利用逻辑回归实现
客户流失率预测分析

目　录

Contents

Part 1

项目分析

Part 2

变量分析

Part 3

特征工程

Part 4

模型训练与评估

Part 5

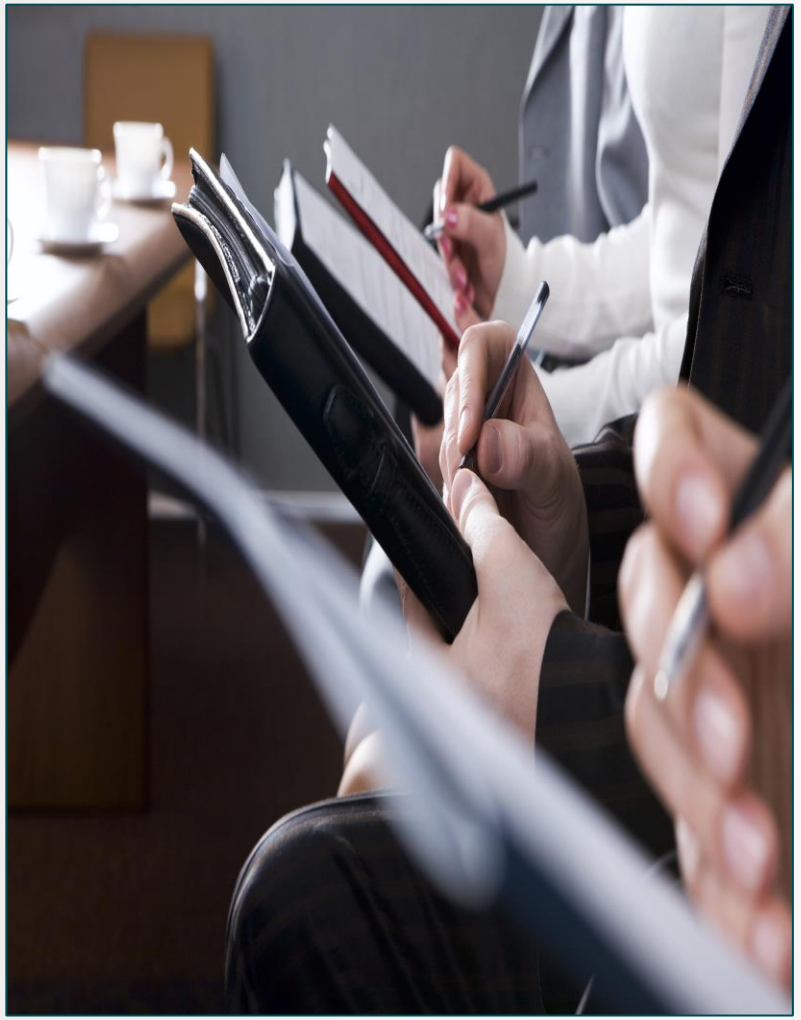
总结

Part 1

项目分析

- 项目概述
- 理解数据集
- 学习目标
- 客户流失率

项目概述



“流失率”是一个商业术语，用来描述客户离开或停止支付产品或服务的费率。

客户流失率问题是电信运营商面临的一项重要课题；根据测算，招揽新的客户比保留住既有的客户花费大得多（通常5-20倍的差距）。

因此如何保留住现有的客户对运营商来说是一项非常有意义的事情。

本项目通过一份公开数据，使用所学的机器算法知识，预测客户流失率，将所学知识运用到实际问题中。

学习目标

01

掌握机器学习的实施

了解机器学习的数据预处理流程、方法与具体实现
了解机器学习的数据探索的思路、方法与具体实现

02

能够使用python进行数据分析

熟练使用Pandas来进行数据准备工作
熟练使用matplotlib等工具进行数据可视化

03

会用算法预测数据

能够利用逻辑回归算法进行数据预测

理解数据集

- 导入pandas库
- import pandas as pd
- 加载数据
- data=pd.read_csv('churn.csv')
- 显示数据
- data.head()

我们使用的数据集是一个长期的电信客户数据集。数据中每行代表客户，每列包含客户属性，例如电话号码，在一天中的不同时间的呼叫分钟，服务产生的费用，终身帐户的持续时间以及客户是否流失。

