拼多多优惠券使用行为预测

本周你的任务就是根据用户的基本信息以及过去的消费行为数据，完成以下事项：

- 使用 Python 建立逻辑回归模型
- 预测用户是否会在活动中使用优惠券
- 找到对用户使用优惠券影响较大的因素


```
# 搭建环境
# 导入 pandas 与 numpy 工具包。
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.colors import ListedColormap
#引入 sklearn 包
from sklearn.linear_model import LinearRegression
%matplotlib inline

#中文字体正常显示
plt.rcParams['font.sans-serif']=['SimHei']

#负号正常显示
plt.rcParams['axes.unicode_minus']=False

#读取数据，将原有的拼多多数据文件的中文名字改成 pdd_coupon.csv，避免因字符出现警告和错误
pdd_data=pd.read_csv(r'C:/Users/Marty/Desktop/week7/pdd_coupon.csv')
```

#Step1.数据预处理

pdd_data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25317 entries, 0 to 25316
Data columns (total 9 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   age                         25317 non-null  int64
 1   job                         25317 non-null  object
 2   marital                     25317 non-null  object
 3   default                     25317 non-null  object
 4   returned                    25317 non-null  object
 5   loan                        25317 non-null  object
 6   coupon_used_in_last6_month  25317 non-null  int64
 7   coupon_used_in_last_month   25317 non-null  int64
 8   coupon_ind                  25317 non-null  int64
dtypes: int64(4), object(5)
memory usage: 1.7+ MB
```

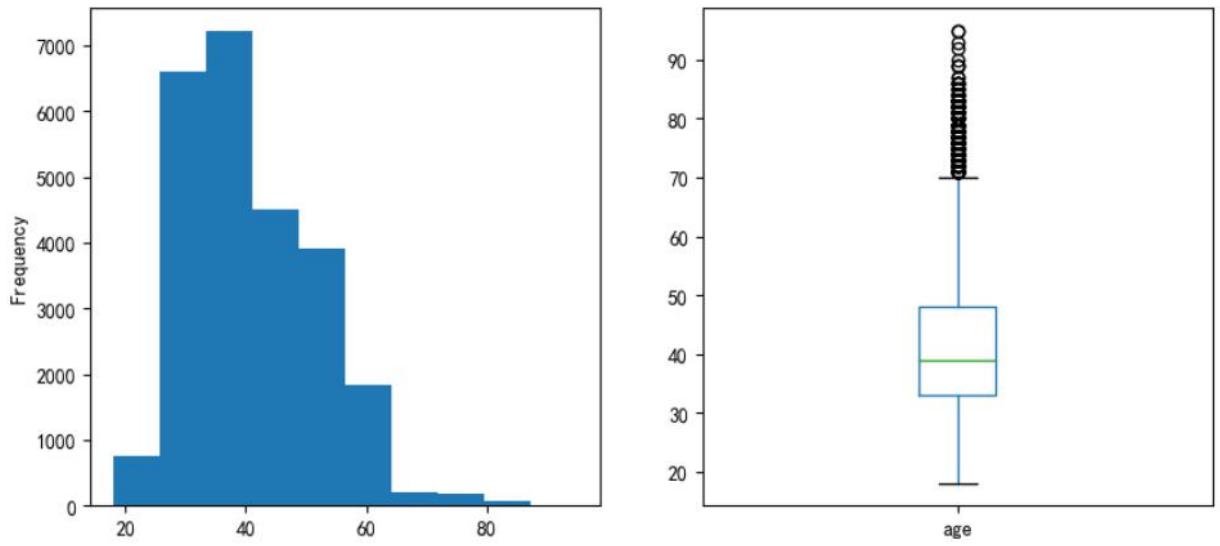
pdd_data.head()

	age	job	marital	default	returned	loan	coupon_used_in_last6_month	coupon_used_in_last_month	coupon_ind
0	43	management	married	no	yes	no	2	0	0
1	42	technician	divorced	no	yes	no	1	1	0
2	47	admin.	married	no	yes	yes	2	0	0
3	28	management	single	no	yes	yes	2	0	0
4	42	technician	divorced	no	yes	no	5	0	0

#数值型数据查看

#年龄数据可视化

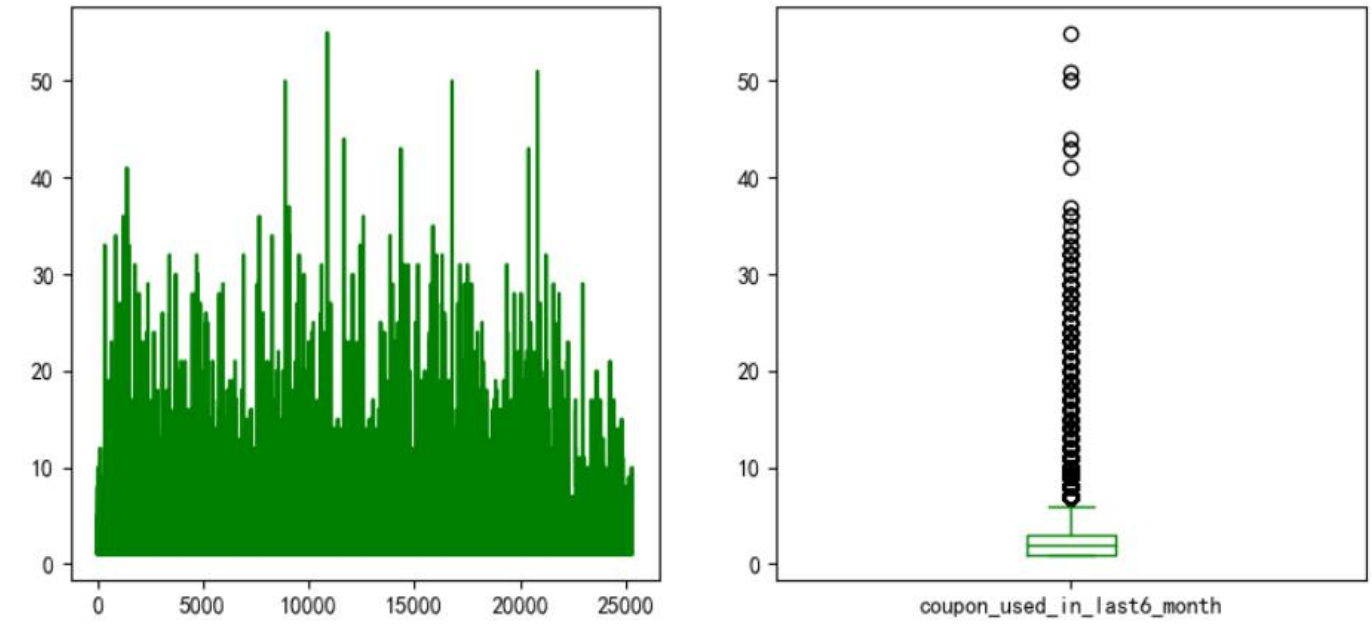
```
plt.figure(figsize=(10,10), dpi=100)
plt.subplot(221)
pdd_data['age'].plot(kind='hist');
plt.subplot(222)
pdd_data['age'].plot(kind='box');
```



#过去六个月使用的优惠券次数 数据可视化

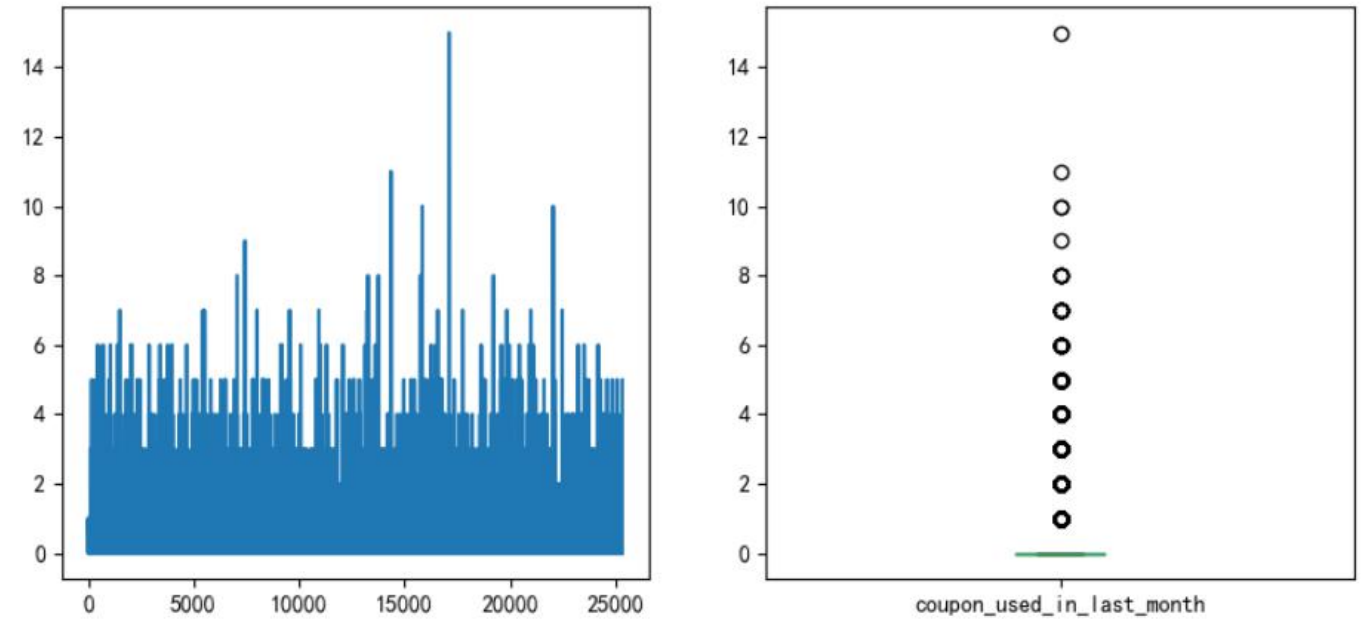
```
plt.figure(figsize=(10,10), dpi=100)
plt.subplot(221)
```

