厦门大学

XIAMEN UNIVERSITY

B2C 电商高转化潜力用户购买意向预测

High potential users purchase intention forecast

学　　院：　管理学院

年　　级：　2021 级

成　　员：　邹宇涛　卓翔

二〇二四年 十二月 一日

目录

1 问题定义

本问题来源于京东 2017 年算法大赛-高潜用户购买意向预测。京东作为中国最大的自营式电商，在保持高速发展的同时，沉淀了数亿的忠实用户，积累了海量的真实数据。如何从历史数据中找出规律，去预测用户未来的购买需求，让最合适的商品遇见最需要的人，是大数据应用在精准营销中的关键问题，也是所有电商平台在做智能化升级时所需要的核心技术。以京东商城真实的用户、商品和行为数据（脱敏后）为基础，通过数据挖掘的技术和机器学习的算法，构建用户购买商品的预测模型，输出高潜用户和目标商品的匹配结果，为精准营销提供高质量的目标群体。即使用京东多个品类下商品的历史销售数据，构建算法模型，预测用户在未来 5 天内，对某个目标品类下商品的购买意向。

1.1 研究背景

随着电子商务的快速发展，电商平台积累了大量的用户行为数据、商品数据以及用户评价数据。这些数据蕴含着丰富的用户偏好和购买意图信息，对于电商平台而言，如何有效挖掘和利用这些数据，以精准预测用户的购买行为，成为提升用户体验和销售业绩的关键。本实验旨在通过数据挖掘技术，对给定的用户行为数据、商品数据以及评价数据进行深入分析，预测特定时间段内用户对候选商品子集（P）的购买行为。

1.2 研究意义

首先，通过精准预测用户的购买行为，电商平台可以为用户提供更加个性化的商品推荐，减少用户搜索和筛选商品的时间，提升用户的购物体验。其次，对用户的购买行为进行预测，有助于电商平台优化库存管理、制定针对性的营销策略，从而提高销售效率和盈利能力。再次，本实验将数据挖掘技术应用于实际电商场景中，通过实践验证和优化算法，推动数据挖掘技术在电商领域的发展和应用。

1.3 相关理论与概念

1.3.1 用户行为数据

用户行为数据记录了用户在电商平台上的各种行为，如浏览、加入购物车、下单等。这些数据反映了用户的购买意图和偏好，是构建预测模型的重要输入。

1.3.2 商品数据

商品数据包括商品的编号、属性、品类、品牌等信息。这些信息有助于了解商品的特征和市场需求，是构建预测模型时需要考虑的重要因素。

1.3.3 评价数据

评价数据记录了用户对商品的评论和评分等信息。这些数据反映了用户对商品的满意度和口碑，对于预测用户的购买行为具有重要参考价值。

2 数据准备与前期处理

2.1 用户与商品数据预处理

由于爬取的信息种类繁多，数量庞大，因此，我们需要预先对所得到的数据进行处理。数据主要分为四类：用户数据（df_user）、商品数据（df_product）和用户行为数据（df_action），具体的操作步骤包括：数据读取、数据类型转换与缺失值处理、数据过滤、数据合并与同步、清洗结果保存与加载。这里对部分步骤作出说明。

2.1.1 数据类型转换

将用户表中的性别和年龄、商品表中的商品类别和品牌从的浮点数或字符串转换为整型。将用户表中的注册时间字符串，评论表中的评论时间字符串，行为表中的行为时间转换为日期时间格式。通过数据类型转换以保证数据一致性，避免后续操作因类型不匹配导致错误。

2.1.2 缺失值处理

将缺失的性别和年龄用-1 填充，将商品表所有缺失值统一填充为-1，将缺失的评论数填充为 0，直接删除行为表中的缺失记录。

部分输出结果如下图所示。

```
user_id                    int32
age                        int32
sex                        int32
user_lv_cd                 int32
user_reg_tm    datetime64[ns]
```

图 1 用户表数据类型

	user_id	age	sex	user_lv_cd	user_reg_tm
0	200001	-1	-1	5	2016-01-26
1	200002	-1	0	1	2016-01-26
2	200003	-1	1	4	2016-01-26
3	200004	-1	-1	1	2016-01-26
4	200005	-1	0	4	2016-01-26

图 2 用户表输出结果

2.1.3 数据过滤

对于用户数据，通过筛选用户表(user_df)中注册时间大于等于 2016-04-16 的用户。保留的用户为目标分析人群，即在指定时间范围内注册的活跃用户。对于行为数据，筛选出行为表(df_action)中发生在 2016-04-16 及以后的用户行为数据。剔除发生在 2016-04-16 及以后的行为数据，保证行为数据与分析的时间范围一致。数据过滤操作确定了特征提取的时间范围，确保生成的特征与当前分析目标匹配。

	user_id	age	sex	user_lv_cd	user_reg_tm
0	200001	-1	-1	5	2016-01-26
4	200005	-1	0	4	2016-01-26
13	200014	-1	-1	4	2013-04-10
14	200015	-1	1	3	2016-01-26
16	200017	-1	-1	4	2016-01-26

图 3 过滤后的用户表

2.1.4 数据合并与同步

用户表和行为表合并， 将用户的属性（如年龄、性别、注册时间等）与行为数据整合起来，并过滤掉没有对应行为数据的用户，或没有对应用户信息的行为数据。

行为表和商品表合并，按照商品 ID(sku_id)将行为表与商品表进行左连接。保留行为数据中的所有记录，同时引入商品的属性信息（如类别 cate 和品牌 brand）。将用户的行为数据与商品属性关联起来，提取商品类别和品牌相关的特征。

通过时间过滤操作，确保用户注册时间和行为发生时间一致。在行为表中仅保留属于过滤后用户的数据记录。保证行为数据与目标用户群体同步，避免出现行为数据与用户数据时间范围不一致的情况。确保后续提取的特征是基于目标用户的有效行为。

	user_id	age	sex	user_lv_cd	user_reg_tm
0	200001	-1	-1	5	2016-01-26
4	200005	-1	0	4	2016-01-26
13	200014	-1	-1	4	2013-04-10
14	200015	-1	1	3	2016-01-26
16	200017	-1	-1	4	2016-01-26

	sku_id	a1	a2	a3	cate	brand	comment_num	has_bad_comment	\
16	100119	2	1	2	8	812	1	0	
27	100174	2	2	1	8	812	2	0	
40	100310	3	2	-1	8	214	0	0	
45	100344	3	1	1	8	214	0	0	
69	100462	3	1	2	8	214	3	0	

	bad_comment_rate
16	0.0
27	0.0
40	0.0
45	0.0
69	0.0

	user_id	sku_id	time	model_id	type	cate	brand
0	266079	138778	2016-01-31 23:59:02	NaN	1	8	403
1	266079	138778	2016-01-31 23:59:03	0.0	6	8	403
2	200719	61226	2016-01-31 23:59:07	NaN	1	8	30
3	200719	61226	2016-01-31 23:59:08	0.0	6	8	30
15	266079	138778	2016-01-31 23:59:40	0.0	6	8	403

