电信客户流失分析

1. 背景

现如今，在电信行业蓬勃发展的同时，电信市场也趋于饱和，获取一个新客户的难度要远远高于维系一个老客户的难度，而老客户的流失意味着收益的流失和市场占有率的下降。可以说，电信运营商的竞争就是针对客户资源的竞争。然而，客户流失从来都是无法避免的，客户流失的原因也不尽相同。若想做到客户流失前预防、流失后召回，就必须通过数据分析和建模来总结经验、预测未来。因此，本文选取了Kaggle中一个现实世界里的电信公司的客户流失数据，用其来探究电信客户流失背后的主要原因，并建立客户流失预警模型，帮助电信公司为优化业务、提高客户留存、减少流失制定策略。无论是互联网行业还是传统行业，所有产品都需要关注用户流失，用户流失原因的拆解思路也都大致相同。因此，本文的探究思路是比较具有现实的推广意义的。

2. 理解数据

2.1 理解各个字段的数据含义

序号	字段	含义	值	数据类型
1	customerID	客户id		object
2	gender	性别	Male/Female	object
3	SeniorCitizen	是否是老年人?	0,1	int64
4	Partner	是否有伴侣	Yes, No	object
5	Dependents	是否有家属	Yes, No	object
6	tenure	客户持有这家公司的时间		int64
7	PhoneService	客户是否开通了电话服务	Yes, No	object
8	MultipleLines	客户是否开通了multipleline服务	Yes，No，No phone service	object
9	InternetService	internet服务类型	DSL，Fiber Optic， No	object
10	OnlineSecurity	线上安全	Yes、No、 No internet service	object

11	OnlineBackup	是否开通onlinebackup功能	Yes、No、 No internet service	object
12	DeviceProtection	是否开通设备保护	Yes、No、 No internet service	object
13	TechSupport	是否有技术支持	Yes、No、 No internet service	object
14	StreamingTV	是否开通streaming TV功能	Yes、No、 No internet service	object
15	StreamingMovies	是否开通StreamingMovies功能	Yes、No、 No internet service	object
16	Contract	合同时长	Month-to-month,，One year, Two year	object
17	PaperlessBilling	paperless billing	Yes，No	object
18	PaymentMethod	支付方式	Electronic check,Mailed check，Bank transfer (automatic)，Credit card (automatic)	object
19	MonthlyCharges	用户每月支付费用		float64
20	TotalCharges	总支付费用		object
21	Churn	用户是否流失	Yes， No	object

2.2 数据集大小探索

共有7043条用户信息

共21列数据

3. 分析思路

任务一：探究客户流失的原因

将数据集中的标签属性划分为不同的维度，在每一个维度下，通过数据处理和数据可视化的手段，逐个探究各个因素与客户流失率之间是否具有相关性。

任务二：建立流失预警模型

处理数据并建模，对数据做特征工程处理，然后选取模型建立电信客户流失预警机制。

任务三：从业务角度和用户角度为电信公司提出建议

4. 数据清洗

4.1 缺失值

```
dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64 
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64 
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
```

4.2 重复值

```
dataset.duplicated().sum()
#0没有重复值
```

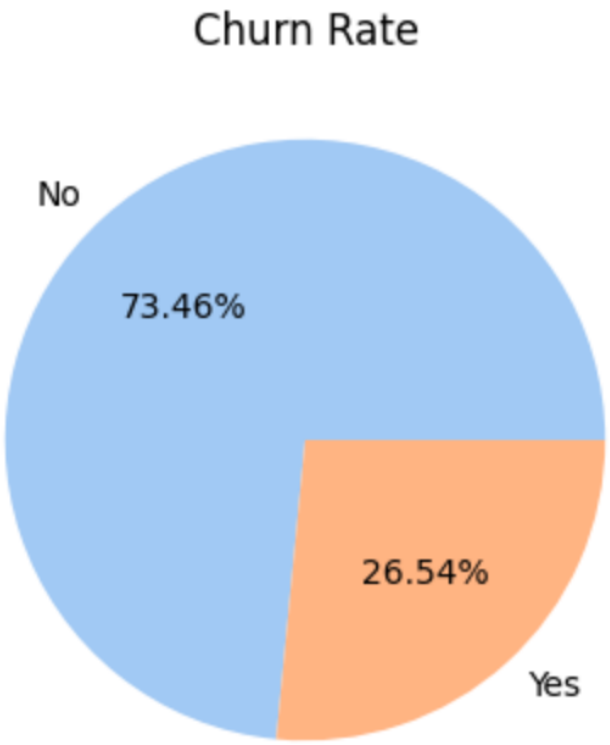
4.3 异常值

5. 数据分析

5.1 探究客户流失原因

5.1.1 客户流失比例

```
#流失客户比例
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(4, 4))

colors = sns.color_palette('pastel')
plt.pie(dataset['Churn'].value_counts(), labels=dataset['Churn'].value_counts().
plt.title('Churn Rate')
plt.show()
```

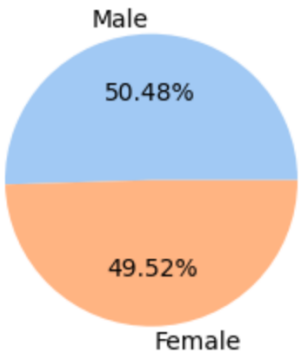


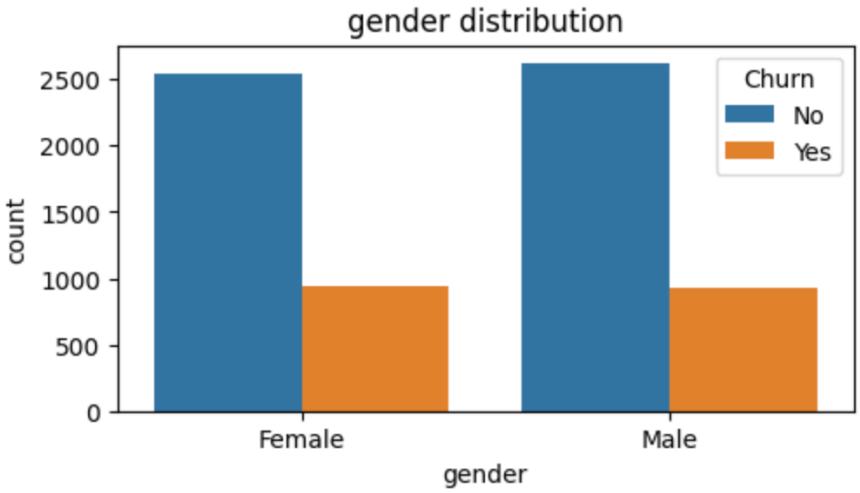
流失客户占比26.54%，未流失客户占比73.4%.

5.1.2 客人个人属性对流失率的影响

包括Gender性别，SeniorCitizen（是否为老年人）, Partner(是否有配偶）， Dependents（是否有亲属）

```
#流失率影响因素分析

#性别分布
plt.figure(figsize=(12, 6))

plt.subplot(221)
plt.pie(dataset['gender'].value_counts(), labels=dataset['gender'].value_counts(

plt.subplot(222)
gender = sns.countplot(x='gender', hue='Churn', data=dataset)
plt.title('gender distribution')

plt.show()
```

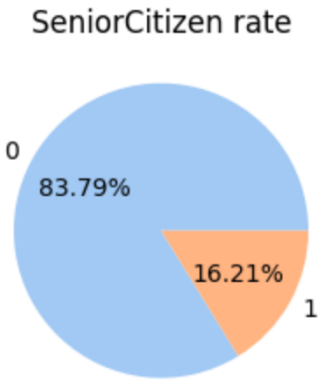


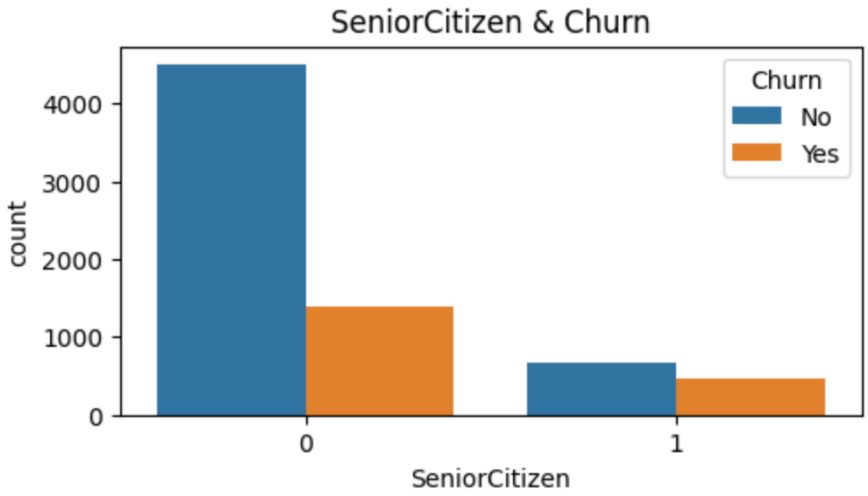


由图可知，性别对于电信企业客户流失几乎没有影响。

```
#SeniorCitizen（是否为老年人）
plt.figure(figsize=(12, 6))

plt.subplot(221)
plt.pie(dataset['SeniorCitizen'].value_counts(), labels=dataset['SeniorCitizen']
plt.title('SeniorCitizen rate')

plt.subplot(222)
gender = sns.countplot(x='SeniorCitizen', hue='Churn', data=dataset)
plt.title('SeniorCitizen & Churn')

plt.show()

#老年人的流失率比例
senior_churn_rate = dataset.loc[(dataset['SeniorCitizen']==1) & (dataset['Churn'
nosenior_churn_rate = dataset.loc[(dataset['SeniorCitizen']==0) & (dataset['Chur

```

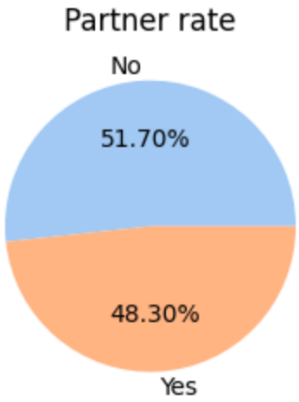
```
print('老年人流失率为: {}'.format(round(senior_churn_rate, 3)))
print('非老年人流失率: {}'.format(round(nosenior_churn_rate, 3)))

#输出:
#老年人流失率为: 0.417
#非老年人流失率: 0.236
```