```
pdd_data['coupon_used_in_last6_month'].plot(kind='line',color='green');
plt.subplot(222)
pdd_data['coupon_used_in_last6_month'].plot(kind='box',color='green');
```



#过去一个月使用的优惠券次数 数据可视化

```
plt.figure(figsize=(10,10), dpi=100)
plt.subplot(221)
pdd_data['coupon_used_in_last_month'].plot(kind='line');
plt.subplot(222)
pdd_data['coupon_used_in_last_month'].plot(kind='box');
```



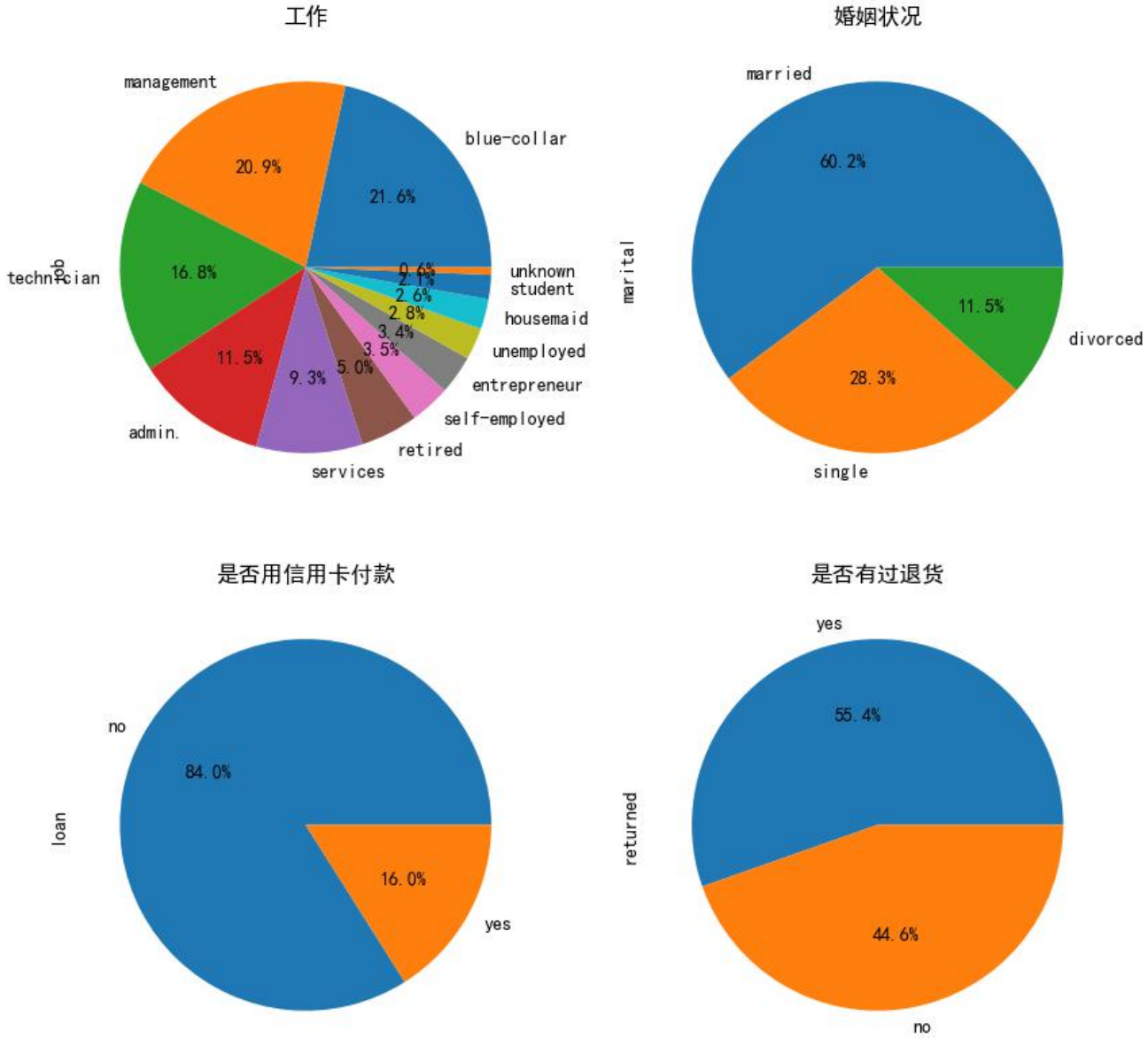
#分类型数据查看