图 4 同步清洗后的用户、商品、行为表

2.2 原始数据可视化分析

本章基于本组代码中数据预处理部分后的可视化操作阐述了特征工程前可能被考虑的一些因素及对于原始数据可观察的一些趋势。

2.2.1 用户行为数据可视化

我们将源数据中发生购买行为的用户数（user_cnt），被购买商品个数（sku_cnt），加总得到的购买次数（buy_cnt）按周/月进行可视化分析。

从总的趋势来看，整体来看，一周的购买趋势呈现出一种波动性，周二和周四是销售高峰，而周末则是低谷。

从周一到周二，我们可以看到购买商品用户数和总购买次数都有所增加，这表明周二是一周中的销售高峰。然而，尽管周二的购买商品用户数较高，但购买商品数并没有显著增加，这可能意味着用户在周二购买的商品种类或数量并没有显著增加，或者可能存在一些大额购买行为。周三的数据出现了明显的下降。周四的数据再次上升，尤其是购买商品用户数，这可能表明用户在经历了周三的低谷后，又开始活跃起来。周五的数据与周四相似，但周六和周日的数据再次下降，可能是由于周末用户的行为模式发生了变化，他们可能更倾向于休闲等其他活动。

从图表上我们可以进行简单的乘除得到平均购买商品数的情况，似乎周三和周日的用户平均购买商品数最高，可能可以表明这两天的用户消费力较强且有多个商品消费意愿，而不是单一商品选择，尽管总的购买次数和商品数不是最高。

周六的平均购买次数最低，但平均购买商品数却最高，这可能表明周六的用户在购买时更倾向于一次性购买更多商品。

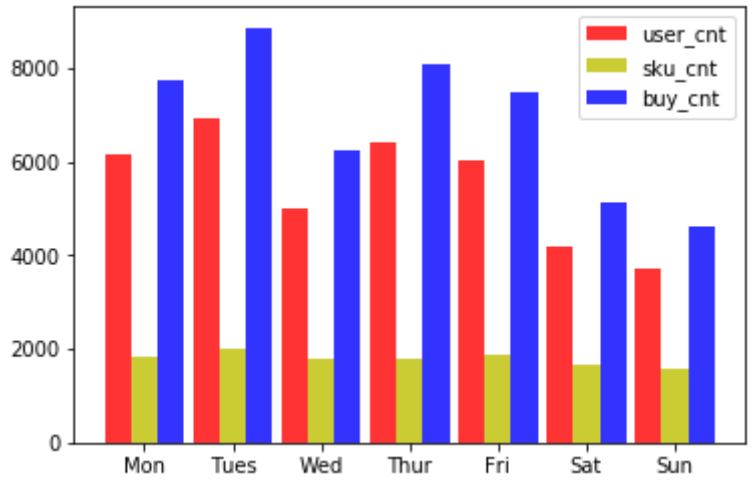

图 5　周一到周日有购买行为用户数/商品数/购买行为次数数据

我们又将以上三种数据拆分到一个月的数据周期当中来看，整体来说一个月的购买趋势呈现出先减后增的趋势，第一周的用户购买次数、商品数、购买行为用户数都在逐渐下降，第二周虽然低但是保持稳定，直到第二周末第三周初开始上升，成 U 字形，第三周和第四周都维持了良好且稳定的数据水平。

由此可得，月初和月末的用户购买行为较为频繁，且日期可能是一个可以加入参与考虑的因素。

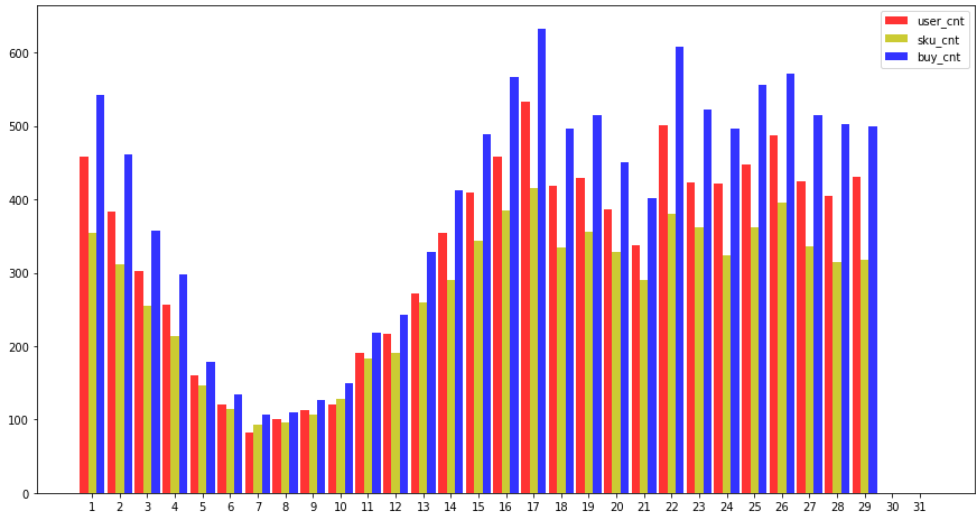

图 6　一个月周期内有购买行为用户数/商品数/购买行为次数数据

2.2.2 品类销售数据可视化

大部分品类在周内的销售并没有太多特色，随着整体销售情况的上升/下降保持着均匀的趋势，然而消费者对于某些品类的在不同日期的偏爱可能会影响该品类销售数据的占比。

如我们观察到，品类 8 在星期二和星期五的销售数量显著增加，尤其是星期二，销售数量接近 3000，在其余日子也是销售情况最好的一类，表明消费者对于该品类的喜爱。

品类 4 和品类 5 的销售数量则内相对稳定且占比较高，尽管如品类 4 在星期四有突增的波动，但整体较为稳定。

品类 9 在星期二的销售表现类似品类 8，但后续销售回落并稳定，可能说明星期二对于品类 8 和品类 9 是某种促销或者引起消费者关注的日子。

整体销售趋势上看，品类 8 大于品类 4，大于品类 5，大于品类 6，大于品类 7，大于屁呢里 9，大于品类 10 再大于品类 11 是一个大致情况，说明消费者对于这些品类有着天然的消费偏好。

