基于机器学习方法的电信用户流失的挖掘分析报告

目　录

一、项目背景 1
二、运行环境 1
三、技术路线 1
四、数据准备 1
4.1 源数据描述 1
4.2 数据预处理 7
4.3 数据预处理结果展示 10
4.4 探索性数据分析 11
五、建模分析 19
5.1 数据划分 19
5.2 绘制热力图观察变量之间的相关性 19
5.3 绘制 churn 与变量关系图 20
5.4 特征选择 20
5.5 模型选择与参数调优 20
5.6 评估策略 21
5.7 运行结果展示与分析 22
5.8 分析总结 27
六、创新性 29
七、项目总结 29

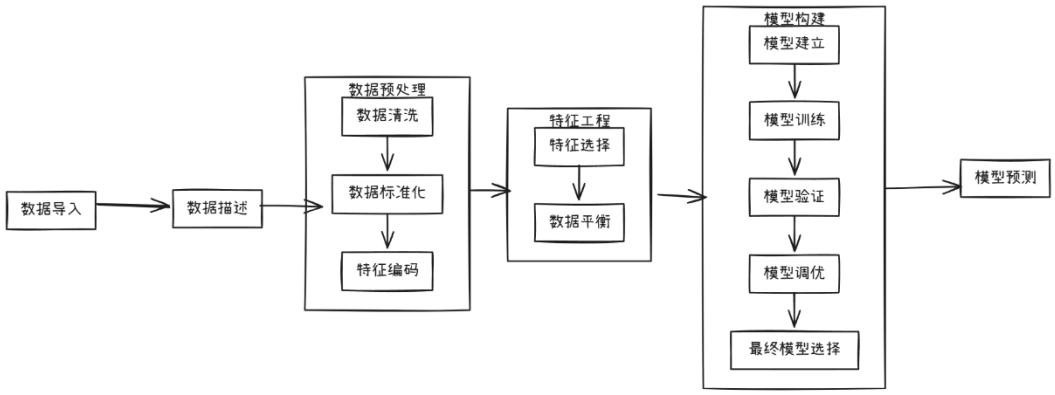
一、项目背景

随着市场饱和度的上升，电信运营商的竞争也越来越激烈，再加上高昂的客户获取成本，流失分析就变得非常关键。流失率是一种指标，用于描述取消或未续订公司套餐的客户数量。对于客户流失率而言，每增加 5%，利润就可能随之降低 25%-85%。因此，如何减少电信客户流失的分析与预测至关重要。基于从客户流失分析中获得的信息，电信公司可以制定战略、瞄准细分市场，提高所提供服务的质量以改善客户体验，从而培养客户的信任度。

