深圳大学考试答题纸

(以论文、报告等形式考核专用)

二○一八～二○一九学年度第二学期

课程编号	2706019	课程名称	数据仓库与数据挖掘	主讲教师	陈小军	评分	

学　　号	1810273007	姓名	孙文斌	专业年级	软件工程

教师评语：

项目名称：　　ofo 优惠券使用概率预测

摘　要：本次比赛的问题是一个分类问题，首先对数据集进行处理，提取出相关的特征值，然后对数据进行标签处理，使用分类器模型进行训练，最后使用训练好的模型对测试集进行测试，得到不同用户会使用优惠券进行购物的概率。

深圳大学课程项目报告

课程名称：数据挖掘导论

项目名称：ofo 优惠券使用概率预测

学　　院：计算机与软件学院

专　　业：软件工程

任课教师：陈小军

报 告 人：孙文斌　学号：1810273007

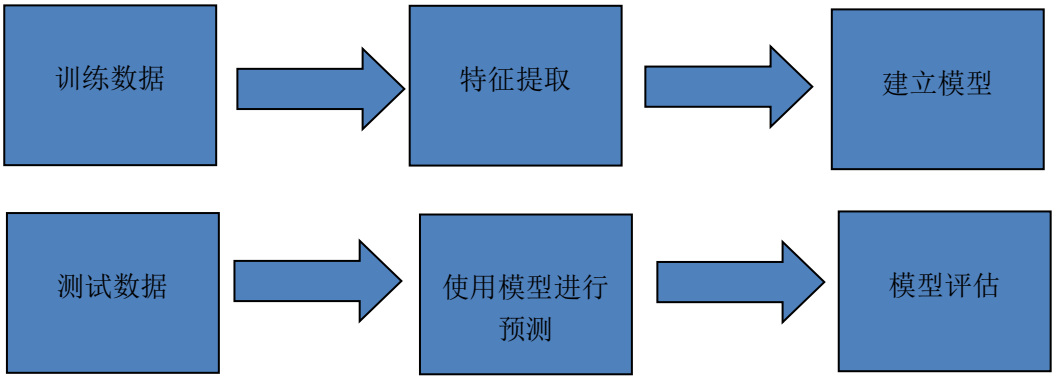
提交时间：2019.6.15

教 务 处 制

一、 项目介绍

本赛题的比赛背景是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。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（15 天）内使用相应优惠券。

从机器学习模型的角度来说，这是一个典型的分类问题，其过程就是根据已有训练集进行训练，得到的模型再对测试进行测试并分类。整个过程如下图所示：



二、 数据统计

对于数据挖掘模型来讲，数据集是我们首先需要理解清楚的，所以接下来我们首先对数据集进行一个分析。本赛题提供用户在 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，预测用户在 2016 年 7 月领取优惠券后 15 天以内的使用情况。 总共有四个文件，分别是

1) ccf_offline_stage1_test_revised.csv
2) ccf_offline_stage1_train.csv
3) ccf_online_stage1_train.csv
4) sample_submission.csv

其中，第 2 个是线下训练集，第 1 个是线下测试集，第 3 个是线上训练集（不会用到），第 4 个是预测结果提交到官网的文件格式。也就是说我们使用第 2 个文件来训练模型，对第 1 个文件进行预测，得到用户在 15 天内使用优惠券的概率值。接下来，对 2、1、4 文件中字段进行列举，字段解释如下图所示。

ccf_offline_stage1_train.csv：用户线下消费和优惠券领取行为

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；

ccf_offline_stage1_test_revised.csv：用户 020 线下优惠券使用预测样本

Field	Description
User_id	用户ID
Merchant_id	商户ID
Coupon_id	优惠券ID
Discount_rate	优惠率：x \in [0,1]代表折扣率；x:y表示满x减y.
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
Coupon_id	优惠券ID
Date_received	领取优惠券日期
Probability	15天内用券概率，由参赛选手给出

首先导入以上三个数据集，显示训练数据集前五行，对数据集的构成有一个初步的认识。

```
dfoff = pd.read_csv('ccf_offline_stage1_train.csv')
dftest = pd.read_csv('ccf_offline_stage1_test_revised.csv')
print(dfoff.head(5))
```

```
   User_id  Merchant_id  Coupon_id Discount_rate  Distance  Date_received  \
0  1439408         2632        NaN           NaN       0.0            NaN
1  1439408         4663    11002.0        150:20       1.0     20160528.0
2  1439408         2632     8591.0          20:1       0.0     20160217.0
3  1439408         2632     1078.0          20:1       0.0     20160319.0
4  1439408         2632     8591.0          20:1       0.0     20160613.0

        Date
0  20160217.0
1         NaN
2         NaN
3         NaN
4         NaN
```

