

赛题背景:	2
赛题任务:	2
字段表:	2
数据探索:	3
数据集划分:	8
特征工程.....	8

生活大实惠：O2O 优惠券使用预测

赛题背景：

随着移动设备的完善和普及，移动互联网+各行各业进入了高速发展阶段，这其中以 O2O (Online to Offline) 消费最为吸引眼球。据不完全统计，O2O 行业估值上亿的创业公司至少有 10 家，也不乏百亿巨头的身影。O2O 行业天然关联数亿消费者，各类 APP 每天记录了超过百亿条用户行为和位置记录，因而成为大数据科研和商业化运营的最佳结合点之一。以优惠券盘活老用户或吸引新客户进店消费是 O2O 的一种重要营销方式。然而随机投放的优惠券对多数用户造成无意义的干扰。对商家而言，滥发的优惠券可能降低品牌声誉，同时难以估算营销成本。个性化投放是提高优惠券核销率的重要技术，它可以让具有一定偏好的消费者得到真正的实惠，同时赋予商家更强的营销能力。本次大赛为参赛选手提供了 O2O 场景相关的丰富数据，希望参赛选手通过分析建模，精准预测用户是否会在规定时间内使用相应优惠券。

赛题任务：

通过分析线上和线下的数据分析，预测用户在 2016 年 7 月领取优惠券后 15 天以内的使用情况。

提供数据：**ccf_offline_stagel_test_revised.csv**,
ccf_offline_stagel_train.csv, **ccf_online_stagel_train.csv**,
sample_submission.csv

字段表：

Table 1: 用户线下消费和优惠券领取行为(ccf_offline_stagel_train.csv)

Field	Description
User_id	用户 ID
Merchant_id	商户 ID
Coupon_id	优惠券 ID：null 表示无优惠券消费，此时 Discount_rate 和 Date_received 字段无意义
Discount_rate	优惠率：x \in [0,1] 代表折扣率；x:y 表示满 x 减 y。单位是元
Distance	user 经常活动的地点离该 merchant 的最近门店距离是 x*500 米（如果是连锁店，则取最近的一家门店），x \in [0,10]；null 表示无此信息，0 表示低于 500 米，10 表示大于 5 公里；
Date_received	领取优惠券日期
Date	消费日期：如果 Date=null & Coupon_id != null，该记录表示领取优惠券但没有使用，即负样本；如果 Date!=null & Coupon_id = null，则表示普通消费日期；如果 Date!=null & Coupon_id != null，则表示用优惠券消费日期，即正样本；

Table 2: 用户线上点击/消费和优惠券领取行为(cf_online_stagel_train.csv)

Field	Description
User_id	用户 ID
Merchant_id	商户 ID
Action	0 点击，1 购买，2 领取优惠券
Coupon_id	优惠券 ID：null 表示无优惠券消费，此时 Discount_rate 和 Date_received 字段无意义。“fixed”表示该交易是限时低价活动。
Discount_rate	优惠率：x \in [0,1] 代表折扣率；x:y 表示满 x 减 y；“fixed”表示低

	价限时优惠;
Date_received	领取优惠券日期
Date	消费日期: 如果 Date=null & Coupon_id != null, 该记录表示领取优惠券但没有使用; 如果 Date!=null & Coupon_id = null, 则表示普通消费日期; 如果 Date!=null & Coupon_id != null, 则表示用优惠券消费日期;

Table 3: 用户 020 线下优惠券使用预测样本(ccf_offline_stagel_test_revised.csv)

Field	Description
User_id	用户 ID
Merchant_id	商户 ID
Coupon_id	优惠券 ID: null 表示无优惠券消费, 此时 Discount_rate 和 Date_received 字段无意义
Discount_rate	优惠率: $x \in [0, 1]$ 代表折扣率; $x:y$ 表示满 x 减 y 。单位是元
Distance	user 经常活动的地点离该 merchant 的最近门店距离是 $x*500$ 米 (如果是连锁店, 则取最近的一家门店), $x \in [0, 10]$; null 表示无此信息, 0 表示低于 500 米, 10 表示大于 5 公里;
Date_received	领取优惠券日期