二、运行环境

操作系统：Windows11
开发环境：python3.10,JupyterNotebook

三、技术路线



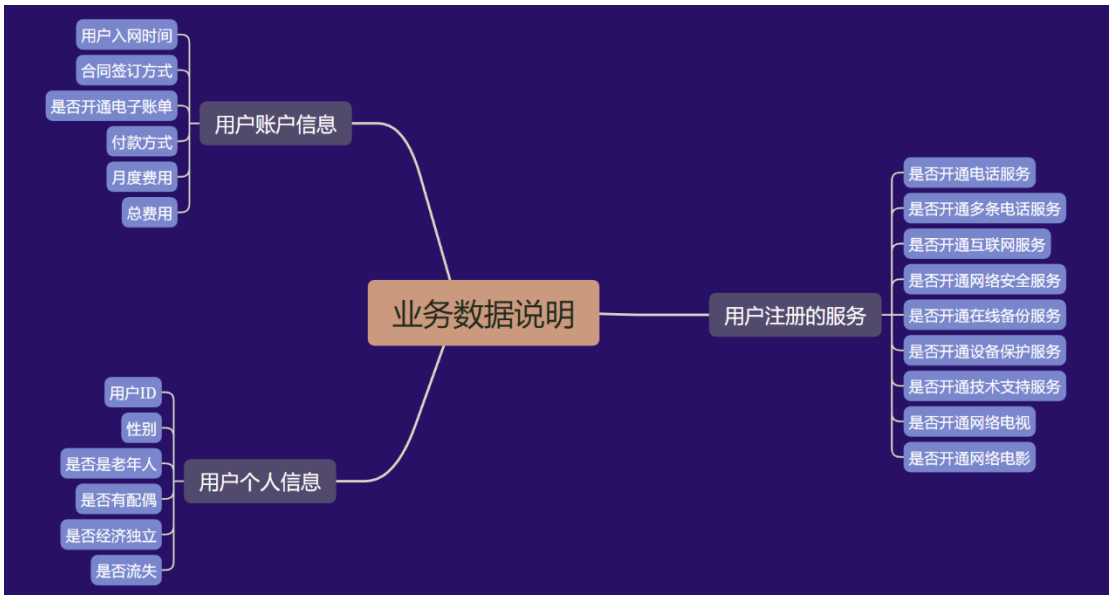
数据预测流程图

四、数据准备

4.1 源数据描述

一、 数据集描述

本数据集为电信客户流失情况，包括三个部分用户账户信息、用户个人信息、用户注册的服务



用户账户信息

tenure ： 任期

MonthlyCharges：月费用

TotalCharges：总费用

Contract：签订合同方式 （按月，一年，两年）

PaperlessBilling：是否开通电子账单（Yes or No）

PaymentMethod：付款方式（bank transfer，credit card，electronic check，mailed check）

用户注册的服务

PhoneService ： 是否开通电话服务业务 （Yes or No）

MultipleLines：是否开通了多线业务（Yes 、No or No phoneservice）

InternetService：是否开通互联网服务 （No, DSL 数字网络，fiber optic 光纤网络）

OnlineSecurity：是否开通网络安全服务（Yes，No，No internetserive）

OnlineBackup：是否开通在线备份业务（Yes，No，No internetserive）

DeviceProtection：是否开通了设备保护业务（Yes，No，No internetserive）

TechSupport：是否开通了技术支持服务（Yes，No，No internetserive）

StreamingTV：是否开通网络电视（Yes，No，No internetserive）

StreamingMovies：是否开通网络电影（Yes，No，No internetserive）

用户个人信息

Churn：该用户是否流失（Yes or No）

customerID ： 用户 ID。

gender：性别。（Female & Male）

SeniorCitizen ： 老年人 （1 表示是，0 表示不是）

Partner ： 是否有配偶 （Yes or No）

Dependents ： 是否经济独立 （Yes or No）

```
import pandas as pd
import seaborn as sns


import matplotlib.pyplot as plt
from sklearn.utils import resample
plt.rcParams['font.sans-serif'] = 'SimHei'
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

df = pd.read_csv("C:\\Users\\Aitong\\Desktop\\临时文件\\数据挖掘\\Telco Customer Churn 电信客户流失\\WA_Fn-UseC_-Telco-Customer-Churn.csv")
df.info()
```

使用 info 函数可以看到本数据集包括 7043 条数据，21 个变量。数据类型包括 object，int64，float64

发现 MonthlyCharges：月费用的数据类型错误应为浮点数类型

```
<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)
memory usage: 1.1+ MB
```

二、 数据描述

1. 查看分类属性变量的取值情况

划分数值属性和分类属性

```
categoryVar = [i for i in df.select_dtypes(object).columns ]
numberVar =[i for i in df.select_dtypes(include =[int,float]).columns]
for i in categoryVar:
    print(i,":",df[i].unique())
```

```
gender : ['Female' 'Male']
Partner : ['Yes' 'No']
Dependents : ['No' 'Yes']
PhoneService : ['No' 'Yes']
MultipleLines : ['No phone service' 'No' 'Yes']
InternetService : ['DSL' 'Fiber optic' 'No']
OnlineSecurity : ['No' 'Yes' 'No internet service']
OnlineBackup : ['Yes' 'No' 'No internet service']
DeviceProtection : ['No' 'Yes' 'No internet service']
TechSupport : ['No' 'Yes' 'No internet service']
StreamingTV : ['No' 'Yes' 'No internet service']
StreamingMovies : ['No' 'Yes' 'No internet service']
Contract : ['Month-to-month' 'One year' 'Two year']
PaperlessBilling : ['Yes' 'No']
PaymentMethod : ['Electronic check' 'Mailed check' 'Bank transfer (automatic)'
 'Credit card (automatic)']
TotalCharges : ['29.85' '1889.5' '108.15' ... '346.45' '306.6' '6844.5']
Churn : ['No' 'Yes']
```