```
pdd_data['marital'].value_counts()
```

```
married     15245
single       7157
divorced     2915
Name: marital, dtype: int64
```

#数据可视化

```
plt.figure(figsize=(10,10), dpi=100)
plt.subplot(221)
pdd_data['job'].value_counts().plot(kind='pie',autopct = '%3.1f%%',title='工作');
plt.subplot(222)
pdd_data['marital'].value_counts().plot(kind='pie',autopct = '%3.1f%%',title='婚姻状况');
plt.subplot(223)
pdd_data['loan'].value_counts().plot(kind='pie',autopct = '%3.1f%%',title='是否用信用卡付款');
plt.subplot(224)
pdd_data['returned'].value_counts().plot(kind='pie',autopct = '%3.1f%%',title='是否有过退货');
```



本次收录职业数据大部分为蓝领人员，其次到管理者，学生占比最低（未知不算）；
大部分人都已结婚，占 60.2%；
拼多多平台用户不习惯用信用卡付款，只有 16%的用户用信用卡付款；
大部分都有过退货经历，占 55.4%；

#数值型变量与关键列的关系

```
pdd_data.groupby('coupon_ind')['age'].std()
```

```
coupon_ind
0    10.17824
1    13.56260
Name: age, dtype: float64
```

#分类型变量与关键列的关系

```
pdd_data.groupby('coupon_ind')['marital'].value_counts(1)
```

```
coupon_ind  marital
0           married     0.611916
            single      0.273260
            divorced    0.114824
1           married     0.528538
            single      0.353934
            divorced    0.117528
Name: marital, dtype: float64
```

```
pdd_data.groupby('coupon_ind')['job'].value_counts(1)
```

```
coupon_ind  job
0           blue-collar      0.226740
            management       0.203972
            technician       0.168188
            admin.           0.114868
            services         0.095321
            retired          0.043612
            entrepreneur     0.035293
            self-employed    0.034845
            housemaid        0.027062
            unemployed       0.026257
            student          0.017445
            unknown          0.006396
1           management       0.248565
            technician       0.162445
            blue-collar      0.130699
            admin.           0.115164
            retired          0.100642
            services         0.071260
            student          0.048294
            unemployed       0.038501
            self-employed    0.035461
            entrepreneur     0.022627
            housemaid        0.019588
            unknown          0.006754
Name: job, dtype: float64
```

可以看到，整个拼多多数据表中整体数据不存在缺失值。
其中，类别型变量占比正常。
数字型变量不存在较明显异常值。

#转化为哑变量
#发现有 10 列数据，其中 5 种为类别型变量。其余变量用 get_dummies 转换成数据型变量，然后删除重复变量‘default_no’,‘returned_no’, ‘loan_no’
#将要分析的因变量 ‘coupon_ind’自定义名字为‘flag’