	State	Account Length	Area Code	Phone	Int'l Plan	VMail Plan	VMail Message	Day Mins	Day Calls	Day Charge	...	Eve Calls	Eve Charge	Night Mins	Night Calls	Night Charge	Intl Mins	Intl Calls	Intl Charge	CustServ Calls	Churn?
	KS	128	415	382-4657	no	yes	25	265.1	110	45.07	...	99	16.78	244.7	91	11.01	10.0	3	2.70	1	False.
	OH	107	415	371-7191	no	yes	26	161.6	123	27.47	...	103	16.62	254.4	103	11.45	13.7	3	3.70	1	False.
	NJ	137	415	358-1921	no	no	0	243.4	114	41.38	...	110	10.30	162.6	104	7.32	12.2	5	3.29	0	False.
	OH	84	408	375-9999	yes	no	0	299.4	71	50.90	...	88	5.26	196.9	89	8.86	6.6	7	1.78	2	False.
	OK	75	415	330-6626	yes	no	0	166.7	113	28.34	...	122	12.61	186.9	121	8.41	10.1	3	2.73	3	False.

rows × 21 columns

'State'：州名
'Account Length'：账户长度
'Area Code'：区号
'Phone'：电话号码
"Int'l Plan"：国际计划
'VMail Plan'：语音邮箱
'VMail Message'：语言消息
'Day Mins'：白天通话分钟数
'Day Calls'：白天电话个数
'Day Charge'：白天通话收费
'Eve Mins'：晚间通话分钟数
'Eve Calls'：晚间电话个数
'Eve Charge'：晚间通话收费
'Night Mins'：夜间通话分钟数
'Night Calls'：夜间电话个数
'Night Charge'：夜间通话收费
'Intl Mins'：国际分钟数
'Intl Calls'：国际电话个数
'Intl Charge'：国际通话收费
'CustServ Calls'：客服电话数
'Churn?'：流失与否

