# O2O 线上优惠券使用预测报告

郭锦红

# 一、实训背景与内容

## 1.1.实训背景

本赛题的比赛背景：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。

## 1.2.实训内容

（1） 数据导入及预处理。

（2） 特征构建。

（3） 特征拼接。

（4） 训练样本和测试样本划分

（5） 模型建立和预测（python 部分）

## 1.3.理解数据

本题提供用户在 2016 年 1 月 1 日至 2016 年 6 月 30 日之间真实线上线下消费行为 ccf_offline_test.xlsx，预测用户在 2016 年 7 月领取优惠券后 15 天以内的使用情况。如下图表 1

注意： 为了保护用户和商家的隐私，所有数据均作匿名处理，同时采用了有偏采样和必要过滤。

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 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 | 150:20:00 | 1 | 20160528 | null |
| 4 | 1439408 | 2632 | 8591 | 20:01 | 0 | 20160217 | null |
| 5 | 1439408 | 2632 | 1078 | 20:01 | 0 | 20160319 | null |
| 6 | 1439408 | 2632 | 8591 | 20:01 | 0 | 20160613 | null |
| 7 | 1439408 | 2632 | null | null | 0 | null | 20160516 |
| 8 | 1439408 | 2632 | 8591 | 20:01 | 0 | 20160516 | 20160613 |
| 9 | 1832624 | 3381 | 7610 | 200:20:00 | 0 | 20160429 | null |
| 10 | 2029232 | 3381 | 11951 | 200:20:00 | 1 | 20160129 | null |
| 11 | 2029232 | 450 | 1532 | 30:05:00 | 0 | 20160530 | null |
| 12 | 2029232 | 6459 | 12737 | 20:01 | 0 | 20160519 | null |
| 13 | 2029232 | 6459 | null | null | 0 | null | 20160626 |
| 14 | 2029232 | 6459 | null | null | 0 | null | 20160519 |
| 15 | 2747744 | 6901 | 1097 | 50:10:00 | null | 20160606 | null |
| 16 | 196342 | 1579 | null | null | 1 | null | 20160606 |
| 17 | 196342 | 1579 | 10698 | 20:01 | 1 | 20160606 | null |
| 18 | 2223968 | 3381 | 9776 | 10:05 | 2 | 20160129 | null |
| 19 | 73611 | 2099 | 12034 | 100:10:00 | null | 20160207 | null |
| 20 | 163606 | 1560 | 5054 | 200:20:00 | 10 | 20160421 | null |

| 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； |

# 二、实训步骤

## 2.1.数据预处理

### 2.1.1.根据时间间隔将用户分成三类

在原表中添加 label（字符），根据时间间隔将用户分成三类：-1，0，1（-1 代表普通用户，1 代表正样本，0 代表负样本）

正样本 1：领了优惠券并在 15 天内使用的样本。

负样本 0: 领了优惠券但在 15 天内没有使用的样本和领了但在 15 天后才使用的样本。

其它-1

### 2.1.2.对 Distance 字段进行缺失值填充

对 Distance 字段进行缺失值填充(用平均值填充)

## 2.2.特征构建

- 用户与商家的距离；（根据 label 分组，初步判断用户消费与商家的平均距离有关）
- **注：**被使用优惠券都是指在发放后 15 天内被使用。

### 2.2.1.优惠券相关特征（coupon)

| Coupon_id | Discount_rate | Discount | Coupon_used(num) | Coupon_received(num) | Coupon_popu |
|---|---|---|---|---|---|
| 0 | null | (Null) | 0 | 0 | (Null) |
| 3 | 50:1 | 0.98 | 3 | 0 | 0.0000 |
| 4 | 0.85 | 0.85 | 2 | 0 | 0.0000 |
| 6 | 10:1 | 0.9 | 2 | 0 | 0.0000 |
| 10 | 20:1 | 0.95 | 8 | 6 | 0.7500 |
| 12 | 10:1 | 0.9 | 3 | 0 | 0.0000 |
| 15 | 0.95 | 0.95 | 2 | 1 | 0.5000 |

- 优惠券的折扣率（Discount），将满减优惠改写成折扣率形式（300：30 等价于 0.9 折，定义函数完成）
- 优惠券流行度（Coupon_popu），根据 Coupon_id 分组，算出每组的（被使用优惠券/优惠券总数）

### 2.2.2.商家相关特征（merchant)

| Merchant_id | Merch_coupon_used(num | Merch_coupon_grant(nun | Merchant_popu |
|---|---|---|---|
| 2 | 0 | 2 | 0.0000 |
| 5 | 0 | 8 | 0.0000 |
| 11 | 0 | 1 | 0.0000 |
| 12 | 0 | 1 | 0.0000 |
| 14 | 1 | 3 | 0.3333 |
| 15 | 1 | 8 | 0.1250 |
| 17 | 2 | 12 | 0.1667 |

- 商户流行度（Merchant_popu）：根据 Merchant_id 分组，算出各商户所发放优惠券的受欢迎程度（被使用优惠券/发放优惠券总数）

### 2.2.3.用户线下相关特征（user_offline)

| User_id | number_received_coupon | number_used_coupon |
|---|---|---|
| 947 | 6 | 1 |
| 1302 | 1 | 1 |
| 1536 | 1 | 1 |
| 2227 | 2 | 2 |
| 2266 | 7 | 3 |
| 4135 | 1 | 1 |

- 用户领取的优惠券数量（number_received_coupon）根据 User_id 分组,算出每组的优惠券数量

- 用户消费过的优惠券数量（number_used_coupon）根据 User_id 分组,算出每组 label=1 的数量

#### 2.2.4.用户-商家交互特征(user_merchant)

| User_id | Merchant_id | user_merchant_received_coupon | user_merchant_used_coupon | user_merchant_cus |
|---|---|---|---|---|
| 739236 | 7019 | 2 | 1 | 2 |
| 741027 | 775 | 10 | 1 | 10 |
| 741249 | 5341 | 7 | 3 | 7 |
| 741597 | 1041 | 12 | 1 | 11 |
| 741666 | 6485 | 2 | 1 | 2 |
| 744678 | 3621 | 3 | 1 | 1 |

- 用户在商家使用优惠券的次数(user_merchant_used_coupon)根据 User_id 和 Merchant_id 分组，算出每组 label=1 的个数
- 用户在商家领取的优惠券数（user_merchant_received_coupon）根据 User_id 和 Merchant_id 分组，算出每组 Coupon_id 的个数
- 用户在商家消费的次数(user_merchant_cus)根据 User_id 和 Merchant_id 分组，算出每组 Date 的个数

### 2.3.特征拼接

拼接后的表为 all_feature

（1） 拼接特征 2：折扣率（discount）

（2） 拼接特征 3（优惠券流行度 Coupon_popu）

（3） 拼接特征 4（商户流行度 Merchant_popu）

（4） 拼接特征 5（用户领取的优惠券数量 number_received_coupon）

（5） 拼接特征 6（用户消费的优惠券数量 number_used_coupon）

（6） 拼接特征 7（用户在商家使用优惠券的次数 user_merchant_used_coupon）

（7） 拼接特征 8（user_merchant_received_coupon）

（8） 拼接特征 9 (user_merchant_cus)

### 2.4.训练样本和测试样本数据划分

用 sklearn 的 train_test_split 方法划分

## 2.5.高斯朴素贝叶斯模型

选取高斯朴素贝叶斯模型（Gaussian Naive Bayes），利用 python 进行模型训练，查看效果