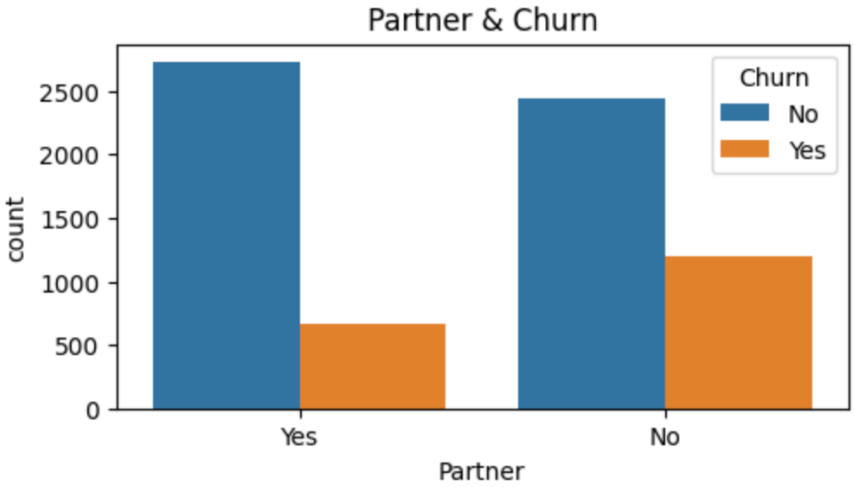




由图可知，从客户体量来说，客户群体偏向年轻化，老年人只占了16.21%。老年人的流失率为41.7%，非老年人的流失率为23.6%，老年人的流失率更大。根据现实经验，老年人在电子商品的使用上往往出现困难，尤其是在使用网络服务的时候，因此可以提取出老年人的主要开通业务情况，来验证下我们的猜想。

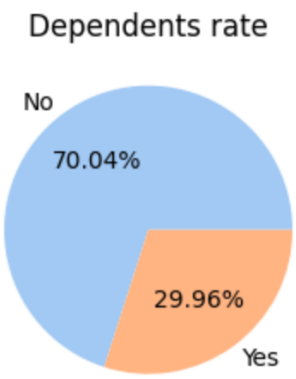
```
#是否有配偶，是否有亲属

plt.figure(figsize=(12, 6))

plt.subplot(221)
plt.pie(dataset['Partner'].value_counts(), labels=dataset['Partner'].value_count
plt.title('Partner rate')

plt.subplot(222)
gender = sns.countplot(x='Partner', hue='Churn', data=dataset)
plt.title('Partner & Churn')

plt.show()
```

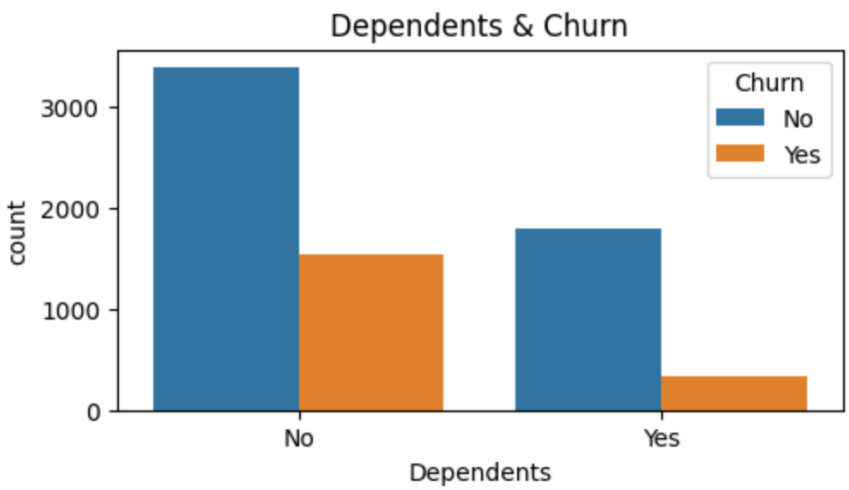




由图可知，用户是否有伴侣的人数相当，但是没有伴侣的用户流失率高。

```
#是否有亲属
plt.figure(figsize=(12, 6))

plt.subplot(221)
plt.pie(dataset['Dependents'].value_counts(), labels=dataset['Dependents'].value
plt.title('Dependents rate')

plt.subplot(222)
gender = sns.countplot(x='Dependents', hue='Churn', data=dataset)
plt.title('Dependents & Churn')

plt.show()

print('有亲属的客户流失率为: {}'.format(round(dependents_churn, 3)))
print('没有亲属的客户流失率为: {}'.format(round(no_dependents_churn, 3)))

#输出:
#有亲属的客户流失率为: 0.155
#没有亲属的客户流失率为: 0.313
```





由图可知，没有亲属的用户占比较大，流失率为31.3%, 有亲属的用户流失率为15.5%.

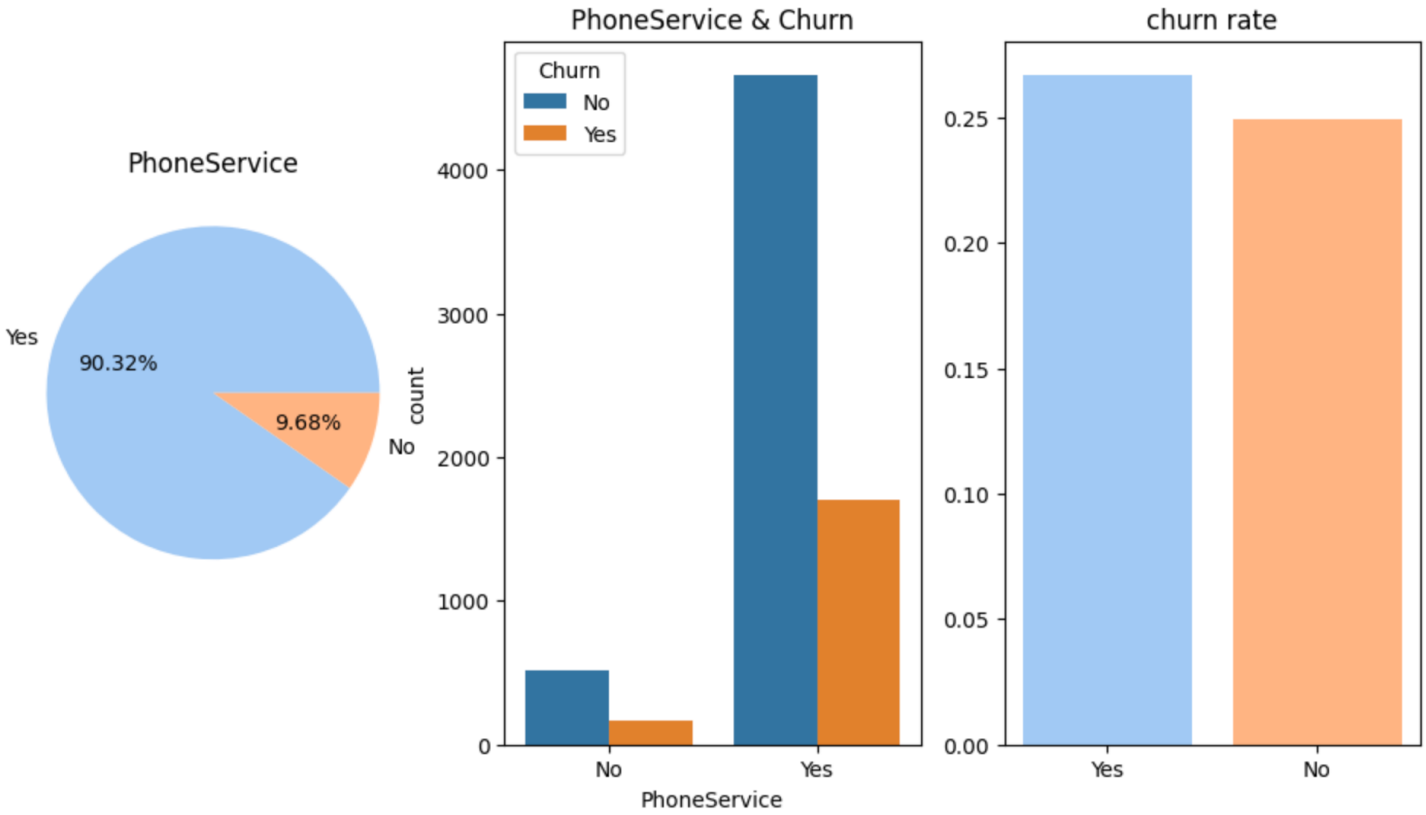
小结：

1. 性别与是否流失无关；年龄和亲属关系与是否流失有关。
2. 老年客户的流失率更高，造成这种高流失率的原因可能是老年客户存在上网困难问题
3. 有亲属的客户（即有伴侣或者子女）的客户流失率较低。