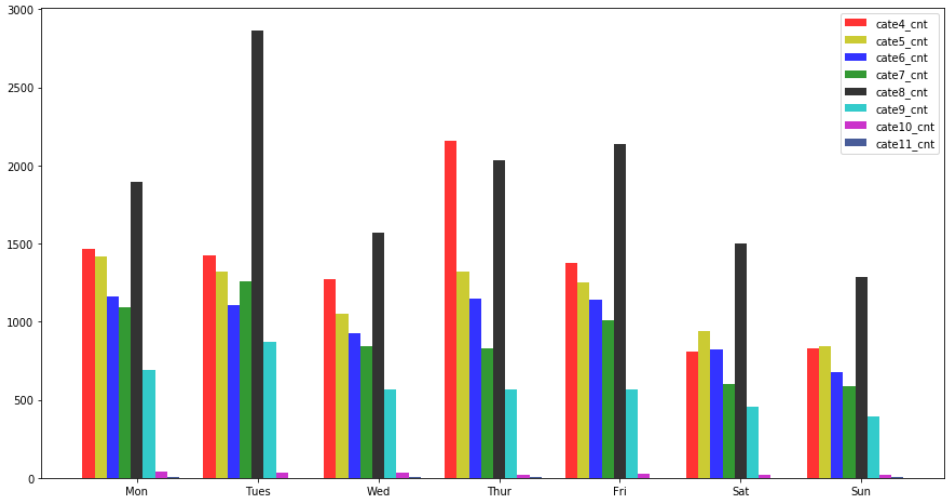

图 7　周内不同品类销售数据

我们又将以上数据拆分到一个月的数据周期当中来看，整体来说一个月的购买趋势中，月初、月中和月末呈现出比较不同的品类销售情况，月中时品类 8 销售占比尤其高，月初时品类 7 销售情况良好，在月中和月末的时候品类 4

销售情况优秀，其余时间段品类销售情况大体一致维持了良好且稳定的数据水平。

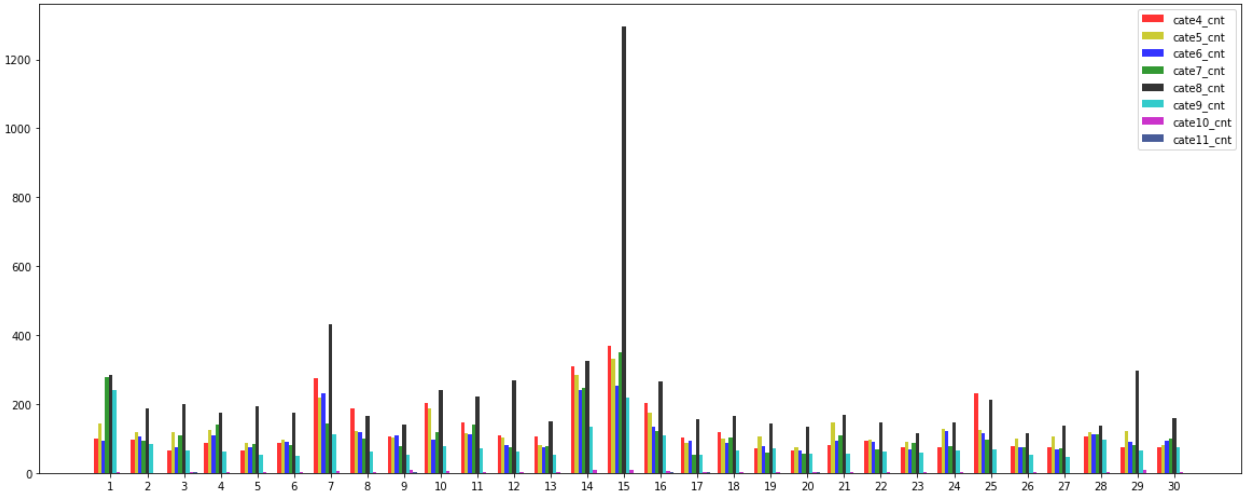

图 8　一个月内不同品类销售数据

2.3 特征工程

特征工程是将原始数据转化为特征，并根据所研究问题的领域知识提取、构 建、删减或组合变化得到新的特征，以提升模型预测能力的过程。

我们分别获取了用户的行为总特征、用户对某类别商品的行为特征、用户对某品牌商品的行为特征、用户对某特定商品的行为特征、商品被用户行为的特征以及用户的固有特征和商品的固有特征。通过对用户在行为表中的行为（type）进行分组统计，可以得到每个用户不同行为（如浏览、加入购物车、购买等）的发生次数，并通过占比计算，得到用户购买行为占用户总行为次数的比例，衡量用户行为的转化效率。

2.3.1 用户行为总特征

首先进行时间段的筛选，只保留行为发生在指定时间段内的数据。将数据按 user_id 和 type 分组，统计行为发生的次数，并用购买次数除以其他行为的次数，计算出购买率，最后将比率值限制在合理范围（[0,1]）。

	user_id	browse_cnt	addcart_cnt	delcart_cnt	buy_cnt	follow_cnt	click_cnt	browse_rate	addcart_rate	delcart_rate	follow_rate	click_rate
0	200001	57.0	12.0	6.0	1.0	0.0	86.0	0.017544	0.083333	0.166667	1.0	0.011628
1	200005	36.0	0.0	0.0	0.0	0.0	56.0	0.000000	0.000000	0.000000	0.0	0.000000
2	200014	144.0	10.0	4.0	0.0	0.0	151.0	0.000000	0.000000	0.000000	0.0	0.000000
3	200015	392.0	2.0	2.0	0.0	0.0	660.0	0.000000	0.000000	0.000000	0.0	0.000000
4	200017	16.0	0.0	0.0	1.0	0.0	17.0	0.062500	1.000000	1.000000	1.0	0.058824
...	...	...	...	...	...	...	...	...	...	...	...	...
29425	305301	124.0	1.0	0.0	0.0	0.0	269.0	0.000000	0.000000	0.000000	0.0	0.000000
29426	305308	98.0	1.0	0.0	1.0	0.0	192.0	0.010204	1.000000	1.000000	1.0	0.005208
29427	305313	38.0	1.0	0.0	1.0	0.0	87.0	0.026316	1.000000	1.000000	1.0	0.011494
29428	305317	54.0	0.0	0.0	0.0	0.0	52.0	0.000000	0.000000	0.000000	0.0	0.000000
29429	305318	50.0	3.0	0.0	1.0	0.0	79.0	0.020000	0.333333	1.000000	1.0	0.012658