Table 4: 选手提交文件字段, 其中 user_id, coupon_id 和 date_received 均来自 Table 3, 而 Probability 为预测值

Field	Description
User_id	用户 ID
Merchant_id	商户 ID
Coupon_id	优惠券 ID: null 表示无优惠券消费, 此时 Discount_rate 和 Date_received 字段无意义
Date_received	领取优惠券日期
Probability	15 天内用券概率, 由参赛选手给出

数据探索:

首先对所给的数据进行预处理分析, 对给予的几个基本数据集进行分析。

1. 对于 ccf_online_stagel_train.csv 文件中给出的用户线上消费用户券的情况, 如下:

```

Main.py × ccf_online_stage1_train.csv × ccf_offline_stage1_train.csv × sample_submission.csv × ccf_offline_st
1 User_id,Merchant_id,Action,Coupon_id,Discount_rate,Date_received,Date
2 13740231,18907,2,100017492,500:50,20160513,null
3 13740231,34805,1,null,null,null,20160321
4 14336199,18907,0,null,null,null,20160618
5 14336199,18907,0,null,null,null,20160618
6 14336199,18907,0,null,null,null,20160618
7 14336199,18907,0,null,null,null,20160618
8 14336199,18907,0,null,null,null,20160618
9 14336199,18907,0,null,null,null,20160618
10 14336199,18907,0,null,null,null,20160618
11 14336199,18907,0,null,null,null,20160618
12 14336199,38810,0,null,null,null,20160126
13 14336199,38810,0,null,null,null,20160126
14 14336199,38810,0,null,null,null,20160126
15 14336199,38810,0,null,null,null,20160126
16 14336199,18907,0,null,null,null,20160127
17 14336199,18907,0,null,null,null,20160127
18 14336199,37005,0,null,null,null,20160412
19 14336199,14305,0,null,null,null,20160127
20 14336199,18907,0,null,null,null,20160618
21 10539231,12008,1,null,null,null,20160618
22 10539231,31904,0,null,null,null,20160107

```

从文件的 title 可知该数据集共有 8 个属性:User_id, Merchant_id, Action, Coupon_id, Discount_rate, Date_received, Date。其中数据类型如下:

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11429826 entries, 0 to 11429825
Data columns (total 7 columns):
User_id          int64
Merchant_id      int64
Action           int64
Coupon_id        object
Discount_rate     object
Date_received     float64
Date             float64
dtypes: float64(2), int64(3), object(2)
memory usage: 610.4+ MB

```

发现 on_line 训练集中共有 11429826 个样本, 现对样本中的八个属性做缺省分析。通过 groupby 语句, 对八个属性按属性值分类处理(类似决策树), 因为 groupby 过滤了空值, 所以可以间接得到每个属性的缺失数量。代码如下:

```

5 import pandas as pd
6 import numpy as np
7
8 # from sklearn import preprocessing
9
10 def get_dict(data, split_line):
11     dict = data.groupby(split_line).groups
12     length = sum(map(lambda x: len(x), dict.values()))
13     if len(data) != length:
14         print(split_line + ": 分组有异常数据", "缺失数量", len(data) - length)
15     return dict
16
17 data = pd.read_csv("/home/ubuntu/PycharmProjects/TianChi/aliset/ccf_online_stage1_train.csv")
18 # print(data.columns)
19 print("训练集中共有数据元组: ", len(data))
20 # print(data.info())
21 # print(data.describe())
22 super_dict = {}
23 for column in data.columns:
24     super_dict[column] = get_dict(data, column)
25 print(type(super_dict['User_id']))
26

```

结果:

```

/usr/local/bin/python3.6 /home/ubuntu/PycharmProject
训练集中共有数据元组: 11429826
Coupon_id: 分组有异常数据 缺失数量: 10557469
Discount_rate: 分组有异常数据 缺失数量: 10557469
Date_received: 分组有异常数据 缺失数量: 10557469
Date: 分组有异常数据 缺失数量: 655898
<class 'dict'>

Process finished with exit code 0

```

分组有异常数据即表示, 该在 table 中该属性下有为 null 的属性值, 缺失数量便是该属性中 null 属性值的数量。通过分析得知

Coupon_id, Discount_rate, Date_received 的 null 值相同, 现猜想, 三者是否同时为 null。验证代码如下:

```

testItem = data[data.Coupon_id.isna() & data.Discount_rate.isna() & data.Date_received.isna()]
print(len(testItem))

```

```

/usr/local/bin/python3.6 /home/ubuntu/PycharmP
10557469

```

这与上面的缺失数量相吻合, 说明三个属性是同时为 null, 严格来说是 Coupon_id 为 null 时, 则 Discount_rate 和 Date_received 也为 null, 这证明了 Table 栏中所言, “Coupon_id 为 null 表示无用户券消费时, Discount_rate 和 Date_received 字段无意义”。除此外, 发现 Date 项有 65 万之多数据为空。通过分析得知, Date 为空时 Coupon_id 必然不为空, 这与 Table 栏中所言“如果 Date=null & Coupon_id != null, 该记录表示领取优惠券但没有使用”。相吻合。

```

testItem2 = data[data.Date.isna() & data.Coupon_id.notna()]
print(len(testItem2))

```



```
/usr/local/bin/python3.6 /home/ubuntu/Pyc  
655898
```

通过上面的分析，发现 **online** 数据集中的数据都是正常的，虽然也有不少项为 **null**，但是其依然蕴含了某种特殊的信息，并没有真正意义上的异常数据，都是有用的数据，也就不需要了对 **online** 的数据预处理（例如对缺失项按照均值，众数进行填充，或则直接删掉）。

2. 对于 **ccf_offline_stagel_train.csv** 文件中给出的用户线下消费用户券的情况，如下：

```
1 User_id,Merchant_id,Coupon_id,Discount_rate,Distance,Date_received,Date  
2 1439408,2632,null,null,0,null,20160217  
3 1439408,4663,11002,150:20,1,20160528,null  
4 1439408,2632,8591,20:1,0,20160217,null  
5 1439408,2632,1078,20:1,0,20160319,null  
6 1439408,2632,8591,20:1,0,20160613,null  
7 1439408,2632,null,null,0,null,20160516  
8 1439408,2632,8591,20:1,0,20160516,20160613  
9 1832624,3381,7610,200:20,0,20160429,null  
10 2029232,3381,11951,200:20,1,20160129,null  
11 2029232,450,1532,30:5,0,20160530,null  
12 2029232,6459,12737,20:1,0,20160519,null  
13 2029232,6459,null,null,0,null,20160626  
14 2029232,6459,null,null,0,null,20160519  
15 2747744,6901,1097,50:10,null,20160606,null  
16 196342,1579,null,null,1,null,20160606
```

户

从文件的 **title** 可知该数据集共有 7 个属性
性：**User_id,Merchant_id,Coupon_id,Discount_rate,Distance,Date_re
ceived,Date**。其中数据类型如下：

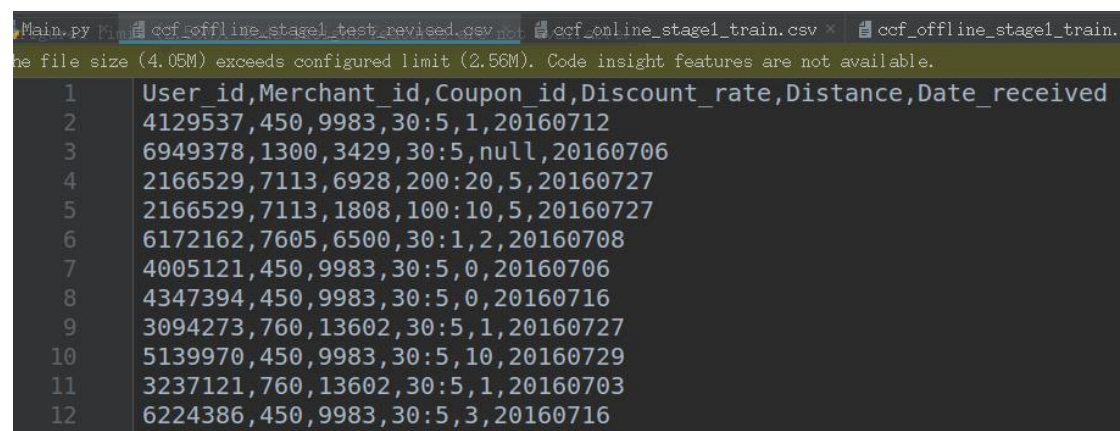
```
训练集中共有数据元组： 1754884  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1754884 entries, 0 to 1754883  
Data columns (total 7 columns):  
User_id          int64  
Merchant_id      int64  
Coupon_id        float64  
Discount_rate    object  
Distance         float64  
Date_received    float64  
Date            float64  
dtypes: float64(4), int64(2), object(1)  
memory usage: 93.7+ MB  
None
```

通过和上面处理 **online** 数据集同样的方式，可得