```
dumpdd=pd.get_dummies(pdd_data)
```

```
dumpdd.head()
```

	age	coupon_used_in_last6_month	coupon_used_in_last_month	coupon_ind	job_admin.	job_blue-collar	job_entrepreneur	job_housemaid	job_management	job_
0	43	2	0	0	0	0	0	0	1	
1	42	1	1	0	0	0	0	0	0	
2	47	2	0	0	1	0	0	0	0	
3	28	2	0	0	0	0	0	0	1	
4	42	5	0	0	0	0	0	0	0	

5 rows × 25 columns

```
dumpdd.drop(['default_no','returned_no','loan_no'],axis=1,inplace=True)
```

```
newpdd=dumpdd.rename(columns={'coupon_ind':'flag'})
```

#Step2.关键变量的选择

```
newpdd.flag.value_counts()
```

```
0    22356
1     2961
Name: flag, dtype: int64
```

```
newpdd.flag.value_counts(1)
```

```
0    0.883043
1    0.116957
Name: flag, dtype: float64
```

#value_counts()查看用户分别“用”和“不用”优惠券的数量及比例，不用优惠券（flag=0）的占多数。

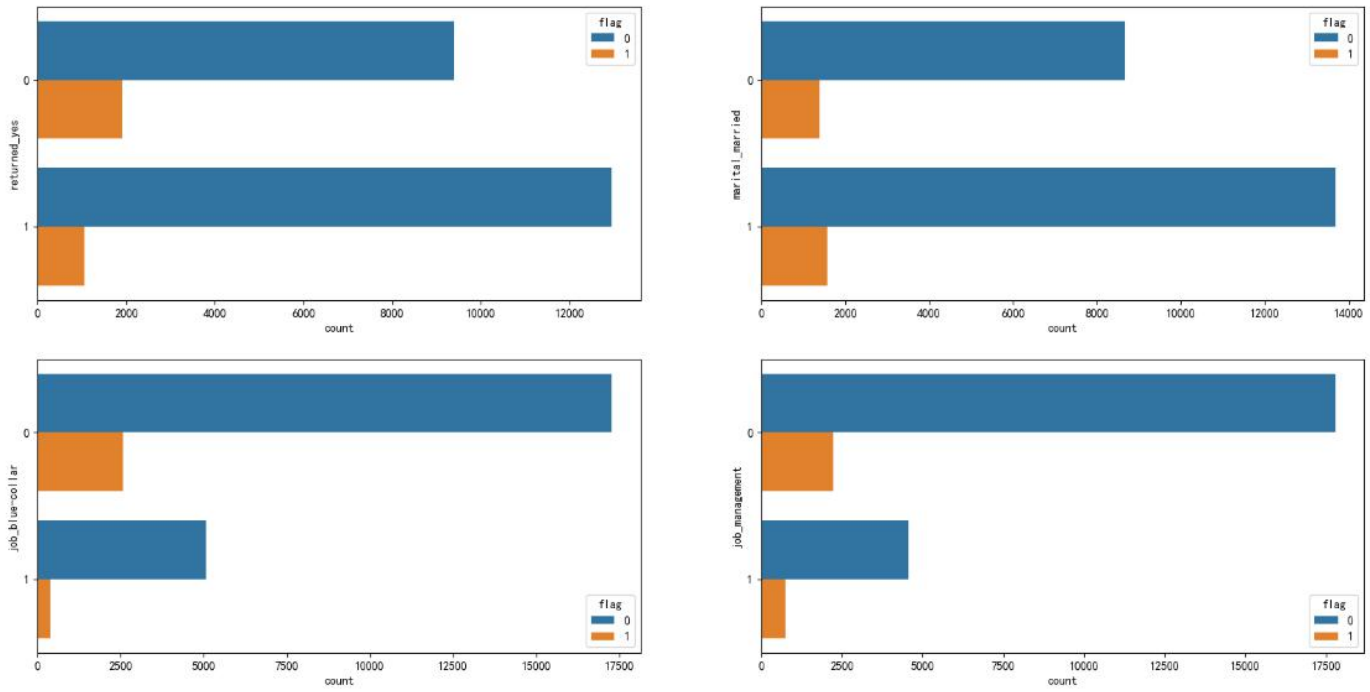
```
summary=newpdd.groupby('flag')
summary.mean().T
```

flag	0	1
age	40.819601	41.809524
coupon_used_in_last6_month	2.857846	2.124282
coupon_used_in_last_month	0.260378	0.537994
job_admin.	0.114868	0.115164
job_blue-collar	0.226740	0.130699
job_entrepreneur	0.035293	0.022627
job_housemaid	0.027062	0.019588
job_management	0.203972	0.248565
job_retired	0.043612	0.100642
job_self-employed	0.034845	0.035461
job_services	0.095321	0.071260
job_student	0.017445	0.048294
job_technician	0.168188	0.162445
job_unemployed	0.026257	0.038501
job_unknown	0.006396	0.006754
marital_divorced	0.114824	0.117528
marital_married	0.611916	0.528538
marital_single	0.273260	0.353934
default_yes	0.018876	0.008781
returned_yes	0.579755	0.357649
loan_yes	0.169037	0.094563

#根据各变量使用优惠券的比例，取同类特征最明显（比例大）的进行可视化分析：