图 9 行为次数和行为比率表

2.3.2 用户对某类别商品的行为特征

在进行时间段筛选后，将行为表和商品表关联，获取类别信息，并按 user_id 和 cate 聚合，删除无关的 sku_id 列，仅保留用户 ID、类别和行为信息统计每种行为的次数，最后同样计算行为比率并限制比率范围。

	user_id	cate	browse_cnt	addcart_cnt	delcart_cnt	buy_cnt	follow_cnt	click_cnt	browse_rate	addcart_rate	delcart_rate	follow_rate	click_rate
0	200001	8	47.0	11.0	0.0	1.0	0.0	65.0	0.021277	0.090909	1.0	1.0	0.015385
1	200005	8	10.0	0.0	0.0	0.0	0.0	19.0	0.000000	0.000000	0.0	0.0	0.000000
2	200014	8	123.0	5.0	3.0	0.0	0.0	126.0	0.000000	0.000000	0.0	0.0	0.000000
3	200015	8	30.0	0.0	0.0	0.0	0.0	26.0	0.000000	0.000000	0.0	0.0	0.000000
4	200017	8	16.0	0.0	0.0	1.0	0.0	17.0	0.062500	1.000000	1.0	1.0	0.058824
...	...	...	...	...	...	...	...	...	...	...	...	...	...
24319	305292	8	14.0	2.0	1.0	1.0	0.0	18.0	0.071429	0.500000	1.0	1.0	0.055556
24320	305296	8	26.0	1.0	0.0	0.0	0.0	32.0	0.000000	0.000000	0.0	0.0	0.000000
24321	305301	8	118.0	1.0	0.0	0.0	0.0	257.0	0.000000	0.000000	0.0	0.0	0.000000
24322	305313	8	5.0	0.0	0.0	0.0	0.0	7.0	0.000000	0.000000	0.0	0.0	0.000000
24323	305317	8	34.0	0.0	0.0	0.0	0.0	35.0	0.000000	0.000000	0.0	0.0	0.000000

图 10 对某类商品的行为次数和比率表

2.3.3 用户对某品牌商品的行为特征

在进行时间段筛选后，将行为表和商品表关联，获取品牌信息。删除无关的 sku_id 列，保留 user_id、brand 和 type 等关键列。按 user_id、brand、type 分组，统计每种行为的发生次数，最后计算行为比率并限制比率范围。

	user_id	brand	type_1_count	type_2_count	type_3_count	type_4_count	type_5_count	type_6_count	browse_rate	addcart_rate	delcart_rate	follow_ra
0	200001	214	2	0	0	0	0	4	0.000000	0.000000	0.0000	0.00
1	200001	306	26	3	1	1	0	43	0.038462	0.333333	0.9999	0.99
2	200001	403	20	4	3	0	0	34	0.000000	0.000000	0.0000	0.00
3	200001	489	5	3	1	0	0	0	0.000000	0.000000	0.0000	0.00
4	200001	800	4	2	1	0	0	5	0.000000	0.000000	0.0000	0.00
...	...	...	...	...	...	...	...	...	...	...	...	
129641	305317	30	8	0	0	0	0	9	0.000000	0.000000	0.0000	0.00
129642	305317	800	46	0	0	0	0	43	0.000000	0.000000	0.0000	0.00
129643	305318	30	28	3	0	1	0	38	0.035714	0.333333	0.9999	0.99
129644	305318	214	20	0	0	0	0	38	0.000000	0.000000	0.0000	0.00
129645	305318	489	2	0	0	0	0	3	0.000000	0.000000	0.0000	0.00

图 11 对某品牌商品的行为次数和比率表

2.3.4 用户对某特定商品的行为特征

进行时间段筛选后只保留 user_id（用户 ID）、sku_id（商品 ID）、type（行为类型）和 time（时间）列，并根据 user_id、sku_id、type 进行分组，并统计每种行为类型发生的次数，汇总每个用户对每个商品的所有行为次数，最后计算行为比率并限制比率范围。

	user_id	sku_id	browse_cnt	addcart_cnt	delcart_cnt	buy_cnt	follow_cnt	click_cnt	browse_rate	addcart_rate	delcart_rate	follow_rate	click_rate
0	200074	62872	1	0	1	0	0	1	0.0	0.0	0.0	0.0	0.0
1	200074	90521	1	0	0	0	0	2	0.0	0.0	0.0	0.0	0.0
2	200074	142829	1	0	0	0	0	1	0.0	0.0	0.0	0.0	0.0
3	200074	145946	3	0	1	0	0	4	0.0	0.0	0.0	0.0	0.0
4	200074	148856	5	1	0	0	0	9	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11528	305225	131300	12	1	0	0	0	31	0.0	0.0	0.0	0.0	0.0
11529	305225	141167	2	0	0	0	0	4	0.0	0.0	0.0	0.0	0.0
11530	305225	153395	6	1	0	0	0	14	0.0	0.0	0.0	0.0	0.0
11531	305225	166876	0	0	0	0	0	4	0.0	0.0	0.0	0.0	0.0
11532	305239	37957	1	2	0	0	0	2	0.0	0.0	0.0	0.0	0.0