```
Coupon_id:分组有异常数据 缺失数量： 701602  
Discount_rate:分组有异常数据 缺失数量： 701602  
Distance:分组有异常数据 缺失数量： 106003  
Date_received:分组有异常数据 缺失数量： 701602  
Date:分组有异常数据 缺失数量： 977900
```

发现和 online 数据集一样，Coupon_id 为 null 时，则 Discount_rate 和 Date_received 也为 null。即没有领取到优惠券时，折扣率和优惠券领取时间数据也就不存在，符合现实逻辑。同样 offline 数据集中也存在只要 Date 为空时 Coupon_id 必然不为空的情况。这表明存在不少领取了优惠券但是没有使用的情况。结合 online 数据集，发现这种现象很是奇怪，即只要用户没有消费，那他一定领取了优惠券，从概率上看，样本数据有点不合逻辑。除此外，发现 offline 数据集中 distance 项有 null，即缺少了对应的 user 经常活动的地点离该 merchant 的最近门店的距离，通过分析发现这些缺失的 distance 对应的用户领取的消费券都不为空，有点匪夷所思。难不成时门店的员工的内部消费？但是从 Distance 属性值上来分析 $x \in [0,10]$ 覆盖了所有大于等于 0 的距离，那么员工应该也在这个范围内啊，不是很能理解。

3. 对于 ccf_offline_stage1_train.csv 文件中给出的用户线下消费用户券的情况，如下：



	User_id	Merchant_id	Coupon_id	Discount_rate	Distance	Date_received
1	4129537	450	9983	30:5	1	20160712
2	6949378	1300	3429	30:5	null	20160706
3	2166529	7113	6928	200:20	5	20160727
4	2166529	7113	1808	100:10	5	20160727
5	6172162	7605	6500	30:1	2	20160708
6	4005121	450	9983	30:5	0	20160706
7	4347394	450	9983	30:5	0	20160716
8	3094273	760	13602	30:5	1	20160727
9	5139970	450	9983	30:5	10	20160729
10	3237121	760	13602	30:5	1	20160703
11	6224386	450	9983	30:5	3	20160716
12						

ccf_offline_stage1_test_revised 是要预测的用户在 2016 年 7 月领取优惠券的测试集。其中有 6 个属性

User_id, Merchant_id, Coupon_id, Discount_rate, Distance, Date_received。显然这些与 offline 数据集的 6 集基本属性是一致的，我们的目标便是根据这 6 个属性来预测该用户在领取优惠券 15 天内使用该券的概率。分析得知除了少部分 Distance 存在 null 缺失量，其他数据正常。

总结：以上通过上述简单的分析，发现只有在线下的数据集中 Distance 部分有缺失，而在实际给出的测试集中也有 Distance 为 Null 的情况，也就是说这个 Distance 属性无论缺失与否都对最终模型的建立有着重要的作用，至于其他属性诸如 Coupon_id 之类的其为 null 蕴含这特殊的信息(没有领取到优惠券等等)。所以对于给定的数据集并不需要在缺失项上的填充或删除等处理，这些数据都是有意义的。以上主要针对了给定的数据集数据项是否缺失那一块进行了分析，实际在根据个属性 groupby 中也发现了不少有用的信息，比如说在 Online 数据集中关于 Discount_rate 那一项，发现只有“x: y”型(满 x 减 y)和 fixed(低价限时优惠)，并不存在折扣率。不少用户(User_id)同时出现在 online 和 offline 数据集中(这一点在用户行为分析上十分重要)，而且通过进一步分析发现在 test 数据集中的绝大多数用户(User_id)和绝大部分商家(Merchant_id)都有出现在 online 和 offline 数据集中，优惠券虽然没有出现在 online 和 offline 中，但是其对应的折扣率

(Discount_rate) 有出现过。综上，从主要的两份原始数据集(online 和 offline)中，我们可以从 online 数据集中提取到与用户相关的线上特征(网上浏览习惯(点击率，领取率等等))，而 offline 数据集则可以提取到更加丰富的特征：用户的线下特征(网上浏览习惯，用户领取优惠券并使用的概率，用户平均核销优惠券次数，用户最喜欢消费的工作日等等)，

商家的相关特征（商家优惠券被领取的次数，商家优惠券领取后核销的次数，商家优惠券被核销的平均时间，商家分发的最受欢迎的优惠券.....），用户和商家的交互特征（商家的回头客比例，用户对商家的惠顾率，用户对折扣优惠券的使用比例等等），优惠券的相关特征（优惠券类型，优惠券的领取时间，领取次数最多的优惠券类型，核销率最高的优惠券类型.....）。

数据集划分：

对于这种给定一段时间的数据集要预测未来 n 天的情况的问题，选择使用滑窗法来对原始训练集的划分，这样可以充分利用到原始数据，尤其是在测试集上，可以利用已知的 **Label** 进行验证模型。特征区间划分的越小，得到的训练数据集越多。我的划分方案如下：

	预测区间（提取 Label）	特征区间（提取 Feature）
训练集 1	20160401~20160430	20160101~20160331
训练集 2	20160501~20160531	20160201~20160430
验证集	20160601~20160630	20160201~20160531
测试集	20160701~20160731	20160301~20160531

代码如下：(详见 data_split.py 文件)