5.1.3 电信业务服务属性

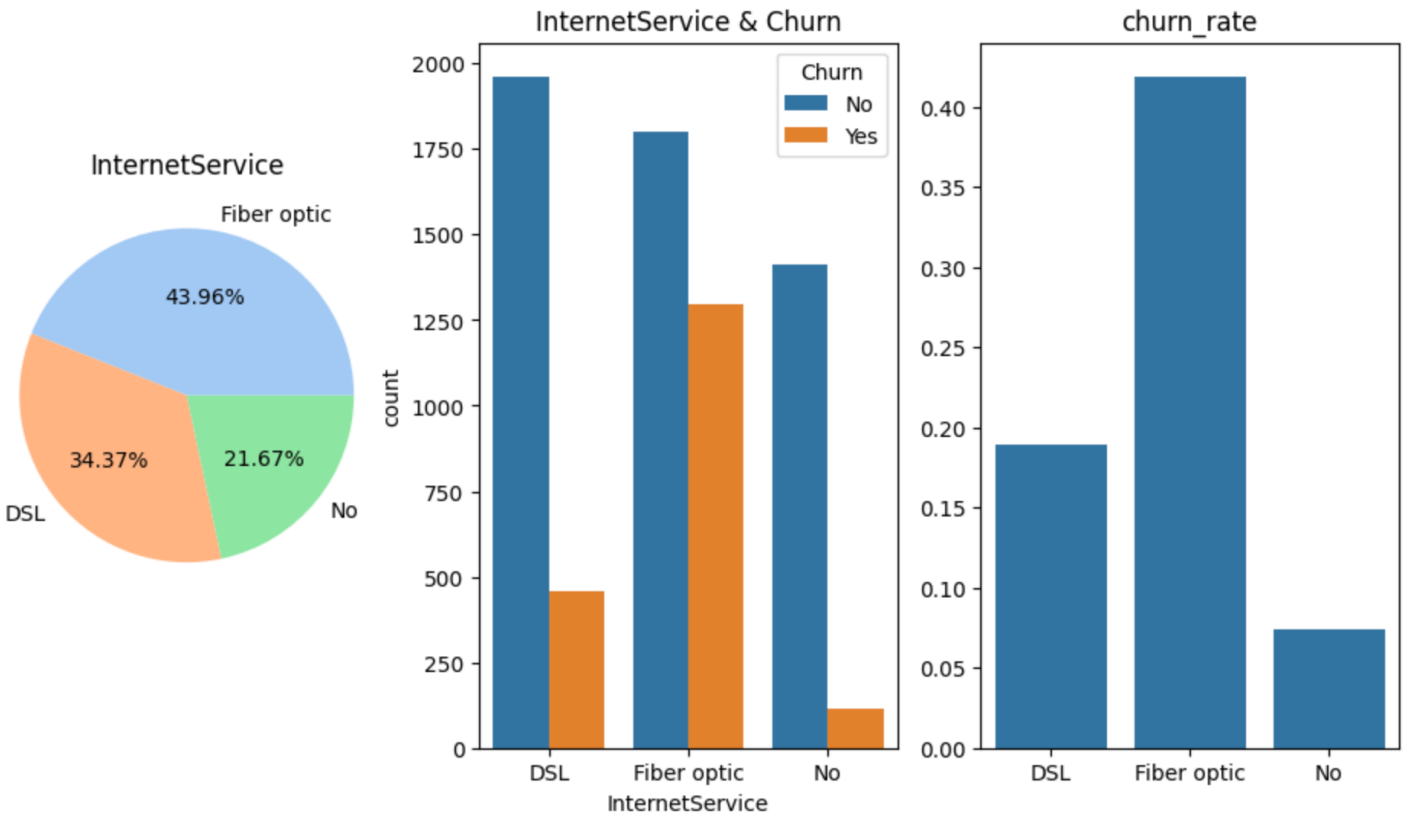
包括PhoneService（电话业务）、InternetService（互联网业务）、MultipleLines（多线业务）、OnlineSecurity（在线安全业务）、OnlineBackup（在线备份业务）、DeviceProtection（设备保护业务）、TechSupport（技术支持业务）、StreamingTV（网络电视）、StreamingMovies（网络电影）。

```
#电信业务服务属性
#电话业务
plt.figure(figsize=(12, 6))

plt.subplot(1,3,1)
plt.pie(dataset['PhoneService'].value_counts(), labels=dataset['PhoneService'].v
plt.title('PhoneService')
#我们可以看到开通电话业务的人占据了90.32%
plt.subplot(1, 3, 2)
sns.countplot(x='PhoneService', hue='Churn', data=dataset)
plt.title('PhoneService & Churn')

plt.subplot(1, 3, 3)
PhoneService_Churn_Rate  = dataset.loc[(dataset['PhoneService']=='Yes')&(dataset
PhoneService_Churn_No_Rate  = dataset.loc[(dataset['PhoneService']=='No')&(datas

x = ['Yes', 'No']
y = [PhoneService_Churn_Rate, PhoneService_Churn_No_Rate]
plt.bar(x, y, color=sns.color_palette('pastel'))
plt.title('churn rate')
plt.show()
```



从电话业务来看，90%以上的客户都开通了电话业务，开通了电话业务的客户流失率为26.71%， 未开通电话业务的客户流失率为24.93%, 两者差异不大，说明是否开通电话服务不是导致流失的原因之一。

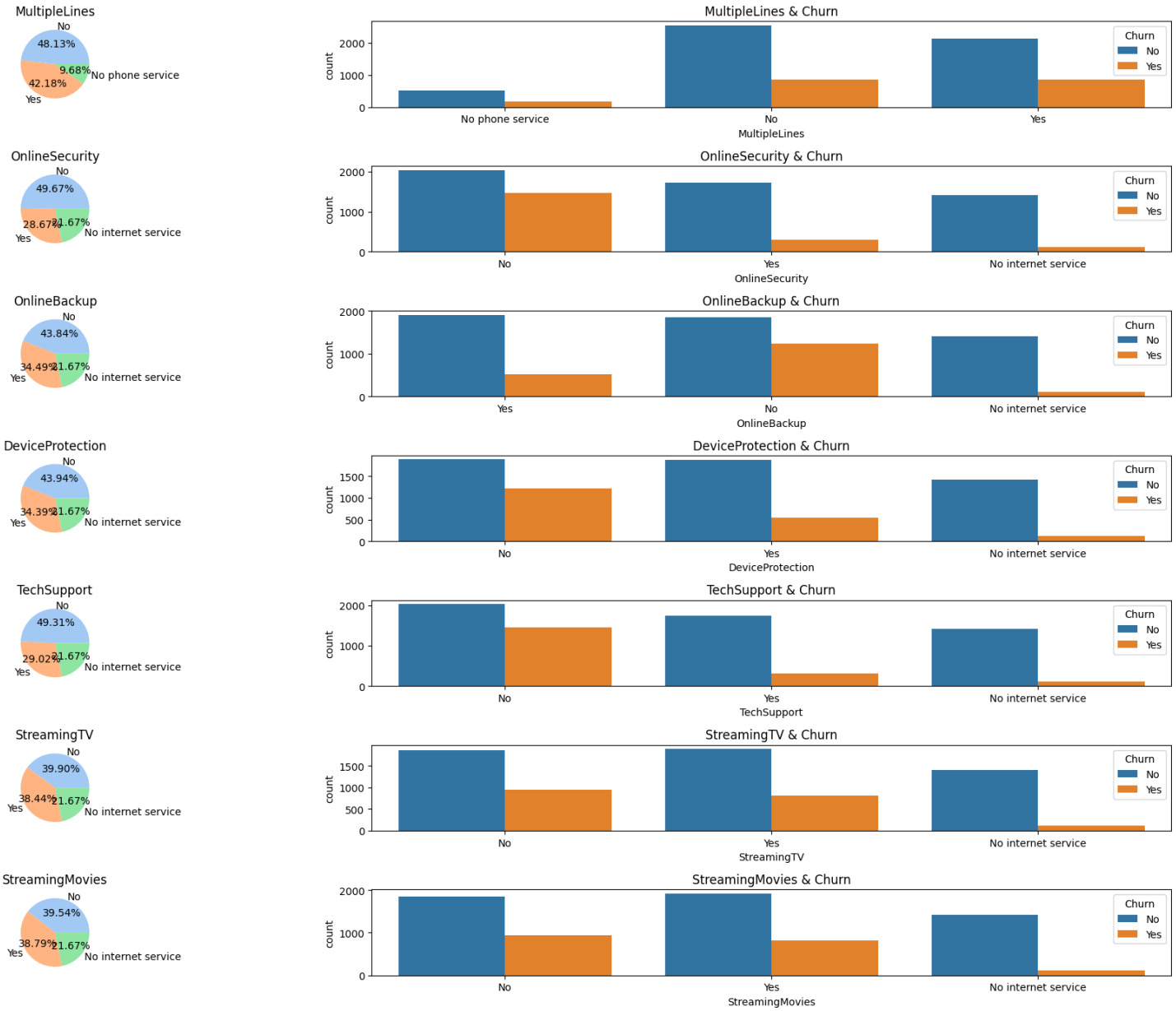
```
#互联网业务
plt.figure(figsize=(12, 6))

plt.subplot(1, 3, 1)
plt.pie(dataset.InternetService.value_counts(), labels=dataset.InternetService.v
plt.title('InternetService')

plt.subplot(1, 3, 2)
sns.countplot(x='InternetService', hue='Churn', data=dataset)
plt.title('InternetService & Churn')

plt.subplot(1, 3, 3)

DSL_Churn_Rate  = dataset.loc[(dataset['InternetService']=='DSL')&(dataset['Chur
Fiber_Churn_Rate  = dataset.loc[(dataset['InternetService']=='Fiber optic')&(dat
noInter_Churn_Rate  = dataset.loc[(dataset['InternetService']=='No')&(dataset['C

sns.barplot(x=['DSL', 'Fiber optic', 'No'], y=[DSL_Churn_Rate, Fiber_Churn_Rate,
plt.title('churn_rate')
plt.show()
```



从互联网业务来看，78.33%的用户都开通了互联网业务，其中开通Fiber optic业务的客户占比较高，但流失率也当当的高，超过40%，差不多是DSL用户流失率的2倍。

根据现实经验，Fiber Optic属于光纤，网速快且稳定，使用体验比DSL更好，然而它的流失率却相当高，因此可以断定这项业务存在一定的问题。

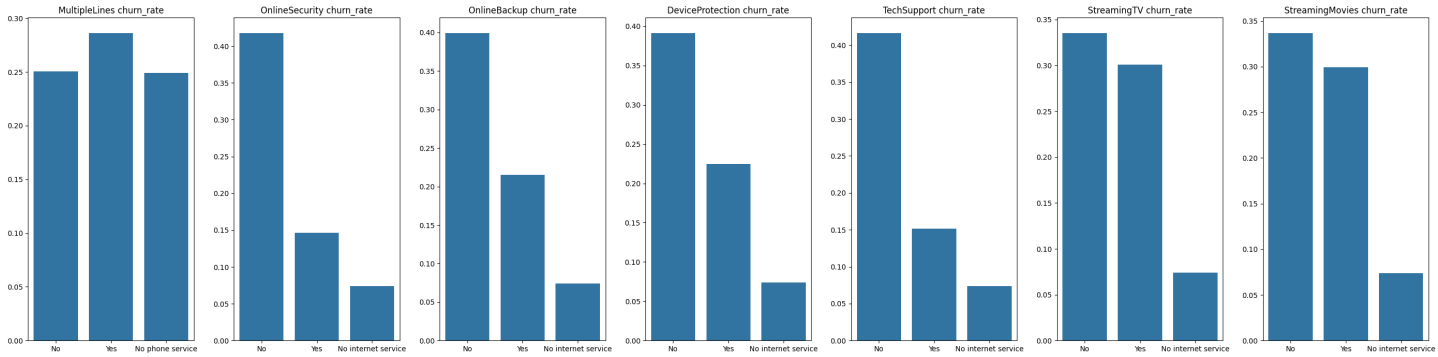
```
#从附加业务来看
plt.figure(figsize=(21, 14))

addt_service = ['MultipleLines', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtec
i = 1
for item in addt_service:
    plt.subplot(7, 2, i)
    plt.pie(dataset[item].value_counts(), labels=dataset[item].value_counts().in
    plt.title(item)

    i = i+1
    plt.subplot(7, 2, i)
    sns.countplot(data=dataset, x=item, hue='Churn')
    plt.title(item + ' & Churn')
    i = i+1

plt.tight_layout()
plt.show()
```