图 12 对某特定商品的行为次数和比率表

2.3.5 商品被用户行为的特征

进行时间段筛选后仅保留 sku_id（商品 ID）、type（行为类型）、time（时间）列，并根据 sku_id 和 type 进行分组，统计对于每个商品，不同类型的行为发生的次数。最后按照 sku_id 进行分组，汇总每个商品在所有行为上的总次数，并计算行为比率，以该比率来衡量每种行为对购买的影响，分析哪些行为更容易促成购买，帮助优化产品和营销策略。

	sku_id	browse_cnt	addcart_cnt	delcart_cnt	buy_cnt	follow_cnt	click_cnt	browse_rate	addcart_rate	delcart_rate	follow_rate	click_rate
0	95	1	0	0	0	0	1	0.0	0.0	0.0	0.0	0.0
1	276	23	0	0	0	0	74	0.0	0.0	0.0	0.0	0.0
2	281	3	0	0	0	0	5	0.0	0.0	0.0	0.0	0.0
3	499	21	0	0	0	1	31	0.0	0.0	0.0	0.0	0.0
4	661	11	0	0	0	0	25	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...
1273	170561	22	0	0	0	0	41	0.0	0.0	0.0	0.0	0.0
1274	170587	6	0	0	0	0	11	0.0	0.0	0.0	0.0	0.0
1275	170870	17	1	0	0	0	17	0.0	0.0	0.0	0.0	0.0
1276	170998	1	0	0	0	0	2	0.0	0.0	0.0	0.0	0.0
1277	171182	7	0	0	0	0	12	0.0	0.0	0.0	0.0	0.0

图 13 商品被用户行为的次数和比率

2.3.6 用户的固有特征(整个时间段的基本特征)

通过已定义的函数分别生成用户的行为特征、对某类别商品的行为特征、对某品牌商品的行为特征以及对某单品商品的行为特征，最后将所有生成的特征（包括用户基本特征和四类行为特征）存储在一个列表中。

```
   user_id  browse_cnt  addcart_cnt  delcart_cnt  buy_cnt  follow_cnt  \
0   200001        57.0         12.0          6.0      1.0         0.0
1   200005        36.0          0.0          0.0      0.0         0.0
2   200014       144.0         10.0          4.0      0.0         0.0
3   200015       392.0          2.0          2.0      0.0         0.0
4   200017        16.0          0.0          0.0      1.0         0.0

   click_cnt  browse_rate  addcart_rate  delcart_rate  follow_rate  click_rate
0       86.0     0.017544      0.083333      0.166667          1.0    0.011628
1       56.0     0.000000      0.000000      0.000000          0.0    0.000000
2      151.0     0.000000      0.000000      0.000000          0.0    0.000000
3      660.0     0.000000      0.000000      0.000000          0.0    0.000000
4       17.0     0.062500      1.000000      1.000000          1.0    0.058824
   user_id  cate  type  cnt
0   200001     8     1   57
1   200001     8     2   12
2   200001     8     3    6
3   200001     8     4    1
4   200001     8     6   86
   user_id  cate  browse_cnt  addcart_cnt  delcart_cnt  buy_cnt  follow_cnt  \
0   200001     8        57.0         12.0          6.0      1.0         0.0
1   200005     8        36.0          0.0          0.0      0.0         0.0
2   200014     8       144.0         10.0          4.0      0.0         0.0
3   200015     8       392.0          2.0          2.0      0.0         0.0
4   200017     8        16.0          0.0          0.0      1.0         0.0

   click_cnt
0       86.0
1       56.0
2      151.0
3      660.0
4       17.0
```

图 14 用户固有特征表

2.3.7 商品的固有特征(整个时间段的基本特征)

复制输入的商品数据 sku_data，保留商品的基本信息，并调用已有函数，计算商品在时间段（2016-01-01 到 2016-06-07）内的用户行为特征。该函数会统计用户对该商品的各类行为，并生成相应的统计数据。最后将商品的基本信息商品的行为特征合并成一个列表，形成一个完整的特征集。

3 模型选择与训练

3.1 模型介绍及选择依据

3.1.1 CatBoost 模型介绍

CatBoost 是一种高效的梯度提升决策树（GBDT）算法。CatBoost 的名字来源于"Categorical Boosting"，表示它对分类特征有非常好的处理能力。CatBoost 主要用于解决结构化数据的回归、分类以及排名任务。

- **主要特点：**

1. 对类别特征的自动处理：CatBoost 可以高效地处理类别特征，无需手动对类别变量进行独热编码（One-Hot Encoding）或标签编码（Label Encoding）。它通过计算类别特征的统计特征（如频率、均值等）来自动生成特征，从而减少了数据预处理的复杂性。
2. 高效性：CatBoost 在计算效率上做了很多优化。通过使用对称树构建和特殊的排序算法，CatBoost 在训练时间上较为高效，尤其在处理大规模数据时，训练速度较 XGBoost 有所提高。
3. 避免过拟合：CatBoost 引入了特殊的排序策略，能够有效避免模型在训练集上过拟合。

CatBoost 模型可以表示为多个决策树的加权和，数学上可以表示为：

$$\hat{y} = f(x) = \sum_{i=1}^{M} \gamma_i h(x; T_i, F_i)$$

其中：

- $\hat{y}$ 是模型的预测值。
- M 是决策树的数量。
- γ_i 是第 i 棵树的权重。
- $h(x; T_i, F_i)$ 是第 i 棵树的预测函数，T_i 表示树的结构，F_i 表示树的参数（例如，分割点）。

CatBoost 的目标是最小化以下损失函数：

$$L(y, f(x)) = \sum_{i=1}^{N} l(y_i, f(x_i)) + \sum_{i=1}^{M} P(T_i, F_i)$$

其中：

- N 是样本数量。
- y_i 是第 i 个样本的真实标签。
- l 是损失函数，例如对数损失（用于分类问题）或均方误差损失（用于回归问题）。
- $P(T_i, F_i)$ 是正则化项，用于控制树的复杂度，防止过拟合。

CatBoost 通过梯度下降算法来优化上述损失函数，每一步都会添加一个新的树来纠正前一步的残差。

3.1.2 XGBoost 模型介绍

XGBoost 是一个高效的梯度提升决策树（GBDT）实现，广泛应用于机器学习竞赛和实际生产环境中。XGBoost 的优越性能和高效性使其成为结构化数据分析的主流工具之一。

• 主要特点

1. 高效性：XGBoost 通过基于树的并行化计算和压缩存储优化了计算效率，使得它在训练大规模数据时表现非常优秀。
2. 正则化:XGBoost 引入了 L1 和 L2 正则化(类似于 Lasso 和 Ridge 回归)，可以有效控制模型复杂度，减少过拟合现象。
3. 支持多种优化算法：XGBoost 支持多种优化算法，如二阶泰勒展开和行列式计算等，确保了模型训练的高效性。
4. 支持缺失值处理：XGBoost 可以在模型训练过程中自动处理缺失值，不需要显式地填充缺失数据。