```
#数据可视化
plt.figure(figsize=(20,10), dpi=100)
#没有退过货的用户（左上）
plt.subplot(221)
sns.countplot(y='returned_yes', hue='flag' , data=newpdd)
plt.subplot(222)
#已婚用户（右上）
sns.countplot(y='marital_married', hue='flag' , data=newpdd)
plt.subplot(223)
#职业 蓝领工作者（左下）
sns.countplot(y='job_blue-collar', hue='flag' , data=newpdd)
plt.subplot(224)
#职业 管理者（右下）
sns.countplot(y='job_management', hue='flag' , data=newpdd)
```

<matplotlib.axes._subplots.AxesSubplot at 0x1dd1eacea08>



业务分析：

发现没有退过货的用户，其在该活动中使用优惠券的比例明显较高；

发现已婚用户，其在该活动中使用优惠券的比例比较低；

发现蓝领，其在该活动中不使用优惠券的比例约为使用优惠券的 1.7 倍；

发现管理人员，其在该活动中使用优惠券的比例高于未使用的比例；

业务理解：没有退过货说明顾客对拼多多平台的服务满意，所以更容易消费，也更有可能使用优惠券；

已婚用户由于已经结婚，经济水平普遍比未婚的小年轻要高，所以不会容易被低价打动。

蓝领人员为产业人员，薪资水平低，购买欲望低，较少参加优惠活动，都是按需购买为主，所以极少遇到优惠情况；

发现管理人员，职业导致其有着精打细算的特质，更趋向于使用优惠券；

#相关性分析