```
data_split.py | ccf_offline_stagel_train.csv | ccf_offline_stagel_test_revised.csv | Main.py
1 import pandas as pd
2 import numpy as np
3
4
5 offline_train = pd.read_csv("/home/ubuntu/PycharmProjects/TianChi/aliset/ccf_offline_stagel_train.csv")
6 online_train = pd.read_csv("/home/ubuntu/PycharmProjects/TianChi/aliset/ccf_online_stagel_train.csv")
7 offline_test = pd.read_csv("/home/ubuntu/PycharmProjects/TianChi/aliset/ccf_offline_stagel_test_revised.csv")
8
9 data = offline_train
10
11
12 feature1 = data[((data.Date>=20160101)&(data.Date<=20160331))
13                |((data.Date.isna())&(data.Date_received>=20160101)
14                &(data.Date_received<=20160331))]
15 dataset1 = data[(data.Date_received>=20160401)&(data.Date_received<=20160430)]
16
17 feature2 = data[(data.Date>=20160201)&(data.Date<=20160430)
18                |((data.Date.isna())&(data.Date_received>=20160201)
19                &(data.Date_received<=20160430))]
20 dataset2 = data[(data.Date_received>=20160501)&(data.Date_received<=20160531)]
21
22 feature3 = data[(data.Date>=20160201)&(data.Date<=20160531)
23                |((data.Date.isna())&(data.Date_received>=20160201)
24                &(data.Date_received<=20160531))]
25 dataset3 = data[(data.Date_received>=20160601)&(data.Date_received<=20160630)]
26
27 feature_test = data[(data.Date>=20160301)&(data.Date<=20160531)
28                    |((data.Date.isna())&(data.Date_received>=20160301)
29                    &(data.Date_received<=20160531))]
30 dataset = offline_test
31
```

特征工程

基于以上的数据探索，将特征分组为用户特征，商家特征，用户商家特征，优惠券特征。

一. 用户特征（描述每个用户的消费喜好，可进一步分为线上，线下，线上-线下特征）

● 线下特征

- 用户领取优惠券次数
- 用户线下核销率
- 用户核销过的商家数量
- 用户使用所有优惠券的平均时间
- 用户核销优惠券的平均折扣
- 用户核销率最高的优惠券折扣率
- 用户核销的时间是否为节假日
- 用户核销的平均距离

- 等等户
- 线上特征
 - 用户的记录条数
 - 用户点击率
 - 用户领取率
 - 用户购买率
 - 等等
- 线上-线下特征
 - 用户线下记录的占比数户
 - 用户用户线下核销次数的占比数
 - 用户对同一折扣率优惠券的线下使用占比数
 - 等等
- 二. 商家特征（描述商家的受欢迎程度机器商品的被消费规律）
 - 商家优惠券被核销率
 - 商家优惠券被领取的次数
 - 商家所被消费的优惠券的平均消费折扣率
 - 商家优惠券被核销的平均时间
 - 商家所分发的优惠券的种类数量
 - 等等
- 三. 优惠券特征（描述优惠券自身的特征及其类别的历史消费规律）
 - 优惠券折扣率
 - 优惠券核销率
 - 优惠券类型
 - 优惠券被核销的时间
 - 等等
- 四. 用户-商家特征（描述用户对某些商家的消费偏好）
 - 用户领取该商家的优惠券次数
 - 用户在该商家领取的优惠券的占比
 - 用户在该商家的核销量的总占比
 - 用户在该商家的核销率
 - 等等

部分代码如下：（详见 `feature_extract.py` 文件）