接下来，我们来做个简单统计，看一看究竟用户是否使用优惠券消费的情况，这里我们根据 Date_recieved 和 Date 的数据根据他们是否同时为 nan 来区分用户是否使用了优惠券消费。

```
print('有优惠卷,购买商品:%d' % dfoff[(dfoff['Date_received'].notnull())
& (dfoff['Date'].notnull())].shape[0])
print('有优惠卷,未购商品:%d' % dfoff[(dfoff['Date_received'].notnull())
& (dfoff['Date'].isnull())].shape[0])
print('无优惠卷,购买商品:%d' % dfoff[(dfoff['Date_received'].isnull())
& (dfoff['Date'].notnull())].shape[0])
print('无优惠卷,未购商品:%d' % dfoff[(dfoff['Date_received'].isnull())
& (dfoff['Date'].isnull())].shape[0])
```

```
有优惠卷，购买商品：75382
有优惠卷，未购商品：977900
无优惠卷，购买商品：701602
无优惠卷，未购商品：0
```

可见，很多人（701602）购买商品却没有使用优惠券，也有很多人（977900）有优惠券但却没有使用，真正使用优惠券购买商品的人（75382）很少！所以，这个比赛的意义就是把优惠券送给真正可能会购买商品的人。

三、 建模过程

1、特征提取

（1）折扣率：

一般情况下优惠得越多，用户就越有可能使用优惠券。所以首先我们想到对折扣率进行数据处理。

```
print(dfoff["Discount_rate"].unique())
```

```
[nan '150:20' '20:1' '200:20' '30:5' '50:10' '10:5' '100:10' '200:30'
 '20:5' '30:10' '50:5' '150:10' '100:30' '200:50' '100:50' '300:30'
 '50:20' '0.9' '10:1' '30:1' '0.95' '100:5' '5:1' '100:20' '0.8' '50:1'
 '200:10' '300:20' '100:1' '150:30' '300:50' '20:10' '0.85' '0.6' '150:50'
 '0.75' '0.5' '200:5' '0.7' '30:20' '300:10' '0.2' '50:30' '200:100'
 '150:5']
```

观察处理结果我们发现结果中有很多不同的表示折扣的方式，我们统一将他们表示成一个[0,1]之间的小数，同时根据折扣类型将折扣类型分为满减折扣和折扣率的形式，进一步如果是满减类型，那我们在提取出“满”的部分和“减”的部分，对于空值，我们用 np.nan 代替，所以通过折扣率我们可以得到四个特征，下面是提取四个特征值的函数。

```
def getDiscountType(row):
    if pd.isnull(row):
        return np.nan

    elif ':' in row:
        return 1
    else:
        return 0

def convertRate(row):
    """Convert discount to rate"""
    if pd.isnull(row):
        return 1.0
    elif ':' in str(row):
        rows = row.split(':')
        return 1.0 - float(rows[1])/float(rows[0])
    else:
        return float(row)

def getDiscountMan(row):
    if ':' in str(row):
        rows = row.split(':')
        return int(rows[0])
    else:
        return 0

def getDiscountJian(row):
```

```
    if ':' in str(row):
        rows = row.split(':')
        return int(rows[1])
    else:
        return 0
```

（2）距离：

距离字段表示用户与商店的地理距离，很显然距离的远近是消费者是否会使用优惠券消费的重要因素，所以我们需要把距离作为一个特征值。

显示所以距离的值

```
print(dfoff["Distance"].unique())
```

```
[ 0.  1. nan  2. 10.  4.  7.  9.  3.  5.  6.  8.]
```

我们将所有的空缺值使用-1 代替，同时由于这里字符是字符类型，我们转化为整数类型

```
df['distance'] = df['Distance'].fillna(-1).astype(int)
```

（3）领券日期：领券日期同时也是一个影响消费者是否会去使用优惠券消费的重要的特征，首先我们将日期转化周一到周末，然后因为周末领券的人可能会去消费的概率大一些，所以我们再将周六周天区分出来作为一个特征值，最后将星期特征转化为 one-hot 向量。所以我们构建的特征如下：

weekday : {null, 1, 2, 3, 4, 5, 6, 7}

weekday_type : {1, 0}（周六和周日为 1，其他为 0）

Weekday_1 : {1, 0, 0, 0, 0, 0, 0}

Weekday_2 : {0, 1, 0, 0, 0, 0, 0}

Weekday_3 : {0, 0, 1, 0, 0, 0, 0}

Weekday_4 : {0, 0, 0, 1, 0, 0, 0}

Weekday_5 : {0, 0, 0, 0, 1, 0, 0}

Weekday_6 : {0, 0, 0, 0, 0, 1, 0}

Weekday_7 : {0, 0, 0, 0, 0, 0, 1}

代码如下：

转化星期

```
def getWeekday(row):
    if row == 'nan':
        return np.nan
    else:
        return date(int(row[0:4]), int(row[4:6]),
int(row[6:8])).weekday() + 1
```

weekday_type ： 周六和周日为 1，其他为 0

```
dfoff['weekday_type'] = dfoff['weekday'].apply(lambda x : 1 if x in [6,7]
else 0 )
dftest['weekday_type'] = dftest['weekday'].apply(lambda x : 1 if x in
[6,7] else 0 )
print(dfoff['weekday_type'])
```