流失率比例

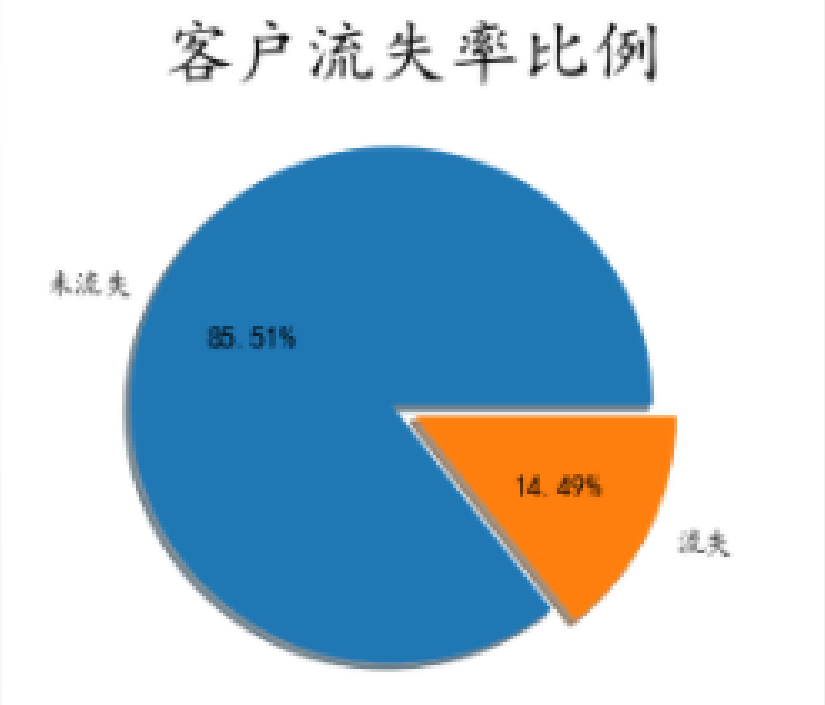
查看数据情况
data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3333 entries, 0 to 3332
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   State           3333 non-null   object
 1   Account Length  3333 non-null   int64
 2   Area Code       3333 non-null   int64
 3   Phone           3333 non-null   object
 4   Int'l Plan      3333 non-null   object
 5   VMail Plan      3333 non-null   object
 6   VMail Message   3333 non-null   int64
 7   Day Mins        3333 non-null   float64
 8   Day Calls       3333 non-null   int64
 9   Day Charge      3333 non-null   float64
 10  Eve Mins        3333 non-null   float64
 11  Eve Calls       3333 non-null   int64
 12  Eve Charge      3333 non-null   float64
 13  Night Mins      3333 non-null   float64
 14  Night Calls     3333 non-null   int64
 15  Night Charge    3333 non-null   float64
 16  Intl Mins       3333 non-null   float64
 17  Intl Calls      3333 non-null   int64
 18  Intl Charge     3333 non-null   float64
 19  CustServ Calls  3333 non-null   int64
 20  Churn?          3333 non-null   object
dtypes: float64(8), int64(8), object(5)
memory usage: 546.9+ KB
```

数据情况很好，无缺失值

查看总体客户流失情况

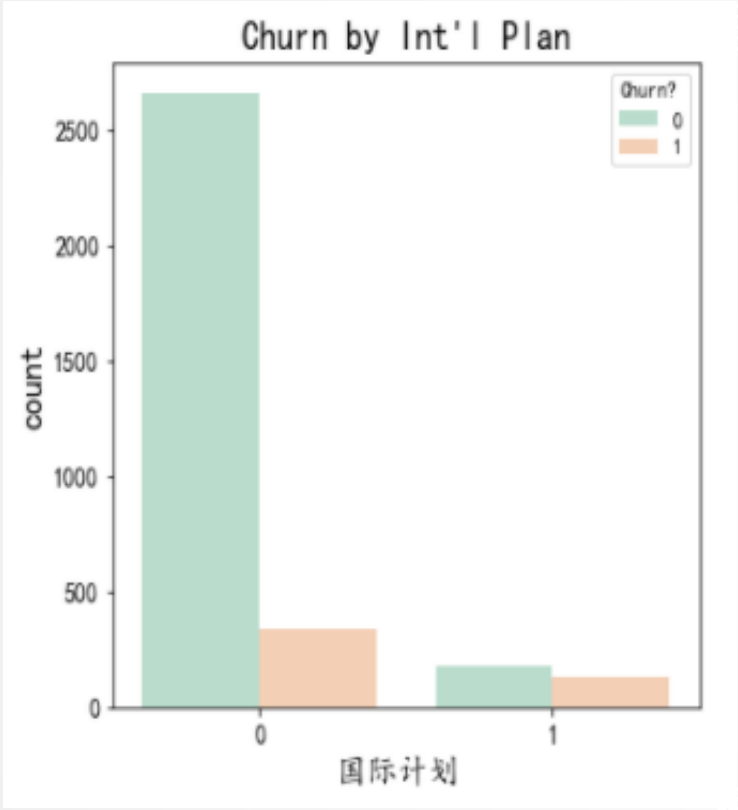
```
churnvalue = data["Churn?"].value_counts()
labels = data["Churn?"].value_counts().index
plt.pie(churnvalue,labels=[“未流失”，“流失”],
explode=(0.1,0),autopct='%.2f%%', shadow=True,)
plt.title("客户流失率比例",size=24)
```



Part 2

变量分析

变量分析



可以看出订购国际计划的客户大部分流失了

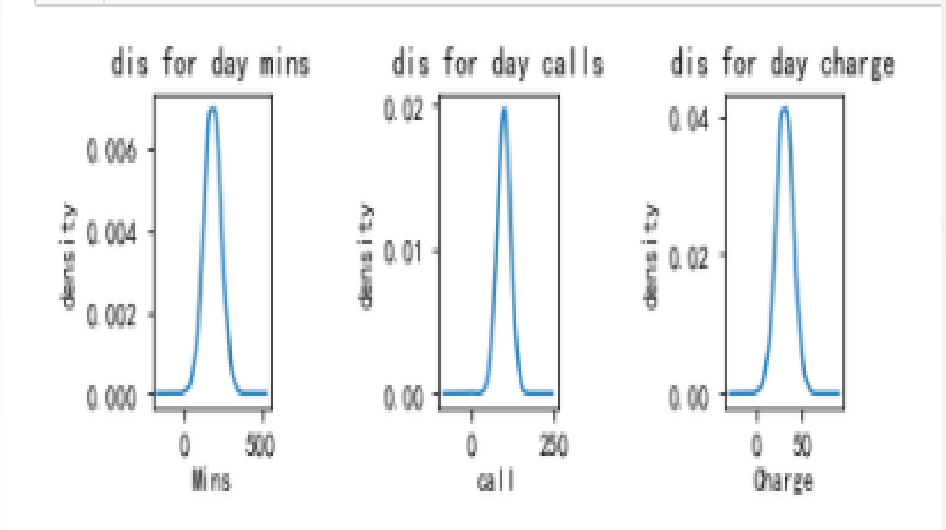
删除同值化数据