从饼图可以看出，基本上每一项附加业务的开通客户数都在总体的30%-40%左右，开通了在线安全，在线备份、设备保护和技术支持这4项附加业务的客户的流失率都比较低。

```
#附加业务的流失率做个比较

plt.figure(figsize=(28, 7))

addt_service = ['MultipleLines', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtec
i = 1

for item in addt_service:
    plt.subplot(1, 7, i)
    index = dataset[item].value_counts().index.tolist()
    item1_churn = dataset.loc[(dataset[item]==index[0])&(dataset['Churn']=='Yes'
    item2_churn = dataset.loc[(dataset[item]==index[1])&(dataset['Churn']=='Yes'
    item3_churn = dataset.loc[(dataset[item]==index[2])&(dataset['Churn']=='Yes'

```

```
    sns.barplot(x=[index[0], index[1], index[2]], y=[item1_churn, item2_churn, i
    i = i+1
    plt.title(item + ' churn_rate')

plt.tight_layout()
plt.show()
```



未开通在线安全服务、在线备份服务、在线保护、在线支持的流失率是开通这些服务的客户的流失率的2倍，说明这4项附加业务的开通确实可以显著的减少客户流失。此外，涉及互联网服务支持的网络电视，电影业务的流失率相对高，又一次印证了公司的互联网服务的确存在着比较大的问题。

小结：

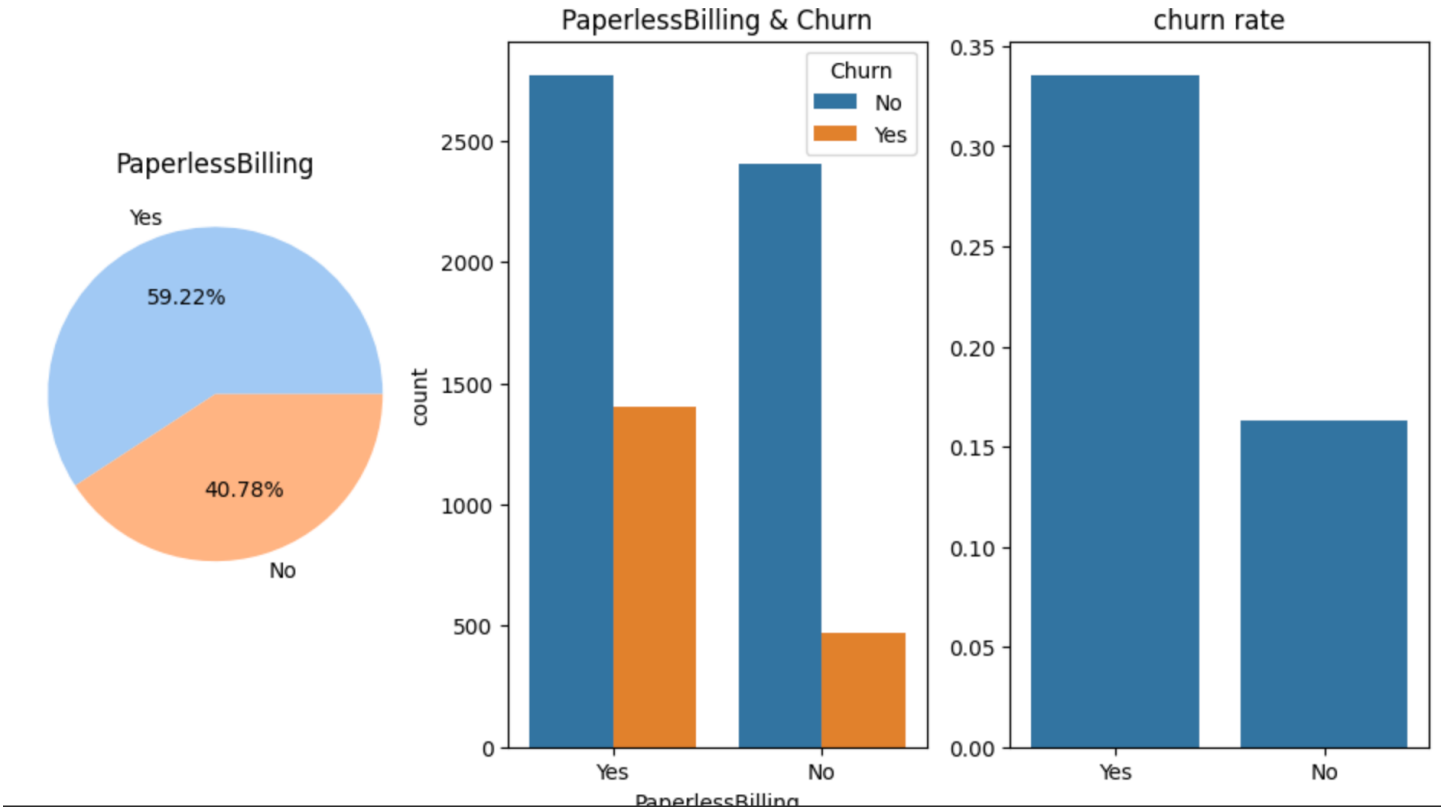
1. 是否开通电话业务对流失率的影响不大。
2. 互联网业务中，78.33%的用户都开通了互联网业务，其中开通Fiber optic业务的客户占比较高，但流失率也当当的高，超过40%，差不多是DSL用户流失率的2倍。根据现实经验，Fiber Optic属于光纤，网速快且稳定，使用体验比DSL更好，然而它的流失率却相当高，因此可以断定这项业务存在一定的问题。
3. 附加业务当中，开通在线安全服务、在线备份服务、在线保护、在线支持这4项服务的用户，比未开通的用户流失率较低，可以通过多项附加业务的补充，将流失率有效降低。

5.1.4 客户消费行为属性

包括PaperlessBilling（账单形式）、PaymentMethod（支付方式）、Contract（合同签订方式）、tenure（在网时长）、MonthlyCharges（月租费）、TotalCharges（总费用）。

```
#从账单形式来看
plt.figure(figsize=(12, 6))

plt.subplot(1, 3, 1)
plt.pie(dataset['PaperlessBilling'].value_counts(), labels=dataset['PaperlessBil
plt.title('PaperlessBilling')

plt.subplot(1, 3, 2)
sns.countplot(data=dataset, x='PaperlessBilling', hue='Churn')
```

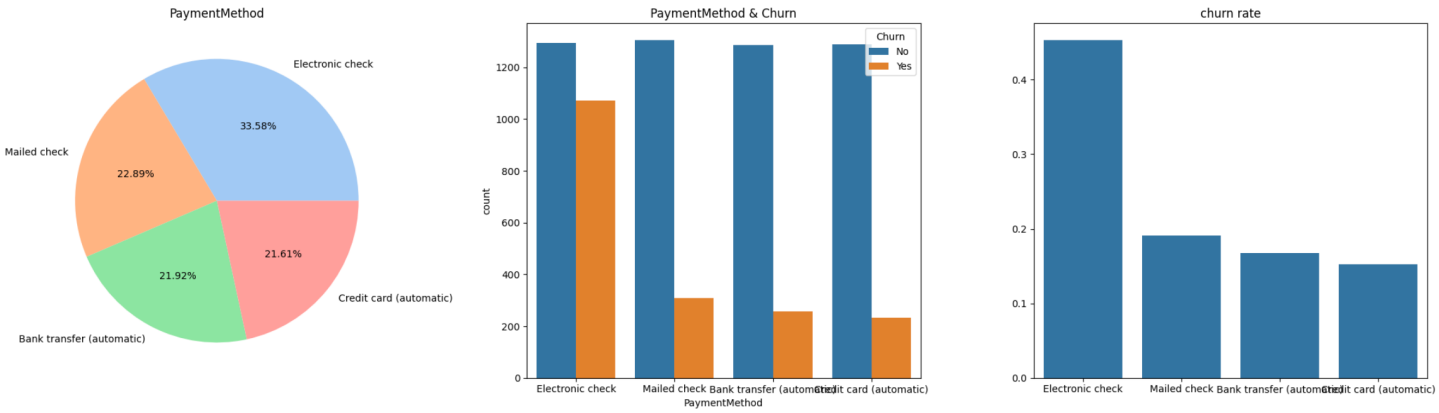
```
plt.title('PaperlessBilling & Churn')

plt.subplot(1, 3, 3)
paperless_yes_Rate  = dataset.loc[(dataset['PaperlessBilling']=='Yes')&(dataset[
paperless_no_Rate  = dataset.loc[(dataset['PaperlessBilling']=='No')&(dataset['C

sns.barplot(x=['Yes', 'No'], y=[paperless_yes_Rate, paperless_no_Rate])
plt.title('churn rate')

plt.show()
```



从账单形式来看，使用电子账单服务的客户占比较大，流失率也更高。可能是因为电子账单比较容易忽视，且安全性和私密性较差，带给客户的体验不佳。

```
#支付方式
plt.figure(figsize=(21, 6))

plt.subplot(1, 3, 1)
plt.pie(dataset['PaymentMethod'].value_counts(), labels=dataset['PaymentMethod']
plt.title('PaymentMethod')

plt.subplot(1, 3, 2)
sns.countplot(data=dataset, x='PaymentMethod', hue='Churn')
plt.title('PaymentMethod & Churn')

plt.subplot(1, 3, 3)
```

```
method1  = dataset.loc[(dataset['PaymentMethod']=='Electronic check')&(dataset['
method2  = dataset.loc[(dataset['PaymentMethod']=='Mailed check')&(dataset['Chur
method3  = dataset.loc[(dataset['PaymentMethod']=='Bank transfer (automatic)')&(
method4  = dataset.loc[(dataset['PaymentMethod']=='Credit card (automatic)')&(da

sns.barplot(x=['Electronic check', 'Mailed check', 'Bank transfer (automatic)',
plt.title('churn rate')

plt.tight_layout()
plt.show()
```