newpdd.corr()[['flag']].sort_values('flag',ascending=False)

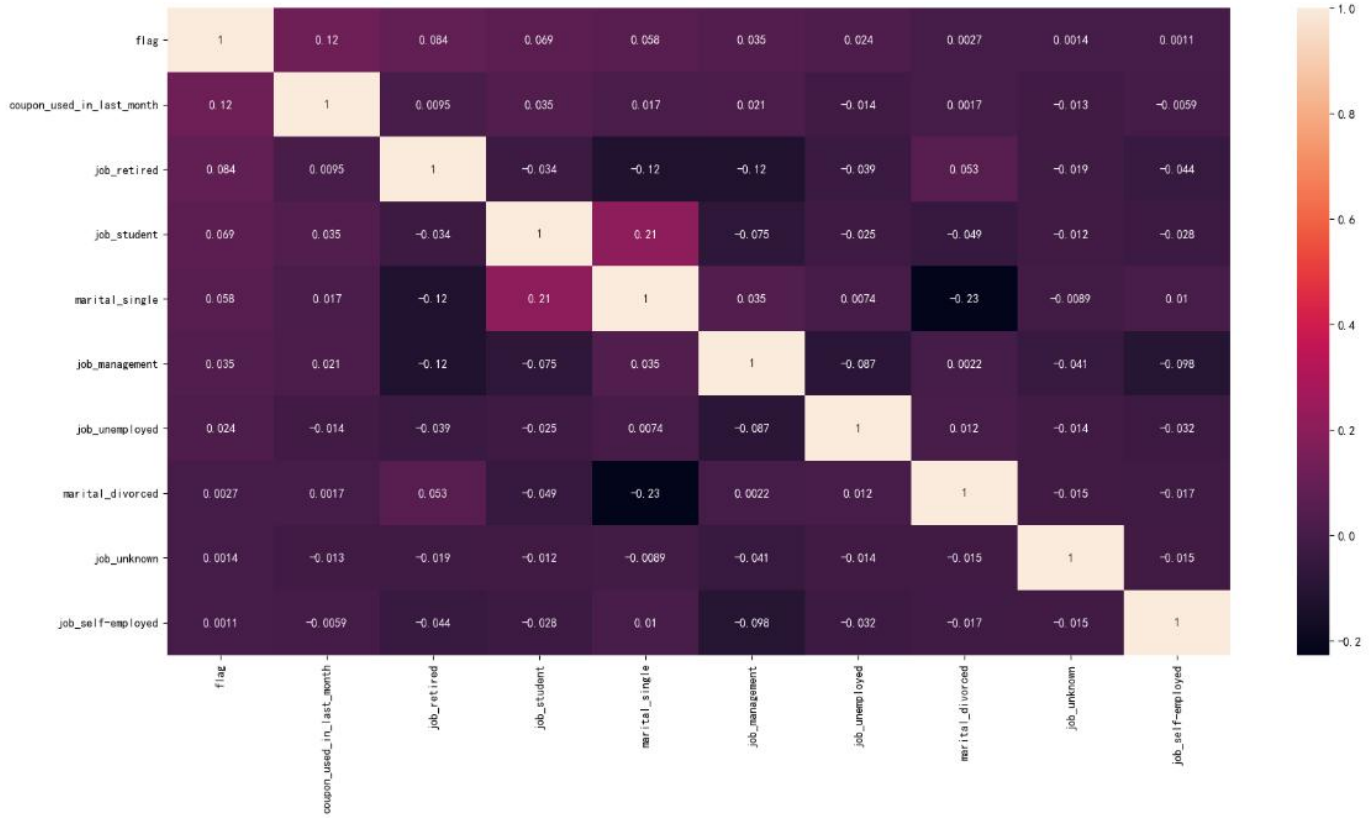
相关系数大于 0.1 只有 coupon_used_in_last_month，而相对比较高的相关系数（除了 coupon_used_in_last_month）有：‘job_retired’，‘job_student’，‘marital_single’，‘job_management’，‘age’
age 是数值型数据，其他都为分类型数据。

	flag
flag	1.000000
coupon_used_in_last_month	0.116550
job_retired	0.083868
job_student	0.069058
marital_single	0.057574
job_management	0.035234
age	0.029916
job_unemployed	0.023980
marital_divorced	0.002723
job_unknown	0.001438
job_self-employed	0.001078
job_admin.	0.000298
job_technician	-0.004942
job_housemaid	-0.015041
job_entrepreneur	-0.022519
default_yes	-0.024608
job_services	-0.026688
marital_married	-0.054746
loan_yes	-0.065231
job_blue-collar	-0.075065
coupon_used_in_last6_month	-0.075173
returned_yes	-0.143589

#使用热力图呈现变量间相关性（所有与 flag 正相关的变量）- 研究相关性的影响