```
for c in data:
    if data[c].nunique()==1:
        print(c,data[c].nunique())
        data=data.drop(c,axis=1)
```

```
fig, axes = plt.subplots(1, 1, figsize=(12,12))
plt.subplot(2,2,1)
# palette参数表示设置颜色
gender=sns.countplot(x="Int'l
Plan",hue="Churn?",data=data,palette="Pastel2")
plt.xlabel("国际计划 ",fontsize=16)
plt.ylabel('count',fontsize=16)
plt.tick_params(labelsize=12)     # 设置坐标轴字体大小
plt.title("Churn by Int'l Plan",fontsize=18)
```

变量分析

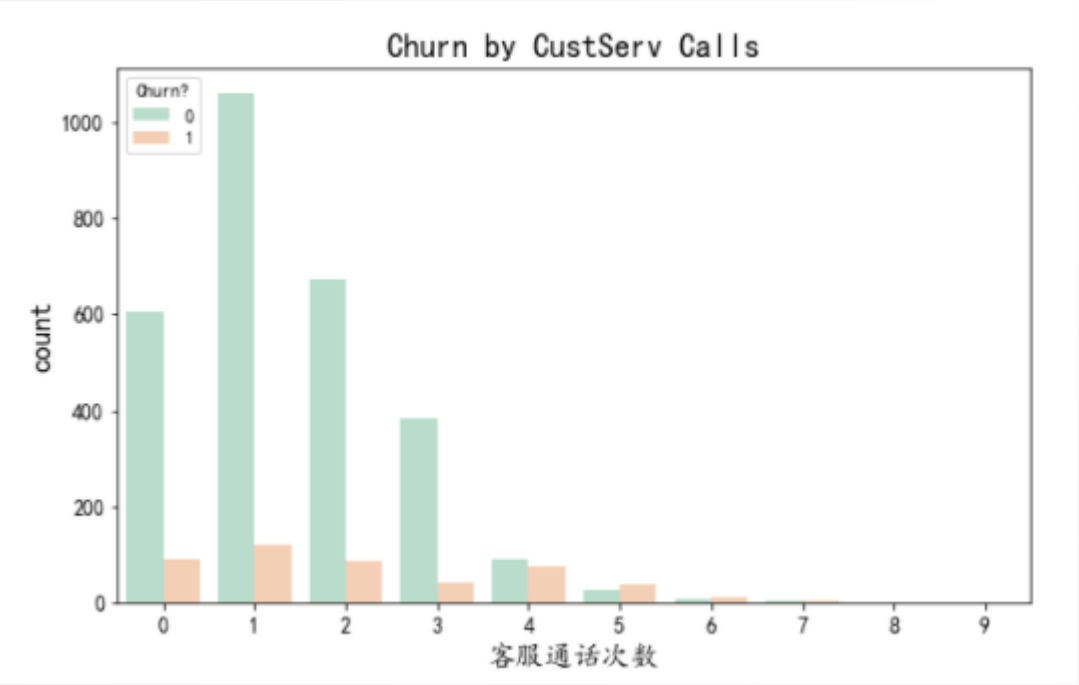
```
plt.subplot2grid(( 2, 5),( 0, 0))
data[ 'Day Mins'].plot(kind= 'kde')
plt.xlabel( u"Mins")
plt.ylabel( u"density")
plt.title( u"dis for day mins")

plt.subplot2grid(( 2, 5),( 0, 2))
data[ 'Day Calls'].plot(kind= 'kde')
plt.xlabel( u"call")
plt.ylabel( u"density")
plt.title( u"dis for day calls")