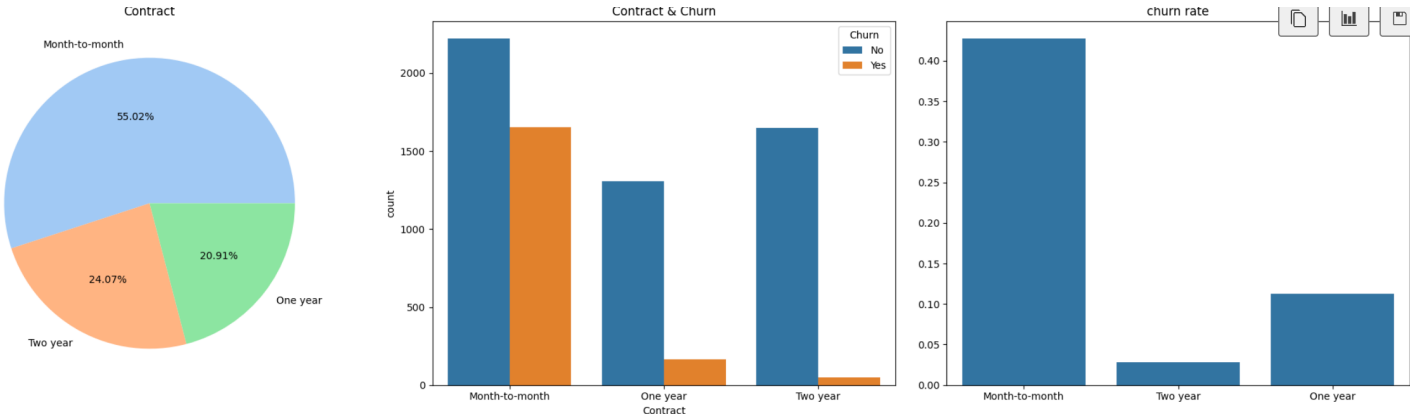


从支付方式来看，使用电子账单支付的客户流失率远远高于其他方式，这可能是因为电子账单的接纳度不够高。

相比之下，使用银行或者信用卡自动转账的客户的流失率相对较低，推测是因为这两种方式更加便捷，到期自动扣款，无需用户人工操作。因此，应当引导客户采取其他三种方式进行支付，尤其是自动转账的方式。

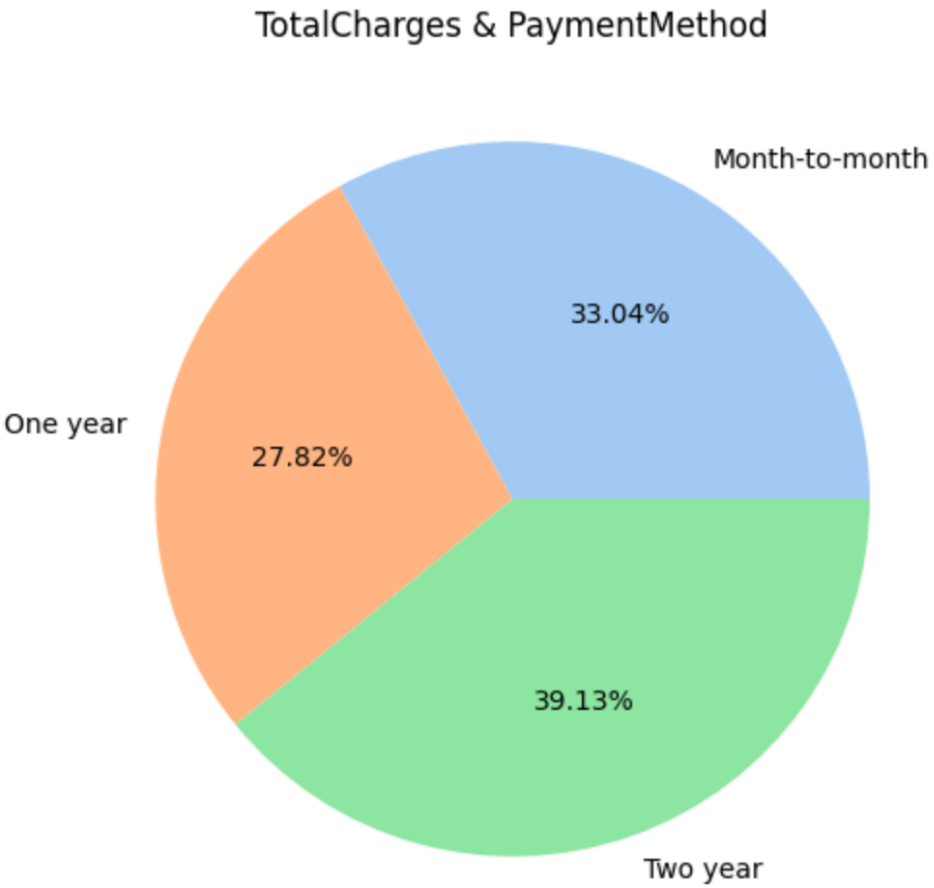
```
#从合同签订方式来看
plt.figure(figsize=(21, 6))

plt.subplot(1, 3, 1)
plt.pie(dataset['Contract'].value_counts(), labels=dataset['Contract'].value_cou
plt.title('Contract')

plt.subplot(1, 3, 2)
sns.countplot(data=dataset, x='Contract', hue='Churn')
plt.title('Contract & Churn')

plt.subplot(1, 3, 3)
method1  = dataset.loc[(dataset['Contract']=='Month-to-month')&(dataset['Churn']
method2  = dataset.loc[(dataset['Contract']=='Two year')&(dataset['Churn']=='Yes
method3  = dataset.loc[(dataset['Contract']=='One year')&(dataset['Churn']=='Yes

sns.barplot(x=['Month-to-month', 'Two year', 'One year',], y=[method1, method2,
```

```
plt.title('churn rate')

plt.tight_layout()
plt.show()
```



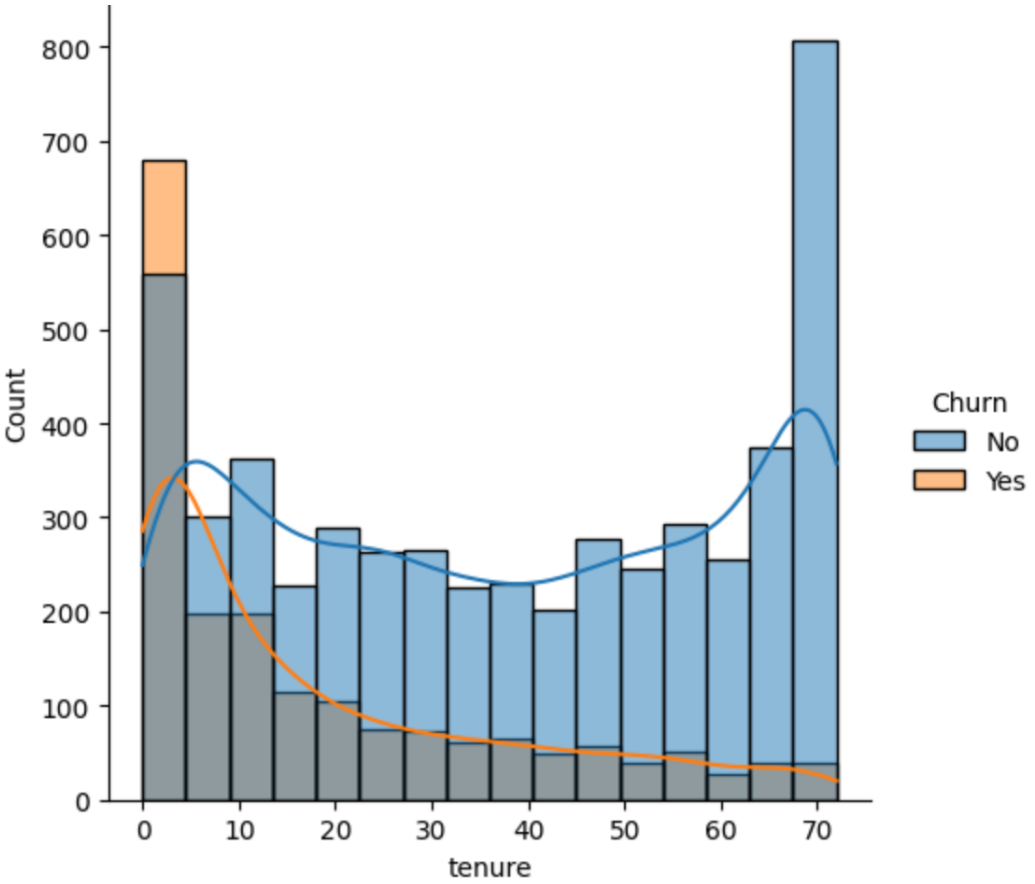
从签订合同的方式来看，按月签订的方式占比更大，流失率也远远高于其他两种方式。两年一签的流失率最低，一年一签的流失率也较低，可见，合同签订方式对流失率的影响较大，合同期限越长，客户的流失率越低，客户的黏性也越大。

```
#dataset[dataset['TotalCharges']==' ']
#发现有11名客户的TotalCharges列为空值，观察发现，这些客户的tenure字段值为0，表明他们的在
#总费用是str类型，所以，我们需要转换成float类型
#将这些用户的tenure值统一设置为1

#替换
dataset.loc[dataset['TotalCharges']==' ', 'TotalCharges']=dataset.loc[dataset['T
#类型转换
dataset['TotalCharges'] = dataset.TotalCharges.apply(lambda s: float(s))
data_group = dataset.groupby('Contract').TotalCharges.sum()
data_group
#做图
plt.figure(figsize=(6, 6))
plt.pie(data_group.values, labels=data_group.index, colors=colors, autopct='%2.2
plt.title('TotalCharges & PaymentMethod')
```



