand_save_count	q



基模型:

LGBM 、 XGBoost 、 MLP 、 GBDT 、
RandomForest

集成学习:

GBM

“

	train	test	final
AUC	0.7112	0.6731	0.6775

”

43	_ssssyy	浙江大学	0.681079	2018-01-10
44	大西瓜瓜	盒子科技	0.681011	2018-10-07
45	大厉	浙江大学	0.680769	2017-12-21
46	控几我寄几	University of Aberdeen	0.679506	2018-05-31
47	DeepDarkFantasy.j...	其它-上海科技大学	0.679450	2018-06-17
48	小七要读博	天津理工	0.678982	2018-05-31
49	凉口三三	重庆邮电大学	0.678950	2018-05-31
50	lccc0312	某厂	0.678352	2017-04-21
51	美帝掌握核心科技	电子科技大学	0.678300	2018-05-31
52	zweiHasen_rcababitt	其它-上海科技大学	0.678231	2018-06-17
53	downle	Downle	0.678102	2017-03-14
54	zweiHasen_meeto	其它-上海科技大学	0.677829	2018-06-17
55	Texas_2019	University of Toronto	0.677663	2018-06-08
56	丁兆云dm杨凯晶	国防科大	0.677507	2018-11-21