plt.subplot2grid(( 2, 5),( 0, 4))
data[ 'Day Charge'].plot(kind= 'kde')
plt.xlabel( u"Charge")
plt.ylabel( u"density")
plt.title( u"dis for day charge")
plt.show()
```



分别展现出了白天的通话次数，通话时间，和收费情况的密度，差不多都呈现出正态分布的样式，三个模型属于一样的图像，三个变量也是呈现出正比的样式

变量分析

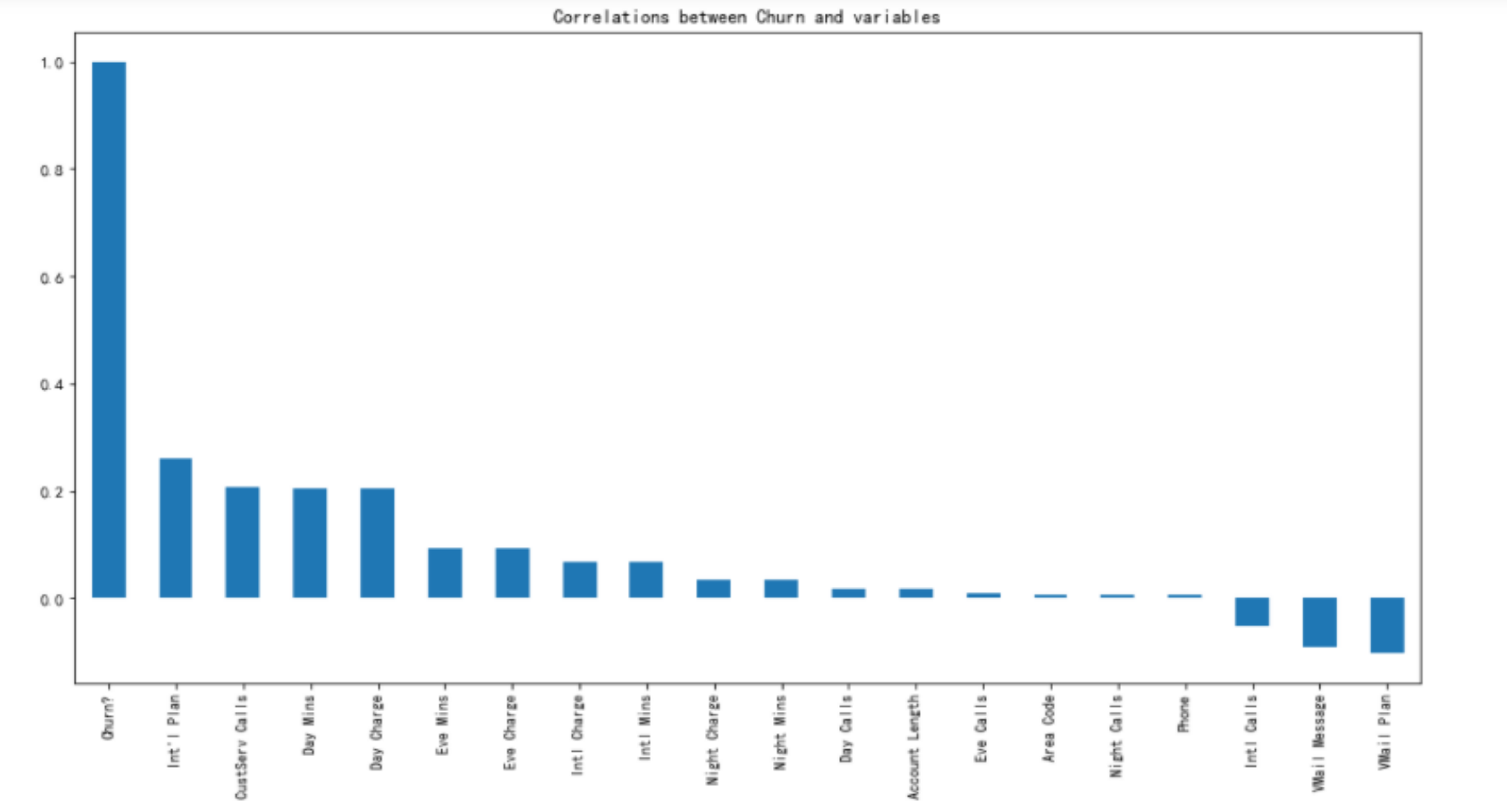
五、完成数据表，总结



差不多在超过3次客服通话次数的客户流失了许多，5个以上基本没有了，说明客户的问题增多，难以解决了

变量分析

五、完成数据表，总结



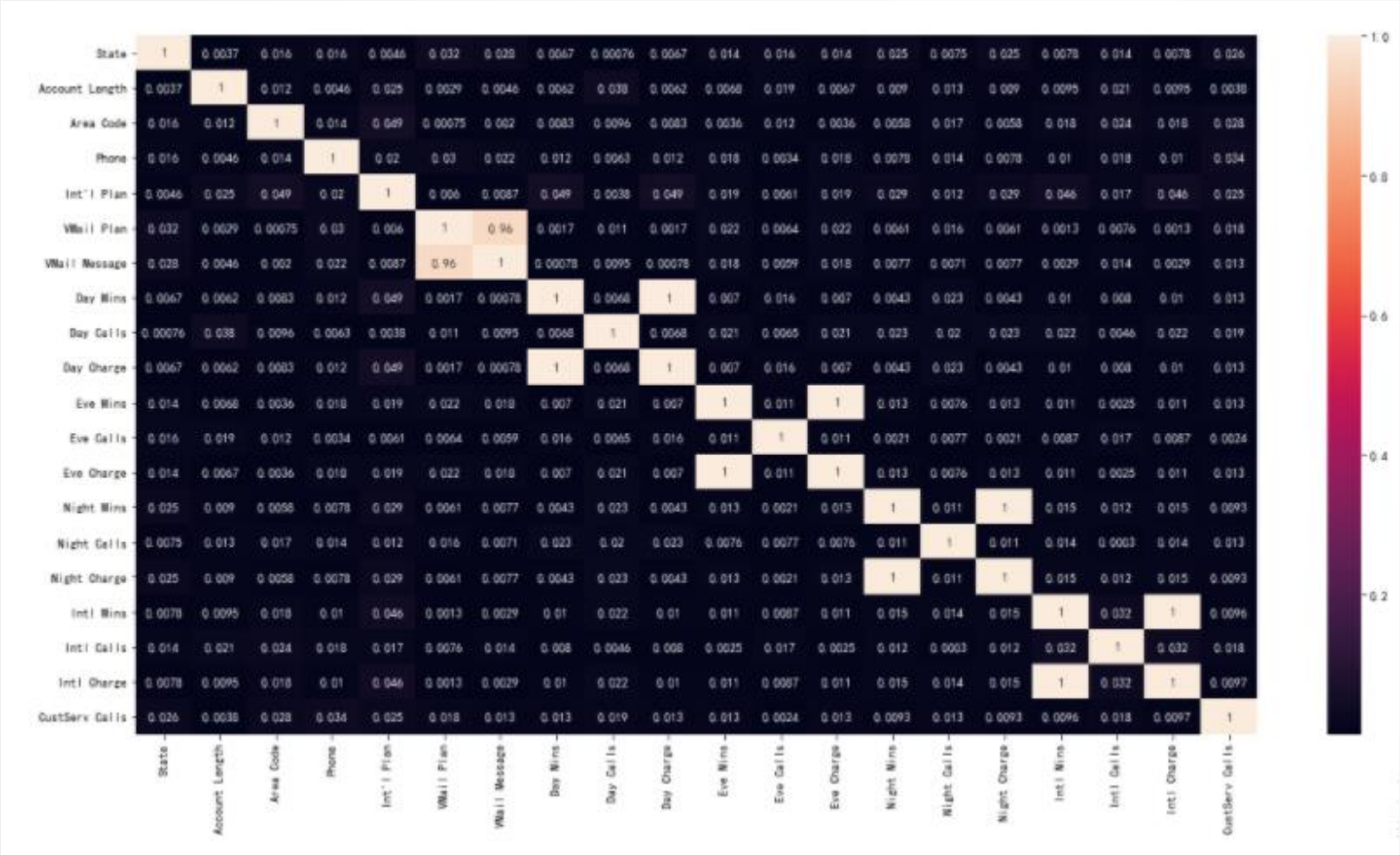
Part 3

特征变量

逻辑回归