将星期转化为 one-hot 编码

```
weekdaycols = ['weekday_' + str(i) for i in range(1,8)]
tmpdf = pd.get_dummies(dfoff['weekday'].replace('nan', np.nan))
tmpdf.columns = weekdaycols
dfoff[weekdaycols] = tmpdf

tmpdf = pd.get_dummies(dftest['weekday'].replace('nan', np.nan))
tmpdf.columns = weekdaycols
dftest[weekdaycols] = tmpdf
```

2、标注标签

有了特征之后，我们还需要对训练样本进行 label 标注，即确定哪些是正样本（y = 1），哪些是负样本（y = 0）。我们要预测的是用户在领取优惠券之后 15 之内的消费情况。所以，总共有三种情况：

领券日期 Date_received	消费日期 Date	y
Nan	Nan	没有领到优惠券，y=-1
！Nan	！Nan	领到优惠券并使用， 正样本 y=1
！Nan	Nan	领了优惠券没有使用 负样本 y=0

定义标签函数

```
def label(row):
    if pd.isnull(row['Date_received']):
        return -1
#15天内使用优惠券 标记为1
    if pd.notnull(row['Date']):
        td = pd.to_datetime(row['Date'], format='%Y%m%d') -
pd.to_datetime(row['Date_received'], format='%Y%m%d')
        if td <= pd.Timedelta(15, 'D'):
            return 1
    return 0
```

查看正负样本数

```
print(dfoff['label'].value_counts())
```

```
 0    988887
-1    701602
 1     64395
```

3、建立模型

接下来就是最主要的建立机器学习模型了。首先确定的是我们选择的特征是上面提取的 14 个特征，为了验证模型的性能，需要划分验证集进行模型验证，划分方式是按照领券日期，即训练集：20160101-20160530，验证集：20160531-20160630。我们采用的模型是的 SGDClassifier。

```
df = dfoff[dfoff['label'] != -1].copy()
train = df[(df['Date_received'] < 20160530)].copy()
valid = df[(df['Date_received'] >= 20160530) & (df['Date_received']
<= 20160630)].copy()


original_feature = ['discount_rate','discount_type','discount_man',
'discount_jian','distance', 'weekday', 'weekday_type'] + weekdaycols
print("----train-----")
model = SGDClassifier(
    loss='log',#损失函数，线性回归
    penalty='elasticnet',
    fit_intercept=True,
    max_iter=100,
    shuffle=True,
    alpha = 0.01,
    l1_ratio = 0.01,
    n_jobs=1,
```

```
    class_weight=None
)
```

4、模型训练

```
model.fit(train[original_feature], train['label'])
```

验证集的验证结果

```
print("验证集的验证结果")
print(model.score(valid[original_feature], valid['label']))
```

```
0.9285003512058067
```

保存模型

```
with open('1_model.pkl', 'wb') as f:
    pickle.dump(model, f)
with open('1_model.pkl', 'rb') as f:
    model = pickle.load(f)
```

四、 模型评估

训练完模型后，我们使用训练好的模型进行对测试集预测，输出用户可能使用的优惠券的概率。

```
y_test_pred = model.predict_proba(dftest[original_feature])
print(y_test_pred)
dftest1 = dftest[['User_id','Coupon_id','Date_received']].copy()
dftest1['Probability'] = y_test_pred[:,1]
dftest1.to_csv('submit1.csv', index=False, header=False)
print(dftest1[['User_id','Probability']])
```

	User_id	Probability
0	4129537	0.107092
1	6949378	0.145981
2	2166529	0.002662
3	2166529	0.014341
4	6172162	0.074920
5	4005121	0.121374
6	4347394	0.109178
7	3094273	0.100426
8	5139970	0.014480
9	3237121	0.082469
10	6224386	0.060757
11	6488578	0.129230

五、 总结

总分（100 分）：

具体要求如下：

1) 给出比赛任务，数据以及目标函数的描述，以及对该问题进行简单的归类（聚类，分类或者回归等）。**（15 分）**

2）对数据实施基本的数据统计和可视化分析**（20 分）**

3）写出比赛的思路，包括数据清洗，特征构造，模型选择（聚类，分类或者回归等），需要指出自己完成的部分。**（30 分）**

4) 算法优缺点分析，根据比赛结果，给出优化方案的思考**（20 分）**

5）给出最终比赛排名，谈谈自身的收获**（15 分）**

注意事项：

1. **项目设计和项目报告由个人独立完成。如存在抄袭行为，考试成绩为零分。**
2. **截止时间：2019 年 6 月 26 日 23：59。**
3. **在截止时间之前需要提交的内容：**
   a) **提交本《项目报告》电子版；6 月 26 日 23：59 前提交到 blackboard 系统。**
   b) **提交本《项目报告》打印版；6 月 27 日上课时交给老师。（第一页《答题纸》单面打印，其他内容双面打印）。**
   c) **逾期提交无成绩（以 blackboard 系统的电子版提交时间为准）**

二、项目知识点

1. 数据预处理；
2. 数据可视化；
3. 特征工程；
4. 模型选择；
5. 实验结果的评价；

三、实验过程

（请在此处描述你的关键代码，解释代码的作用，遇到的错误以及解决方法等）

四、项目结果

（请在此处描述你的实验结果，采用图表的形式，并加以说明）

五、项目总结

（此处写写你的所想、所得）