特性	CatBoost	XGBoost
对类别特征的处理	自动处理类别特征，无需手动编码	需要手动处理类别特征
训练速度	较快，尤其是在大规模数据下	较快，但在类别特征处理上较慢
防过拟合能力	较强，采用特殊的排序机制	需要调参，较容易过拟合
GPU 支持	支持 GPU 加速	支持 GPU 加速
易用性	高，默认参数效果较好	较复杂，需要手动调参
适用场景	更适合类别数据处理的任务	对结构化数据的回归和分类任务广泛适用
适用范围	类别特征非常丰富的数据集	适用于各种类型的结构化数据

表 1 CatBoost vs XGBoost

3.1.3 模型选择依据

我们的模型选择基于特征工程和数据集分割特点的考量，并在对比其他模型后，选择了结果较好的 CatBoost 和 XGBoost 模型。以下是对基于数据特点的模型选择依据分析：

首先，这两个模型核心算法分别为 Ordered Boosting、Regularized Boosting 算法。

数据集中存在大量的类别特征，比如用户 ID（user_id）、商品 ID（sku_id）、类别 ID（cate）等，需要模型能够有效处理类别型变量，而 Ordered Boosting 算法能够直接处理高基数的类别型特征，无需进行独热编码，减少了维度灾难，提高了模型效率。

并且购买行为通常在数据集中属于少数，需要模型具备处理不平衡数据的能力。在处理类别不平衡的数据时，Ordered Boosting 算法能够更好地捕捉少数类别的特征。

同时，在经过特征工程后，数据中包含大量的数值型特征，XGBoost 在处理数值特征上具有高效性。

同时，这两个模型从传统的梯度提升算法发展又各有特点。比如 Ordered Boosting 算法对数据顺序特殊处理从而能够防止目标泄露，而 Regularized Boosting 通过引入正则化项（如 L1 和 L2 正则化），控制模型的复杂度，减少过拟合的风险。

3.2 数据集分割

3.2.1 训练集的构建

在特征工程中我们构建了关于用户行为次数统计和这些行为对于用户购买的转化率这些必要的特征。之后为了精准识别用户的购买行为，我们还需要对这些统计量去进行更加精细的研究，以获取更多关于用户购买行为的特点。

所以我们利用这些统计量，尝试获取了用户某个时间段内总的用户行为特征、某个时间段内用户针对某类别 cate 商品的行为特征、某个时间段内用户针对某品牌 brand 商品的行为特征、某个时间段内用户针对某商品 sku_id 的行为特征以及商品在某段时间内被用户们进行了何种行为的特征这五个方面。

然后对于这五个方面，我们认为在不同的时间段内其具有的意义效果也是有显著差异的。用户在这五个方面的行为在越靠近其产生购买的时间点上，可能会有更显著的差异，例如顾客进行了浏览行为或加入购物车的行为后经过思想决策后可能很快就会执行购买行为，而不是过了 20 多天重新回来看到再进行购买。所以对于购买前的时间段划分我们选择使用购买前 1 天，2 天，3 天，5 天，7 天，10 天，15 天，20 天，30 天这样的时间间隔对其行为特征分别进行统计。

在数据集的构建方面，我们通过保留产生过购买行为的用户数据，将所统计的不同方面的不同时间段的特征属性通过 user_id,sku_id 进行连结，获得了一个具有 565 列属性的数据集。其中包括用户的个人信息、商品的信息，还有占具极大部分的行为特征统计量在不同方面不同划分时间间隔内的数据信息。

```
2016-03-07 00:00:00 2016-03-12 00:00:00
len_df_dataset:  141119
len_df_dataset:  141119
```

	user_id	sku_id	cate	brand	age	sex	user_lv_cd	user_reg_tm	browse_cnt_user_all
0	200015	9702	8	693	-1	1	3	2016-01-26	392.0
1	200015	24371	8	545	-1	1	3	2016-01-26	392.0
2	200015	40399	8	214	-1	1	3	2016-01-26	392.0
3	200015	44854	8	30	-1	1	3	2016-01-26	392.0
4	200015	64467	8	214	-1	1	3	2016-01-26	392.0
...	...	...	...	...	...	...	...	...	...
141114	305313	18024	8	214	-1	1	3	2016-01-26	38.0
141115	305313	38222	8	489	-1	1	3	2016-01-26	38.0
141116	305313	80462	8	30	-1	1	3	2016-01-26	38.0
141117	305313	88295	8	489	-1	1	3	2016-01-26	38.0
141118	305313	155017	8	545	-1	1	3	2016-01-26	38.0

141119 rows × 565 columns

图 15 训练集构建

3.2.2 训练集的划分

我们所使用的数据集属于时间序列数据，对时间序列数据而言，不同时间的数据之间存在相关关系，且过去的数据影响未来的数据取值，未来的数据不会影响过去的数据取值。已知整个原始数据集的有效数据时间跨度是从 2016 年

2 月 1 日到 2016 年 4 月 15 日，为了充分利用这三个月的数据我们将按照数据的时间顺序对数据进行划分用于训练模型。

以一个月为一个时间窗，以十天为一个步长获取一个时间窗内的数据，同时以这个时间窗终止时间段的后五天，客户是否产生购买行为为预测结果，对这一个时间窗中每一行不同用户对于不同的 sku_id 都添加上是否购买的标签属性，即 buy_label。通过上述操作我们成功划分出了五个数据集。具体如下：

```
2016-03-02 00:00:00 2016-03-07 00:00:00
len_df_dataset:  108712
len_df_dataset:  108712
2016-03-12 00:00:00 2016-03-17 00:00:00
len_df_dataset:  181605
len_df_dataset:  181605
2016-03-22 00:00:00 2016-03-27 00:00:00
len_df_dataset:  247687
len_df_dataset:  247687
2016-04-01 00:00:00 2016-04-06 00:00:00
len_df_dataset:  259650
len_df_dataset:  259650
2016-04-11 00:00:00 2016-04-16 00:00:00
len_df_dataset:  254062
len_df_dataset:  254062
```

图 16 训练集的划分

其中 03-02 到 03-07 时间段表示用来产生预测结果的时间段，总共按照划分了出了 5 个时间窗用于训练模型。其中前四个时间窗用来训练模型，第五个时间作为测试集用来验证模型的训练效果。