```
from sklearn.model_selection import train_test_split
用于拆分数据集
from sklearn.linear_model import LogisticRegression
导入逻辑回归
对字符型变量进行LabelEncoder编码
labelencoder=LabelEncoder()
for i in data:
    if data[i].dtypes == 'object':
        data[i] = labelencoder.fit_transform(data[i])
 拆分自变量和因变量
 X=data.drop('Churn?',axis=1)
 y=data[['Churn?’ ]]
 x_train, x_test, y_train, y_test =
 train_test_split(x,y,test_size=0.4,random_state=100)  #
 把数据集拆分为训练测、测试集
 lr=LogisticRegression()  实例化逻辑回归
 lr.fit(x_train,y_train)  训练模型
```

相关性可视化



自变量相关性可视化
X_corr = X.corr().abs()
plt.subplots(figsize=(20,10))
sns.heatmap(X_corr,annot=True)

使用热图进行可视化分析，将相关性>0.7的变量删除
x=X.drop(['VMail Message','Day Charge','Eve Charge','Night Charge','Intl Charge'],axis=1)

Part 4

模型训练与评估

模型训练与评估

导入数据集拆分模型，可以将数据集拆分为训练集个测试集
from sklearn.model_selection import train_test_split
导入模型分析报告,用于评估模型效果
from sklearn.metrics import classification_report

import warnings
warnings.filterwarnings("ignore")忽略警告

x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.4,random_state=100) 把数据集拆分为训练测、测试集
lr=LogisticRegression() 实例化逻辑回归
lr.fit(x_train,y_train) 训练模型