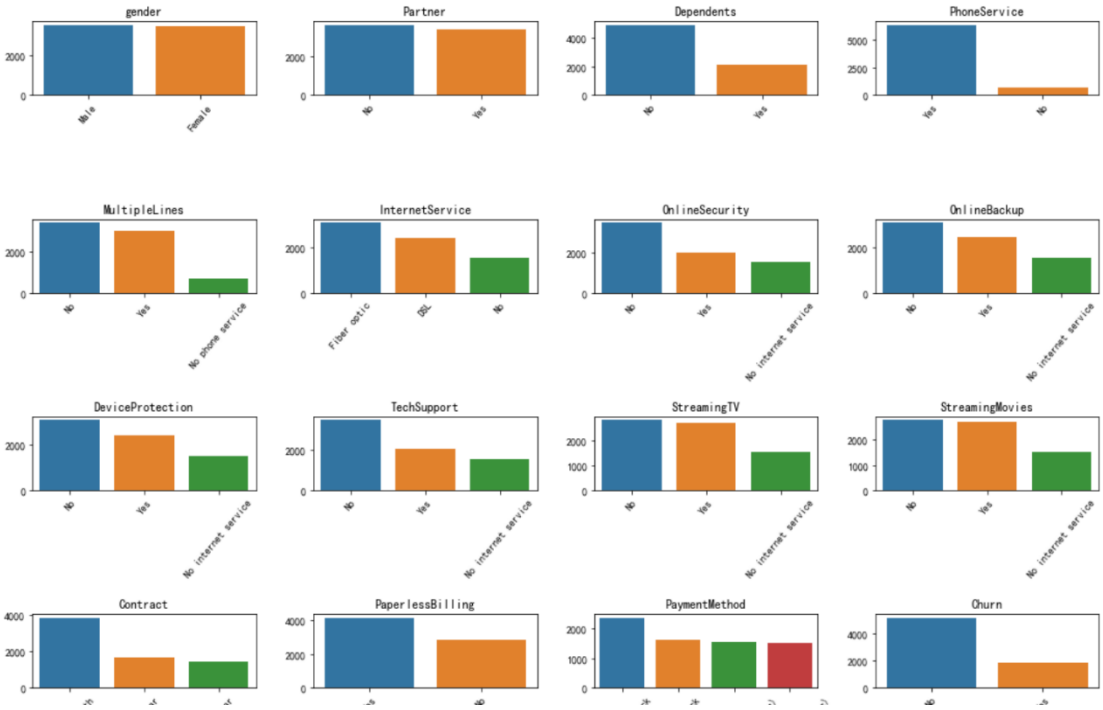
```
for col in categoryVar:
    col_count = df[col].value_counts()
    print(col_count)
```

查看列值分布情况

```
Male      3555
Female    3488
Name: gender, dtype: int64
No     3641
Yes    3402
Name: Partner, dtype: int64
No     4933
Yes    2110
Name: Dependents, dtype: int64
Yes    6361
No      682
Name: PhoneService, dtype: int64
No                  3390
Yes                 2971
No phone service     682
Name: MultipleLines, dtype: int64
Fiber optic    3096
DSL            2421
No             1526
Name: InternetService, dtype: int64
No                     3498
Yes                    2019
No internet service    1526
Name: OnlineSecurity, dtype: int64
```

使用条形图可视化

```
cat_col = [i for i in df.select_dtypes(object).columns if i not in ['customerID','MonthlyCharges','TotalCharges']]
plt.figure(figsize=(15, 20))
i = 1
for j in cat_col:
    plt.subplot(len(cat_col) // 2 + len(cat_col) % 2, 4, i)
    sns.barplot(x=df[j].value_counts().index, y=df[j].value_counts().values)
    plt.xticks(rotation=50)
    plt.title(j)
    i += 1
plt.tight_layout()
plt.show()
```



发现有许多的二分属性还有 No internetserive 的列值

2. 连续属性的描述信息

```
df.describe()
```

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
count	7043.000000	7043.000000	7043.000000	7043.000000
mean	0.162147	32.372710	64.761692	2279.798992
std	0.368612	24.557454	30.090047	2266.730170
min	0.000000	1.000000	18.250000	18.800000
25%	0.000000	9.000000	35.500000	398.550000
50%	0.000000	29.000000	70.350000	1394.550000
75%	0.000000	55.000000	89.850000	3786.600000
max	1.000000	72.000000	118.750000	8684.800000