3.3 模型建立与评价指标

3.3.1 CatBoost 模型建立

首先对复制的数据集中删除了’user_id’和’sku_id’列，这些列不作为模型的输入特征。然后，依次确定变量的类别特征，比如特征中存在分类数据，需要进行特殊处理；有些特征可能在当前数据集中不存在，使用列表推导式筛选实际存在的特征，防止程序报错。最后将目标变量转换为 01 类型，即完成了模型建立前的变量处理

接下来初始化 CatBoost 分类器。其中有以下模型参数：iterations：迭代次数，即弱学习器（决策树）的数量。loss_function：损失函数。task_type：指定使用 CPU 进行训练。depth：树的深度，控制模型的复杂度，防止过拟合。leaf_estimation_method：叶子估计方法，使用 Newton 法可以提高训练速度和精度。one_hot_max_size：当类别特征的类别数量小于等于 2 时，使用 One-Hot 编码，否则使用其他编码方式。

对于具体的初始参数选择，有以下依据：迭代次数决定了模型的训练轮数，即构建的决策树的数量。适当的迭代次数可以确保模型有足够的复杂度来拟合数据，但过多的迭代可能导致过拟合。经过验证，150 次迭代在模型性能和训练时间之间取得了良好的平衡。对于二分类问题，使用 Logloss 是常见选择。树深度为 6 通常能够捕获数据中的主要模式，同时控制模型的复杂性。使用 Newton-Raphson 方法来估计叶节点的值，可以加快模型的收敛速度，提高训练效率，适合于数据量较大的情况。当类别特征的类别数小于等于 2 时，使用独热编码。对于类别数量较多的特征，使用目标平均编码等其他方式，避免因高基数特征导致的维度过高和稀疏矩阵问题。

参数	**设置**	**参数说明**
iterations	num_iterations	定义模型中树的数量，也就是迭代次数。更多的迭代次数可以提高模型的性能，但也可能导致过拟合。
loss_function	Logloss	指定用于优化的损失函数。"Logloss" 适用于二分类问题，衡量预测概率与实际标签之间的差异。
task_type	CPU	指定训练过程使用的硬件类型。"CPU" 表示在中央处理器上进行训练。如果有 GPU 可用，可设置为 "GPU" 以加速训练。

depth	6	决策树的最大深度。较大的深度可以捕捉更多的数据细节，但也增加了过拟合的风险。
leaf_estimation_method	Newton	叶子节点值的估计方法。"Newton" 使用牛顿法进行估计，通常比默认的梯度下降法更快收敛。
one_hot_max_size	2	类别特征进行独热编码的最大类别数阈值。类别数小于或等于该值时进行独热编码，超过则使用其他编码方式（如目标编码）。

表 2　CatBoost 模型具体参数设置

3.3.2 XGBoost 模型建立

首先由于 XGBoost 不原生支持类别特征，所以构造一个名为 encode_categorical_features 的函数，将数据集中的类别特征转换为数值特征。在定义类别特征列表（["cate", "brand", "age", "sex", "a1", "a2", "a3"]）后，筛选存在于数据集中的类别特征进行独热编码。

接着同 CatBoost 模型一样，从数据集中删除 "buy_label"、"user_id"、"sku_id" 列，保留用于训练的特征并且转换目标变量为 01 型。有所不同的是构造了 DMatrix 数据结构，来优化存储和运算速度。

有以下模型参数："learning_rate"：学习率，控制每次迭代的步长，较大的学习率可以加快收敛速度。"gamma"：惩罚项系数，指定了节点分裂所需的最小损失函数下降值，越大越保守。"max_depth"：树的最大深度。"lambda"：L2 正则化项权重。"tree_method"：使用直方图近似算法，加速训练。
"scale_pos_weight"：正样本的权重，可以在数据不平衡时调整。
"objective"：目标函数。"eval_metric"：评价指标。

具体的模型参数设定依据为：将学习率设置为 0.3，较高的学习率加快收敛速度，但可能错过全局最优解，0.3 是平衡二者的结果。最小分裂损失减少设置为 0.6，较大的 gamma 值可以防止过拟合，促使模型生成更简单的树。树的最大深度设定同 CatBoost 一致为 6. L2 正则化权重为 1，对叶子节点权重的 L2 正则化，有助于降低模型的复杂度。由于处理的数据集比较大，所以树构建方法为直方图优化算法，能够加速模型训练。正样本权重比例为 1，因为在数据集分割的窗口处理过程中，已经处理了不平衡问题。目标函数为二分类问题的逻辑回归，输出为概率，用于判断样本属于某一类的可能性。

参数	**设置**	**参数说明**
learning_rate	0.3	控制每棵树对最终结果的贡献。较低的学习率通常需要更多的迭代次数，但能提升模型的泛化能力。
gamma	0.6	节点分裂所需的最小损失函数下降值。较高的值可以使模型更加保守，避免过拟合。
max_depth	6	决策树的最大深度，控制模型的复杂度。深度过大可能导致过拟合，过小则可能欠拟合。
lambda	1	对叶子节点权重的 L2 正则化，帮助降低模型的复杂度，防止过拟合。
tree_method	hist	指定构建树的算法。"hist" 使用直方图近似算法，适用于大规模数据集，能够加速训练过程。
scale_pos_weight	1	用于调整类别不平衡问题。值为正类样本数与负类样本数的比例。如果数据集正负样本平衡，则设为 1。

predictor	cpu_predictor	指定用于预测和训练的硬件。"gpu_predictor" 表示使用 GPU 加速。如果没有 GPU，可设置为 "cpu_predictor"。
objective	binary:logistic	定义优化目标。"binary:logistic" 适用于二分类问题，输出为每个样本属于正类的概率。
eval_metric	logloss	指定用于评估模型性能的指标。"logloss" 是对数损失，适用于二分类问题，衡量预测概率与实际标签的差异。

表 3　XGBoost 模型具体参数设置

3.3.3 评价指标

建立 calculate_test_score 用于评估分类模型在用户购买预测任务中的性能，特别是针对用户级别和商品（SKU）级别的预测，计算出精细化的评价指标。