```
plt.figure(figsize=(20,10), dpi=100)
q1=['flag','coupon_used_in_last_month','job_retired','job_student','marital_single','job_management','job_unemployed','marital_divorced','job_unknown','job_self-employed']
sns.heatmap(newpdd[q1].corr(),annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1dd1d3dcf08>
```



可以发现，除了与 flag 的相关系数外，
单身的大多数是学生，呈高相关性，marital_single 和 job_student 的相关系数为 0.21
离异的有部分退休了，marital_divorced 和 job_retired 呈一定的正相关，相关系数为 0.053

#Step3.建立模型
根据 Step2，选择相关系数较高的变量建立模型，我第一次选择了
‘coupon_used_in_last_month’，‘job_retired’，‘marital_single’这三个变量作为自变量，flag 为因变量

```
y=newpdd['flag']
```

#选取三个变量建模：coupon_used_in_last_month 相关性最高，job 和 marital 选取了最高相关性的类型。

```
x=newpdd[['coupon_used_in_last_month','job_retired','marital_single']]
```

```
from sklearn.model_selection import train_test_split
```

#选取的训练集和测试集比例为 3：7
调用 sklearn 模块，随机抽取测试集和训练集，因为数据量比较小，所以选择测试集占比 0.3，
调用 sklearn 的逻辑回归模块，然后进行模型拟合

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=100)
```

```
from sklearn import linear_model
lr=linear_model.LogisticRegression()
```

```
lr.fit(x_train,y_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

```
lr.intercept_
```

```
array([-2.3347034])
```

```
lr.coef_
```

```
array([[0.33935126, 1.06444793, 0.4009973 ]])
```

```
y_pred_train=lr.predict(x_train)
y_pred_test=lr.predict(x_test)
print(y_pred_train)
```

```
[0 0 0 ... 0 0 0]
```

```
print(y_pred_test)
```

```
[0 0 0 ... 0 0 0]
```

```
import sklearn.metrics as metrics
```

#基于模型的结果，对训练集与测试集中 x 的真实值预测对应的 y

#训练集

metrics.confusion_matrix(y_train,y_pred_train)

```
array([[15600,    23],
       [ 2093,     5]], dtype=int64)
```

metrics.accuracy_score(y_train,y_pred_train)

```
0.8805936459567745
```

#测试集

metrics.confusion_matrix(y_test,y_pred_test)

```
array([[6725,    8],
       [ 863,    0]], dtype=int64)
```

metrics.accuracy_score(y_test,y_pred_test)

```
0.8853343865192207
```

搭建混淆矩阵，根据 pred_correct=(TP+TN)/(TP+TN+FP+FN)公式可得模型的准确率：
训练集：0.8805936459567745
测试集：0.8853343865192207

测试集与训练集不存在明显差异，故判断测试集有效。

```
from sklearn.metrics import roc_curve,auc
fpr,tpr,threshold=roc_curve(y_train,y_pred_train)
roc_auc=auc(fpr,tpr)
```

print(roc_auc)

```
0.5004555168380904
```

可以发现，使用 AUC 评估模型得 0.50046 ，此模型不是很差，但需要优化。

#Step4 模型优化

#（1）**训练集：测试集比例**调成 1：1，改变 Step3（**加粗标注**）的一行代码，其他不变

x_train,x_test,y_train,y_test=train_test_split(x,y,**test_size=0.5**,random_state=100)

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

#训练集 metrics.accuracy_score

```
0.8831568968241429
```

#测试集 metrics.accuracy_score

```
0.8808752666087368
```

#roc_auc

```
0.5002205880199286
```

训练集上升（0.88059 ➞ 0.88315），测试集下降（0.88533 ➞ 0.88088），
auc 评分下降（0.50046 ➞ 0.50022）
可以发现，测试集的样本变多了，反而准确率下降了，说明数据的样本数太少。

#(2)增加自变量优化模型
#把自变量增加至五个（选取相关性排在前列的）
x=newpdd**[['coupon_used_in_last_month','job_retired','job_student','marital_single','job_management']]**

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

#训练集 metrics.accuracy_score

```
0.8808757970769144
```

#测试集 metrics.accuracy_score

```
0.8858609794628752
```

#roc_auc

```
0.5008218554358179
```

训练集（0.88059 ➞ 0.88088），测试集（0.88533 ➞ 0.88586），
auc 评分（0.50046 ➞ 0.50082），三者都略微有所提升。

#模型解读

```
import numpy as np
newx=newpdd[['coupon_used_in_last_month','job_retired','job_student','marital_single','job_management']]
np.exp(lr.coef_)
```

```
array([[1.39292102, 3.22884924, 2.5436426 , 1.36978468, 1.50032324]])
```

由模型斜率可知，
近一个月使用过优惠券的用户，在本次活动使用优惠券的可能性是近一个月没有使用优惠券的 1.39 倍，
退休用户本次活动使用优惠券的可能性远高于职业为管理人员的（3.2288 VS 1.5003）；
学生或者退休用户和在一个月内有过使用优惠券记录的用户更容易使用优惠券；

业务建议：
建议给学生或者退休用户更多低折扣的优惠券增加消费量，
给非学生的在职人员发高折扣的优惠券促进其使用优惠券消费，
同样的道理，为在一个月内没使用优惠券的用户提供高折扣的优惠券。