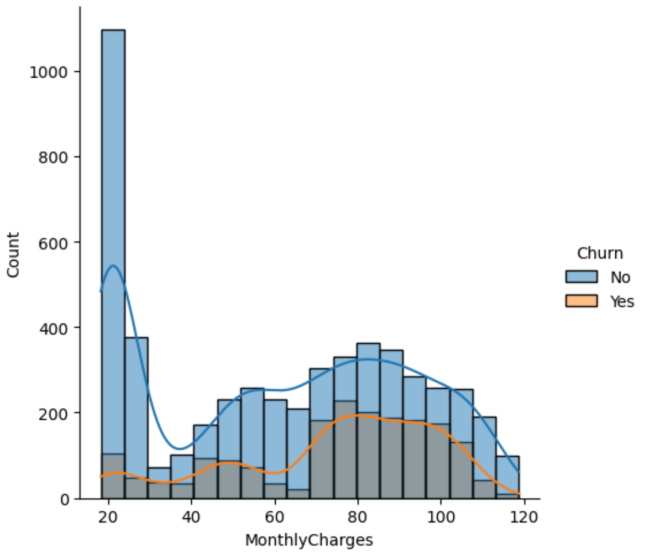
通过合同签订期限与总费用之间的环形图，可以明显看到，签订长期合同的用户贡献了60%以上的收入，也再一次表明，将月租户逐渐发展为年租户应当成为公司的一项重要发展策略。

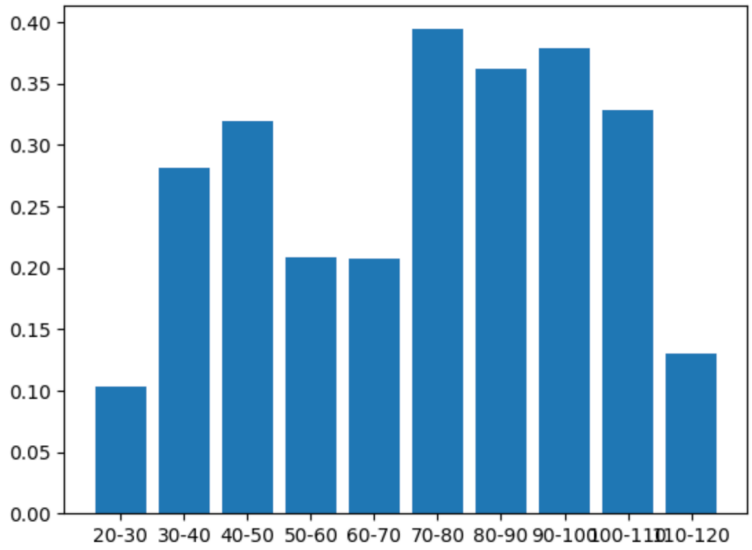
```
#从在网时长来看
dataset.tenure.value_counts()
sns.distplot(dataset['tenure'], kde=True,)

sns.displot(x='tenure', hue='Churn', data=dataset, kde=True)
```



密度曲线表明，客户的流失率随着在网时长的增加而逐渐减少，在网时间越长，说明客户黏性越大，更不容易流失。此外，在网的第20个月是客户是否流失的分水岭，在第3个月左右，客户有着最高的流失率，因此应当在入网的前3个月尽可能的让新客户感受到业务的价值在所在，在第20个月以后，客户的流失率越来越低，此时公司已经拥有了稳定的客户群。

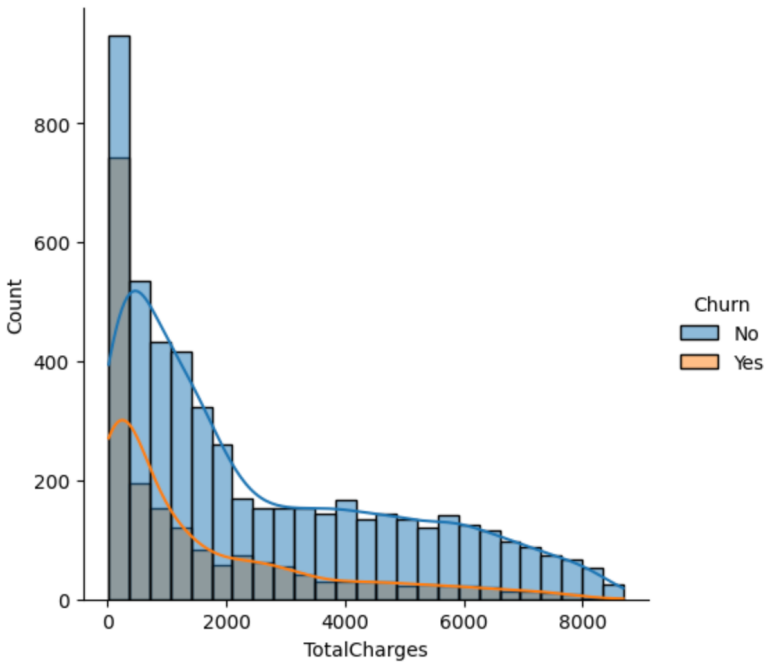
```
#从月租和总费用来看
#sns.displot(x='MonthlyCharges', hue='Churn', data=dataset, kde=True)
#可以看到，月租费在70-100元内的客户更容易流失，可以具体看下，这段费用下的用户流失率

bins = [20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120]
labels = ['20-30' ,'30-40' ,'40-50','50-60' ,'60-70' ,'70-80', '80-90', '90-100'

dataset[ 'Monthly_Bins'] = pd.cut(x=dataset['MonthlyCharges'], bins=bins, labels
churn_rate = []
for label in labels:
    total_cus = dataset[dataset['Monthly_Bins']==label].customerID.count()
    churn_cus = dataset.loc[(dataset['Monthly_Bins']==label) & (dataset['Churn']
    churn_rate.append(round(churn_cus/total_cus, 3))

plt.bar(labels, churn_rate)
plt.tight_layout()
plt.show()
```





从月租来看，月租费在[70， 100]内的客户更容易流失，月租费在70-80元的客户流失率为39.82%，80-90元的客户流失率为36.12%，90-100元的客户流失率为37.80%对于这部分用户，可以通过调研形式询问流失原因，若是由于月费价格高昂，则可以通过优惠券或减免形式对这一区间的在网客户进行补贴，以降低流失率。

```
#从总费用来看
sns.displot(x='TotalCharges', hue='Churn', data=dataset, kde=True)
```



通过总费用我们可以看出，流失率随着总费用的增长而不断降低，比较容易理解，总费用在2000以内的用户流失率最高，这再一次证明，引导客户提高消费额度，延长客户合同存续期是提高客户留存，减少流失的不二之选。

小结：

1. 客户的消费行为背后隐藏着他们的消费偏好，通过数据发现，客户更习惯纸质的账单形式和自动转账的支付方式，而使用电子账单和电子支付的客户的流失率则比较高。

2. 客户在网时间越长，总费用越高，流失率越低，客户黏性越大。月租费在70-100元间的客户、总费用低于2000元的客户更容易流失。易流失客户的生命周期通常为1-3个月，而生命周期达到67个月的则为高度忠诚用户，对这两种客户应当采取不同的留存和维系策略。
3. 延长客户的合同期限，推动客户从月签转向年签，引导客户提高消费额度，是提高留存的重要策略。