```
from sklearn.metrics import accuracy_score
y_train_pred=lr.predict(x_train)  #训练数据预测值
y_test_pred=lr.predict(x_test)  #测试数据预测值
print ('train accuracy_score: %0.3f' % accuracy_score(y_train, y_train_pred))  #打印训练集精确率
print ('test accuracy_score: %0.3f' % accuracy_score(y_test, y_test_pred))  #打印测试集精确率
```

```
train accuracy_score: 0.850
test accuracy_score: 0.862
```

评分报告

评分报告（precision：准确率 recall：召回率）
print(classification_report(y_train,y_train_pred))
print(classification_report(y_test,y_test_pred))

	precision	recall	f1-score	support
0	0.86	0.99	0.92	1697
1	0.55	0.06	0.11	302
accuracy			0.85	1999
macro avg	0.70	0.53	0.51	1999
weighted avg	0.81	0.85	0.80	1999

	precision	recall	f1-score	support
0	0.87	0.99	0.93	1153
1	0.43	0.05	0.09	181
accuracy			0.86	1334
macro avg	0.65	0.52	0.51	1334
weighted avg	0.81	0.86	0.81	1334

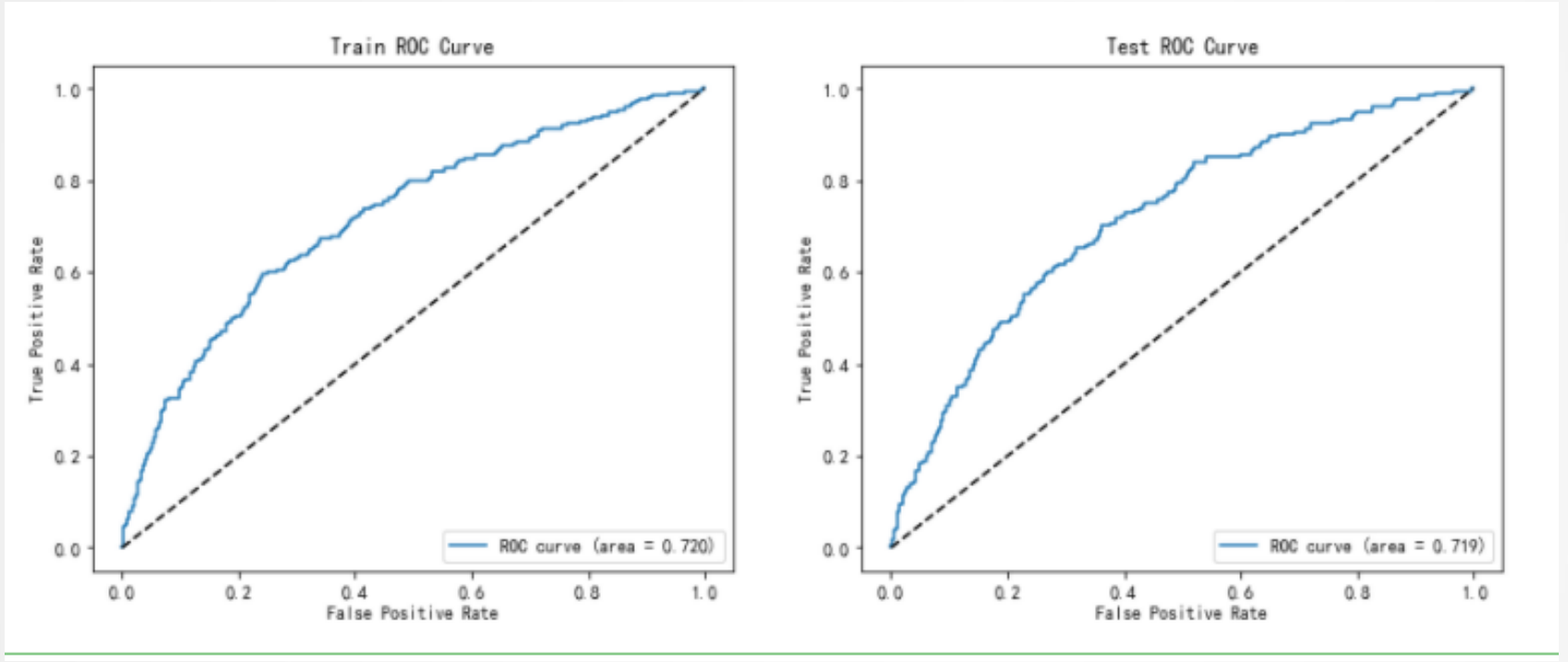
模型预测为 False 的准确率为0.86，召回率为0.99
模型预测为True 的准确率为 0.55，召回率为 0.06，召回率非常低，与本身数据存在很大的关系，需要进行调参

模型训练与评估