1. 用户级别的指标计算过程：首先按照用户汇总预测结果：对 results 按 user_id 进行分组，求和 buy_label 和 predict_label。结果 user_grouped 包含每个用户的总真实购买次数和总预测购买次数。接着计算用户级别的混淆矩阵四个值，即真正例(TP)、假正例（FP)、真负例（TN)、假负例（FN)。接着基于上述指标，计算精度和召回率。最后为了强调召回率的重要性，对传统的 F1 分数进行了调整，相对于传统 F1 分数，更加侧重于召回率（分子中召回率的系数较大)。

$$\text{user\_precision} = \frac{\text{user\_tp}}{\text{user\_tp}+\text{user\_fp}}$$

图 17　精度计算公式

$$\text{user\_recall} = \frac{\text{user\_tp}}{\text{user\_tp} + \text{user\_fn}}$$

图 18　召回率计算公式

$$\text{f1\_user} = \frac{6 \times \text{user\_recall} \times \text{user\_precision}}{5 \times \text{user\_recall} + \text{user\_precision}}$$

图 19　用户级别调整的 F1 计算公式

2. SKU 级别的评价指标计算过程：构建混淆矩阵四个值的过程与上述相同，唯一区别在于 F1 的构建不同。在该指标中，更加强调精度而非召回率。

$$\text{f1\_sku} = \frac{5 \times \text{sku\_recall} \times \text{sku\_precision}}{2 \times \text{sku\_recall} + 3 \times \text{sku\_precision}}$$

图 20　SKU 级别调整的 F1 计算公式

3. 综合得分的构建：将用户级别 F1 分数和 SKU 级别 F1 分数按照权重进行加权求和。其中，SKU 级别的指标权重更高，反映了 SKU 级别预测的重要性。

$$\text{final\_score} = 0.4 \times \text{f1\_user} + 0.6 \times \text{f1\_sku}$$

图 21　综合得分的构建

综合而言，该评价指标定义过程全面考虑了模型在不同层面的性能，包括用户和商品两个维度。调整了 F1 分数的计算，以符合特定的业务关注点。通过加权平均，得到一个综合的最终得分。

4 模型实例化及评估

4.1 模型测试过程

函数包含三个参数，测试数据集、保存的模型文件路径以及分类阈值（用于将预测概率转换为二进制标签，将默认值设为 0.5）。

模型实例化和测试的过程很常规。首先准备测试数据，接着 pickle 加载已经训练并且保存的模型文件，然后对测试特征进行预测，获取预测概率和二进制标签。评估模型性能的部分，除了上文自定义的评价函数，还使用了 AUC 和 ROC 来评估。

特别说明阈值选择，threshold 设置为 0.25，意味着当预测概率大于 0.25 时，即认为样本属于正类。这是基于模型在验证集上的最优表现而选择的值。

两模型的测试过程有些不同之处。在 XGBoost 中，需要将特征数据转换为 DMatrix 格式。预测方法中，CatBoost 模型直接获取每个样本属于正类的概率，而 XGBoost 的 predict 方法返回的是样本属于正类的概率值。预测标签生成中 XGBoost 模型使用列表推导式根据阈值生成预测标签。同时，基于模型在验证集上的性能调优结果，CatBoost 模型、XGBoost 模型的阈值设置分别为 0.25 和 0.165.

4.2 模型结果叙述

4.2.1 用户级别指标

指标	CatBoost	XGBoost
真正例 (TP)	785	847
假正例 (FP)	624	771
假负例 (FN)	192	130
真负例 (TN)	17,556	17,409
精确率 (Precision)	0.5571	0.5235

召回率 **(Recall)**	0.8035	0.8669
F1 分数	0.5871	0.5605

4.2.2 SKU 级别指标

指标	CatBoost	XGBoost
真正例 **(TP)**	745	812
假正例 **(FP)**	782	1,020
假负例 **(FN)**	235	168
真负例 **(TN)**	252,300	252,062
精确率 **(Precision)**	0.4879	0.4432
召回率 **(Recall)**	0.7602	0.8286
F1 分数	0.6215	0.6148

4.2.3 综合评分

指标	CatBoost	XGBoost
综合评分	0.6077	0.5931

F1 分数来看，CatBoost 在用户级别和 SKU 级别的 F1 分数均高于 XGBoost，显示出更好的综合性能。

精度与召回率来看 CatBoost 在精确率方面表现更好，而 XGBoost 在召回率方面更优。根据具体业务需求，若更注重减少误报（假正例），CatBoost 更适合；若更注重不漏掉任何潜在的购买用户或商品，XGBoost 可能更有优势。

如果业务更重视召回率，例如希望尽可能多地识别潜在购买用户或商品，XGBoost 可能更合适。如果业务更重视精确率，例如希望减少误判，以降低后续处理成本，CatBoost 可能更优。

5 特征重要性

5.1 模型选择

在我们的研究中，所得到的数据集包括巨量的销售数据、产品信息、用户数据以及用户评论，各个子数据集之间存在紧密的联系，如果使用简单的数据处理和分析方法，将很难得到有意义的结果。

由于 CatBoost 模型得到了相比 XGBoost 更优的结果，并且其含的 get_feature_importance 方法可以满足特征值重要性排序的要求，我们使用该方法进行。

5.2 特征重要性计算及选择

为了更好地优化特征选取，识别并移除那些对模型预测影响较小的特征，从而简化模型并减少过拟合的风险，提高模型解释性，我们进行了特征重要性计算及选择，具体步骤包括：

①特征重要性计算：使用 CatBoost 模型的 get_feature_importance 方法计算每个特征的重要性分数，部分结果如下：

```
browse_cnt_user_brand_all  :  4.918621665875691
click_rate_user_sku_all  :  4.276246011288491
browse_rate_user_30  :  3.89501469238986
follow_rate_user_brand_all  :  3.612541058833821
browse_cnt_user_sku_5  :  3.5538306096483967
buy_cnt_user_30  :  3.3954911752694366
addcart rate user sku all  :  2.616991029449585
```

图 22　特征重要性计算

②特征重要性排序：将特征按照它们的重要性分数进行排序。

③特征剔除：通过设置一个阈值（1e-3），剔除那些重要性分数低于该阈值的特征，结果如下：

```
# 得到待删除的特征
def features_to_pop(feature_importance_list):
    feat_pop = []
    for f, v in feature_importance_list:
        if v < 1e-3:
            feat_pop.append(f)
    return feat_pop

feat_pop = features_to_pop(feature_importance_list)
len(feat_pop)
```

```
68
```

图 23　特征剔除

6 成员分工

小组成员	负责工作
邹宇涛	特征工程、模型训练
卓翔	数据预处理、原始数据可视化分析