6. 数据预处理

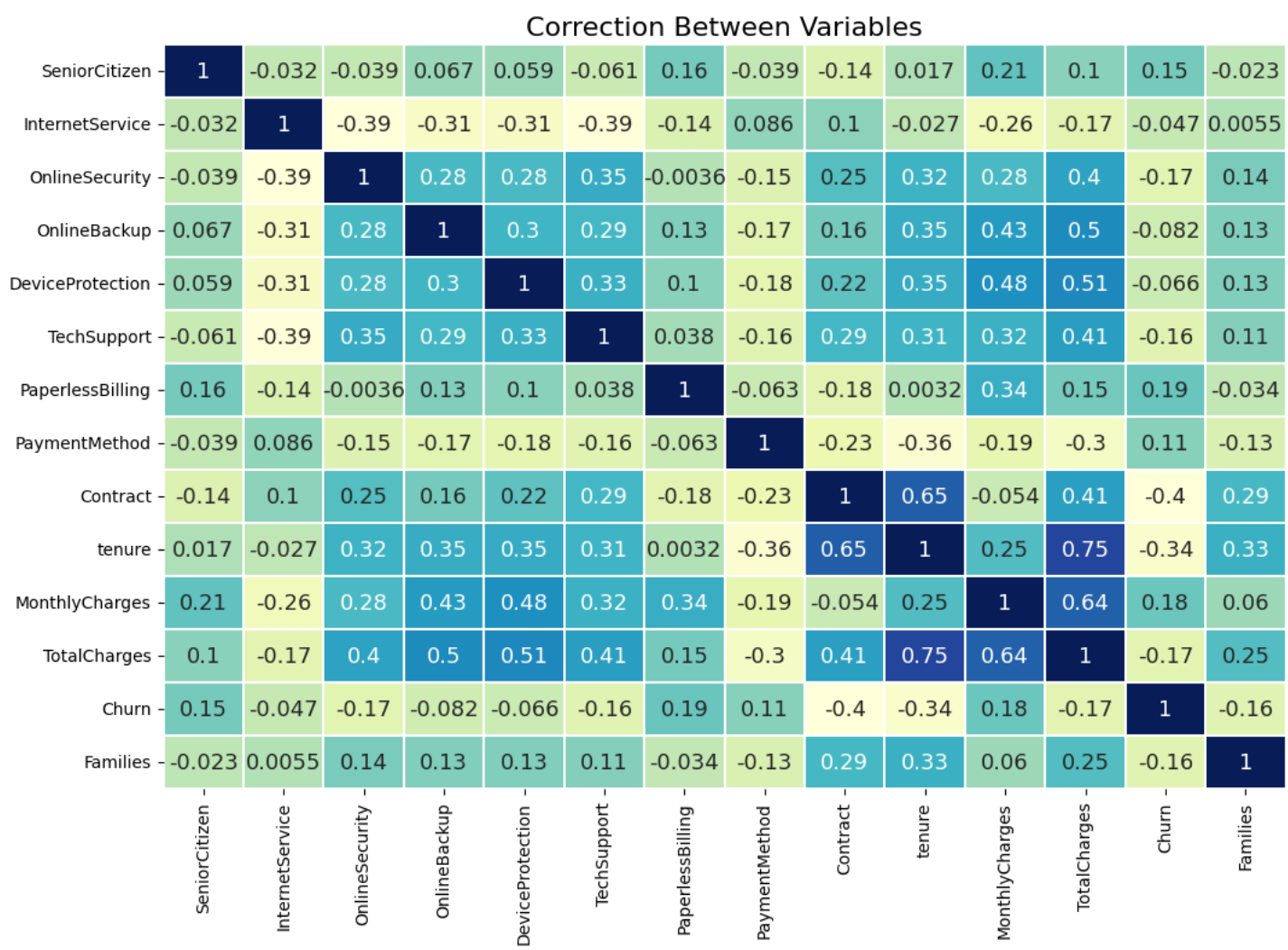
```
#特征工程
#电信客户流失预警模型
#数据预处理
#根据上述分析，手工选取了以下12个与流失率较为相关的特征
#SeniorCitizen,  Partner, Dependents,InternetService, 'OnlineSecurity', 'OnlineB
#PaperlessBilling, PaymentMethod, Contract, tenure, MonthlyCharges, TotalCharges

import warnings

warnings.filterwarnings('ignore')
data_df = dataset[['SeniorCitizen', 'Partner', 'Dependents', 'InternetService',

#对这些特别分类进行处理
data_df['M_Partner'] = data_df['Partner'].map({'Yes': 1, 'No': 0})
data_df['M_Dependents'] = data_df['Dependents'].map({'Yes': 1, 'No': 0})
data_df['M_Families'] = data_df['M_Partner'] + data_df['M_Dependents']
data_df['Families'] = data_df['M_Families'].apply(lambda x: 0 if x==0 else 1)

#综合之前的结果来看，OnlineSecurity， OnlineBackup， DeviceProtection， TechSupport
#它和No的效果是一样的，可以用no代替
data_df.replace(to_replace={'No internet service': 'No'}, inplace=True)
for item in ['OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport'
    data_df[item] = data_df[item].map({'Yes': 1, 'No': 0})

data_df['tenure'] = pd.cut(data_df['tenure'], bins=5, right=False)
data_df['tenure'] = data_df['tenure'].astype('str')
data_df['tenure'] = data_df['tenure'].map({'[0.0, 14.4)': 0, '[14.4, 28.8)': 1,

data_df['MonthlyCharges'] = pd.cut(data_df['MonthlyCharges'], bins=5, right=Fals
data_df['MonthlyCharges'] = data_df['MonthlyCharges'].astype('str')
data_df['MonthlyCharges'] = data_df['MonthlyCharges'].map({'[18.25, 38.35)': 0,

data_df['TotalCharges'] = pd.cut(data_df['TotalCharges'], bins=5, right=False)
data_df['TotalCharges'] = data_df['TotalCharges'].astype('str')
data_df['TotalCharges'] = data_df['TotalCharges'].map({'[18.8, 1752.0)': 0, '[17

```

```
#PaperlessBilling
data_df['PaperlessBilling'] = data_df['PaperlessBilling'].map({'Yes': 1, 'No': 0
#PaymentMethod
data_df['PaymentMethod'] = data_df['PaymentMethod'].map({'Electronic check': 2,
#Contract
data_df['Contract'] = data_df['Contract'].map({'Month-to-month': 0, 'One year':
#churn
data_df['Churn'] = data_df['Churn'].map({'Yes': 1, 'No': 0})
#InternetService
data_df['InternetService'] = data_df['InternetService'].map({'DSL': 0, 'Fiber op

data_df.drop(['Partner', 'Dependents', 'M_Partner', 'M_Dependents', 'M_Families'
data_df.info()
```

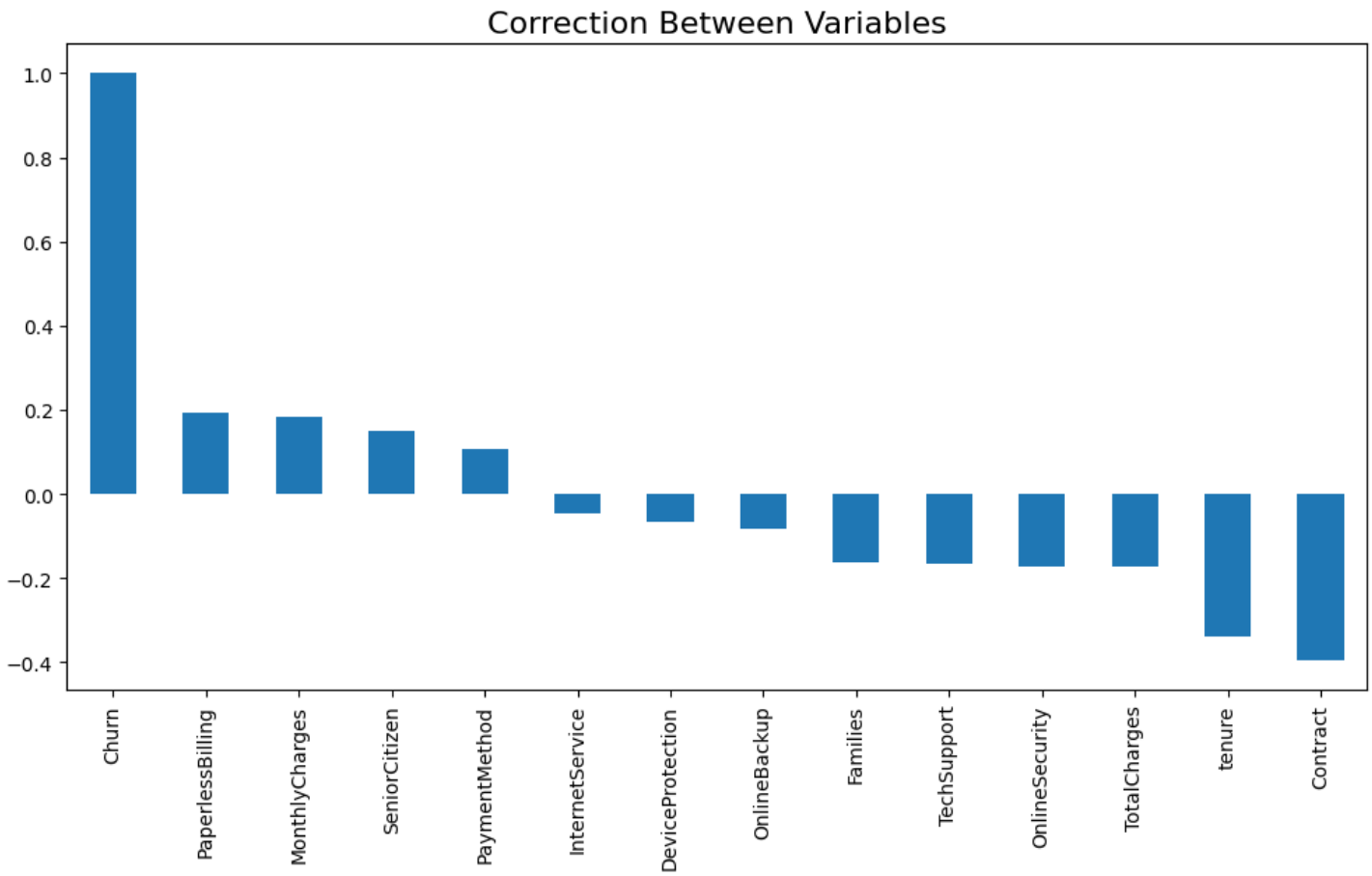
得到处理后的数据：

	SeniorCitizen	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	PaperlessBilling	PaymentMethod	Contract	tenure	MonthlyCharges	TotalCharges	Churn	Families
0	0	0	0	1	0	0	1	2	0	0	0	0	0	1
1	0	0	1	0	1	0	0	3	1	2	1	1	0	0
2	0	0	1	1	0	0	1	3	0	0	1	0	1	0
3	0	0	1	0	1	1	0	0	1	3	1	1	0	0
4	0	1	0	0	0	0	1	2	0	0	2	0	1	0

查看这些数据之间的相关性



颜色越深，代表相关性越强，由图可知，租期和合同、总费用和租期，月租和总费用之间也存在较强的相关性。



7. 建立模型

7.1 建立测试集和训练集

```
#建立测试集和训练集
#由于我们所使用的数据集是不平衡的，所以最好使用交叉验证法来确保训练集和测试集都包含每个样本

from sklearn.model_selection import train_test_split

y = data_df['Churn']
X = data_df.drop(['Churn'], axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8, random
```

7.2 选择机器学习算法

```
#该类型属于有监督的分类算法，这里我选择了随机森林和逻辑回归做预测
from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, roc_curve, auc
```