```
#Roc_Auc曲线
from sklearn.metrics import roc_curve,auc
y_train_prob=lr.predict_proba(x_train)[:,1]  #训练数据预测为客户流失的概率
y_test_prob=lr.predict_proba(x_test)[:,1]  #测试数据预测为客户流失的概率
fpr1, tpr1, _ = roc_curve(y_test, y_test_prob)
fpr2, tpr2, _ = roc_curve(y_train, y_train_prob)
roc_auc1 = auc(fpr1, tpr1)
roc_auc2 = auc(fpr2, tpr2)

fig=plt.figure(figsize=(14,5))
ax1 = fig.add_subplot(1,2,1)
plt.plot(fpr2, tpr2, label='ROC curve (area = %0.3f)' % roc_auc2)
plt.plot([0, 1], [0, 1], 'k--')
plt.legend(loc="lower right")
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Train ROC Curve')
ax2 = fig.add_subplot(1,2,2)
plt.plot(fpr1, tpr1, label='ROC curve (area = %0.3f)' % roc_auc1)
plt.plot([0, 1], [0, 1], 'k--')
plt.legend(loc="lower right")
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Test ROC Curve')
```

ROC曲线，又称接受者操作特征曲线，主要用于评价模型的预测能力

图形展示



训练和测试模型的曲线差不多，说明模型的精确度匹配，与模型评估的精确率相似

Part 5

总结

总结

项目总结

模型的召回率太低，对于如何提高召回率，自己只有一个大概，但具体不知道有何办法，应该多实用套餐更加的合理，在这个模型中对客户的服务存在问题

不足

整个代码做完发现有很多地方或者知识点都没有用到，做的时候卡了很多地方，进行删删减减，对于用哪个知识点比较模糊

收获

在这次项目中，我收获了很多知识，可能我对于整个机器学习的流程理解还不是很透彻，但对我来说，我觉得能做出来真的很不容易了

致 谢

感谢博为峰提供的学习与实践的机会；

感谢老师们的耐心指导；

感谢班主任对我的督促；

感谢同学们对我的帮助；

感谢答辩评审！

感谢您的批评指正