```

1 import pandas as pd
2 import numpy as np
3 from functools import reduce
4
5 """用户相关特征"""
6 # 线上特征
7
8 def user_times(df):
9     """用户记录条数"""
10    Series = pd.Series(1,df.index)
11    Series.name = "Times"
12    frame = df[["User_id"]].join(Series)
13    grouped = frame.groupby("User_id",as_index=False).sum() #统计每个用户领取的优惠券数目
14    df = pd.merge(df,grouped,on="User_id",how="left")
15    return df
16
17 def user_consume_count(df):
18     """用户消费次数"""
19    Series = pd.Series(list(map(lambda x: 1. if x != 'null'
20                                else 0.,df["Date"])))
21    Series.name = "user_consume_count"
22    frame = df[["User_id"]].join(Series)
23    grouped = frame.groupby("User_id",as_index=False).sum()
24    df = pd.merge(df,grouped,on="User_id",how="left")
25
26 def user_received_counts(df):
27     """用户领取到的优惠券数"""
28    frame = df["Coupon_id"].map(lambda x: 1. if x != 'null' else 0.)#每一张非空的优惠券记为1
29    frame.name = "user_received_counts"
30    received_users = df[["User_id"]].join(frame) #按索引连接
31    grouped = received_users.groupby("User_id",as_index=False).sum() #统计每个用户领取的优惠券数目
32    df = pd.merge(df,grouped,on="User_id",how="left")
33
34    return df
35

```