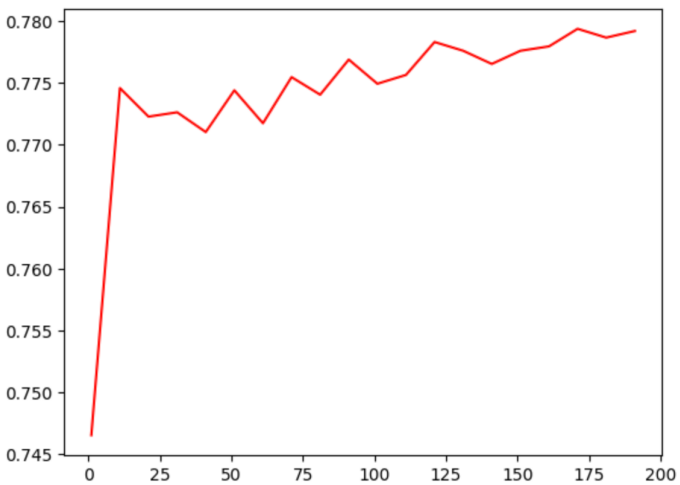
7.3 训练模型

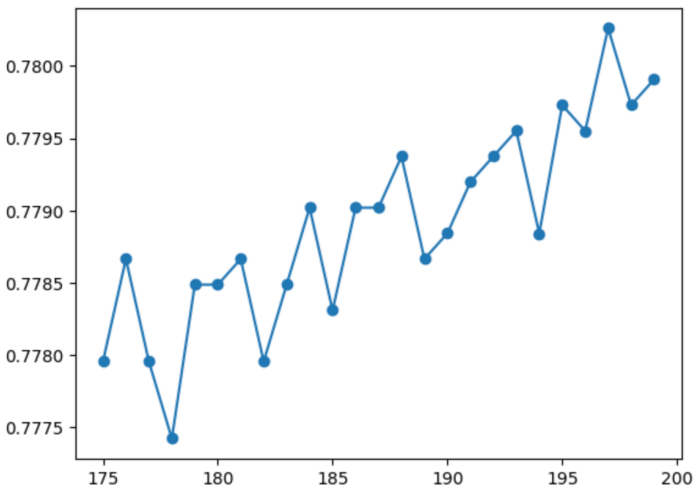
7.3.1 随机森林模型

```
#随机森林算法
rfc = RandomForestClassifier(n_estimators=100, random_state=90)
#训练模型
res = rfc.fit(X_train, y_train)

#预测模型
y_pred = rfc.predict(X_test)
score_rfc = accuracy_score(y_test, y_pred)
print('模型准确率: ', score_rfc)

#交叉验证
score_cross_pred = cross_val_score(rfc, X_train, y_train, cv=10).mean()
print('交叉验证得分为: {}'.format(score_cross_pred))

#调参-迭代器的数量
score_lst_rfc = []
for i in range(0, 200, 10):
    rfc=RandomForestClassifier(n_estimators=i+1, random_state=90)
    score=cross_val_score(rfc, X_train, y_train, cv=10).mean()
    score_lst_rfc.append(score)

print('最大得分: {}'.format(max(score_lst_rfc)))
print('子树数量为: {}'.format(score_lst_rfc.index(max(score_lst_rfc))*10+1))

#绘制学习曲线
x = np.arange(1, 201, 10)
plt.subplot(111)
plt.plot(x, score_lst_rfc, 'r-')
plt.show()

#结果
#模型准确率:  0.7700496806245565
#交叉验证得分为: 0.7752900495068215
#最大得分: 0.7793753070556668
#子树数量为: 171

#优化1，增加迭代器的数量
score_lst_rfc = []
```

```python
for i in range(175, 200):
    rfc = RandomForestClassifier(n_estimators=i, random_state=90)
    score = cross_val_score(rfc, X_train, y_train, cv=10).mean()
    score_lst_rfc.append(score)

print('最大得分: {}'.format(max(score_lst_rfc)), '子树数量: {}'.format(score_lst_r

#绘制学习曲线
x = np.arange(175, 200)
plt.subplot(111)
plt.plot(x, score_lst_rfc, 'o-')
plt.show()

#结果:
#最大得分: 0.7802618318783618 子树数量: 197

#优化2-max_depth
from sklearn.model_selection import GridSearchCV

rfc = RandomForestClassifier(n_estimators=197, random_state=90)
param_grid={'max_depth': np.arange(1, 100)}
gs = GridSearchCV(rfc, param_grid, cv=10)
gs.fit(X_train, y_train)

best_params = gs.best_params_
best_score = gs.best_score_

print("best_params: {}".format(best_params), 'best_score: {}'.format(best_score)
#此步骤可以看出，树的最大深度为7时，预测的精度可以达到0.80

#结果
#best_params: {'max_depth': 7} best_score: 0.8008484184271192

#优化3-max_features
rfc = RandomForestClassifier(n_estimators=197, random_state=90, max_depth=7)

param_grid = {'max_features': np.arange(3, 16)}
gs = GridSearchCV(rfc, param_grid, cv=10)
gs.fit(X_train, y_train)

best_params = gs.best_params_
best_score = gs.best_score_
print('best_params: {}'.format(best_params), 'best_score: {}'.format(best_score)

#可以看到，随机森林模型的准确率有所上升

#结果:
```

```
#best_params: {'max_features': 6} best_score: 0.8017355731075921

#最优参数下的准确率
rfc = RandomForestClassifier(n_estimators=197, random_state=90, max_depth=7, max
rfc.fit(X_train, y_train)

y_pred = rfc.predict(X_test)
accuracy_score_rfc = accuracy_score(y_test, y_pred)
print("模型的准确率为: {}".format(accuracy_score_rfc))

y_prob = rfc.predict_proba(X_test)[:, 1]
fpr_rfc, tpr_rfc, threshold_rfc = roc_curve(y_test, y_prob)
auc_rfc = auc(fpr_rfc, tpr_rfc)

score_cvs = cross_val_score(rfc, X_train, y_train, cv=10).mean()
print("交叉验证得分为: {}".format(score_cvs))

#结果:
#模型的准确率为: 0.7920511000709723
#交叉验证得分为: 0.8017355731075921
```





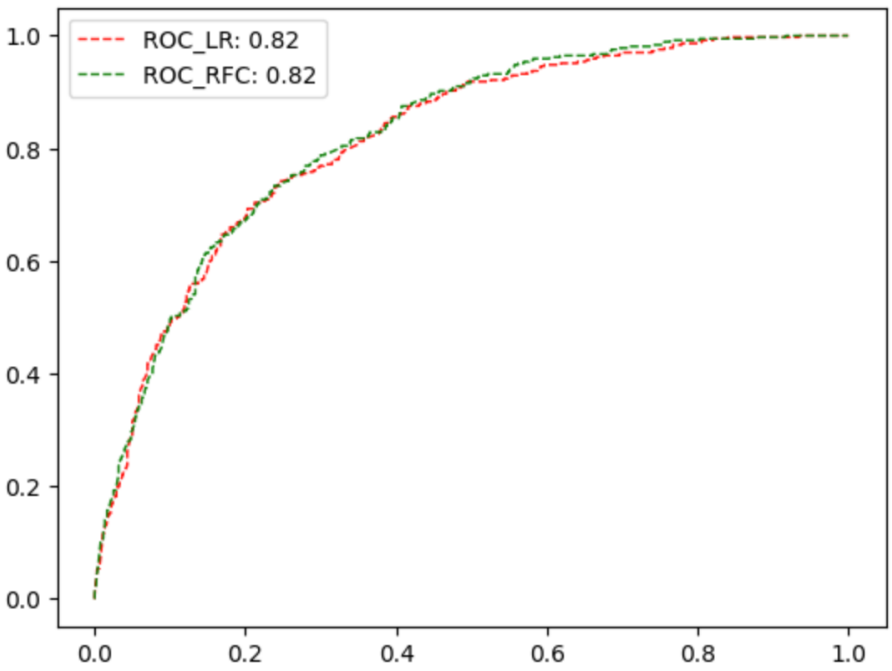
7.3.2 逻辑回归模型

```
#逻辑回归模型
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression()
lr.fit(X_train, y_train)

y_pred = lr.predict(X_test)
accuracy_score_lr = accuracy_score(y_test, y_pred)
print('模型的准确率为: {}'.format(accuracy_score_lr))

```

```
y_prob = lr.predict_proba(X_test)[:, 1]
fpr_lr, tpr_lr, threshold_lr = roc_curve(y_test, y_prob)
auc_lr = auc(fpr_lr, tpr_lr)
print('AUC_LR得分: {}'.format(auc_lr))

#结果:
模型的准确率为: 0.7885024840312278
AUC_LR得分: 0.8166583134945495
```

7.4 评估模型

```
#绘制两个模型的ROC曲线-混淆矩阵在前
fig, ax = plt.subplots()
l1 = ax.plot(fpr_lr, tpr_lr, 'r--', label='ROC_LR: {0:.2f}'.format(auc_lr), lw=1
l2 = ax.plot(fpr_rfc, tpr_rfc, 'g--', label='ROC_RFC: {0:.2f}'.format(auc_rfc),

plt.legend()
plt.show()

#绘制两个模型的ROC曲线，我们可以看到两个模型的准确率都在82分左右，其中，逻辑回归的模型表现
```



7.5 结果预测

```
#结果预测,由于目前没有提供数据集，这里我们选择后10行作为需要预测的数据集
id = dataset.tail(10)['customerID']
pre_x = data_df.drop(['Churn'], axis=1).tail(10)
```

```
pre_y = lr.predict(pre_x)

df = pd.DataFrame({'customerID': id, 'Churn': pre_y})
print(df)

```

```
      customerID  Churn
7033  9767-FFLEM      0
7034  0639-TSIQW      0
7035  8456-QDAVC      1
7036  7750-EYXWZ      0
7037  2569-WGERO      0
7038  6840-RESVB      0
7039  2234-XADUH      0
7040  4801-JZAZL      0
7041  8361-LTMKD      1
7042  3186-AJIEK      0
```

8. 结论与建议

针对此研究，我们将目前电信行业客户流失的原因大体分为3类，分别为客户属性、服务水平、业务费用。先就这3大原因进行如下分析。

a. 根据客户个人属性分群制定策略

从数据来看，老年客户可能因为使用存在困难而流失，而技术支持这项附加业务则可以有效改善这种情况。对此，可以开通老年客户电信服务专线，定期线上回访，进一步拉近与老年客户群体之间的距离，提高老年客户的留存率。

对于有亲属的客户，可以退出包含主要业务和所有附加业务的家庭年费套餐，给予一定的折扣优惠。也可以开发“家庭服务”业务，同城亲属之间通话免费等。

b. 提高服务质量，精准定位客户需求

从客户消费偏好来看，多数客户更倾向于传统支付方式，尚未习惯使用电子支付。因此要大力推荐并引导新老用户使用这两种方式，进而延长客户的存续期。

附加业务当中，开通在线安全服务、在线备份服务、在线保护、在线支持这4项服务的用户，比未开通的用户流失率较低，可以通过多项附加业务的补充，将流失率有效降低。

从电信业务数据来看，目前互联网服务，尤其是光纤方面仍然存在缺陷，对其进行技术改造和升级迫在眉睫。

c. 业务费用

延长客户的合同期限，推动客户从月签转向年签，引导客户提高消费额度，是提高留存的重要策略。

另外，易流失客户的生命周期通常为1-3个月，而生命周期达到67个月的则为高度忠诚用户。对于易流失用户，可以在办理新业务时，给予折扣优惠福利，签订的合同期限越长，给予的折扣优惠力度越大。高度忠诚的用户，需要定期维系和关怀，比如向客户赠送节日礼物等，忠诚客户的个性化需求也要满足。