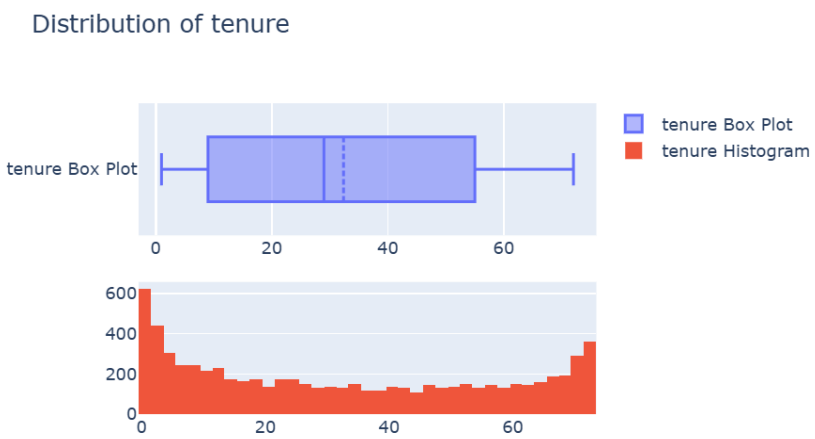
、

可以看出 MonthlyCharges"、"TotalCharges"两个特征跟其他特征相比，量纲差异较大

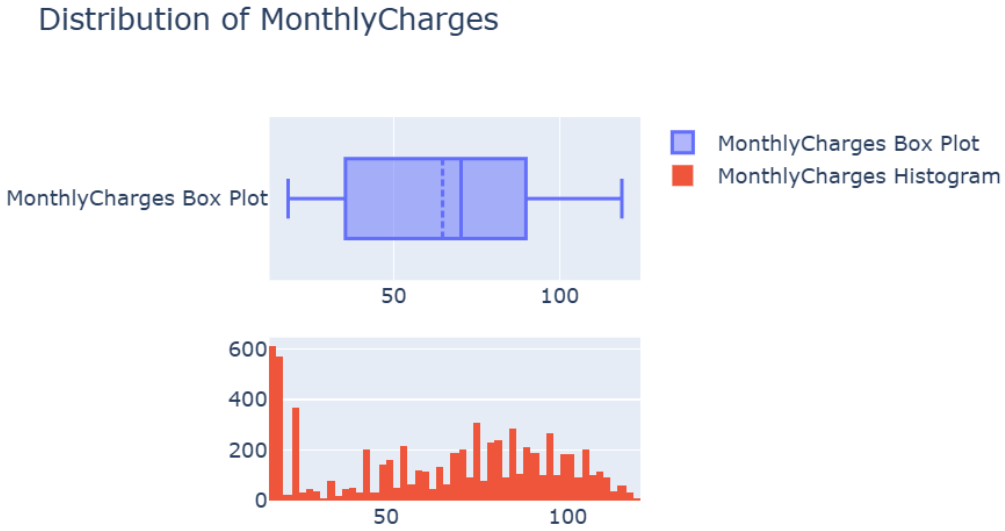
3. 连续属性数据分布情况

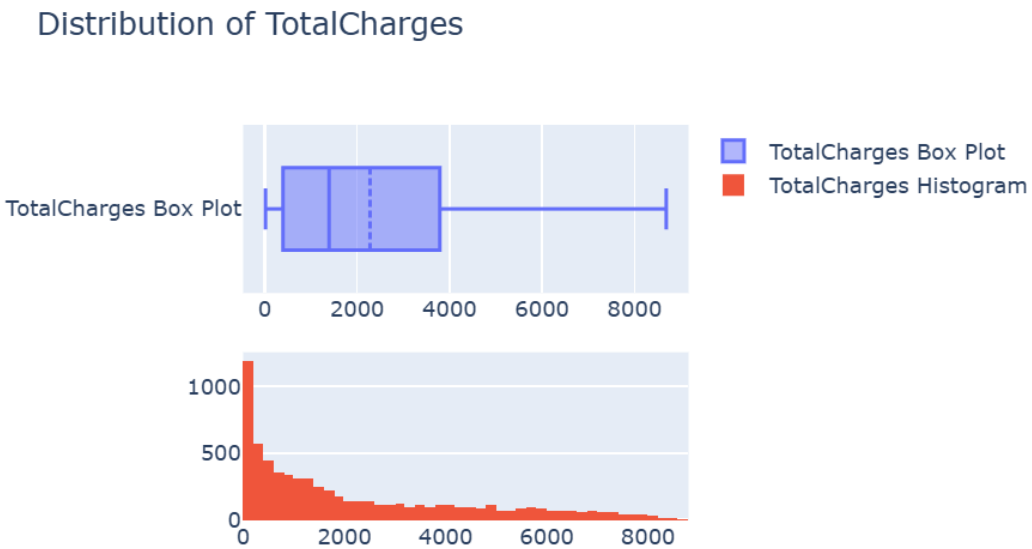
使用箱形图和直方图展现连续属性数据分布，最大值，最小值与中值

```
import plotly.graph_objs as go
from plotly.subplots import make_subplots
for i in numeric_features:
    fig = make_subplots(rows=2, cols=1)
    tr1=go.Box(x=df[i],name=f'{i} Box Plot',boxmean=True)
    tr2=go.Histogram(x=df[i],name=f'{i} Histogram')
    fig.add_trace(tr1,row=1,col=1)
    fig.add_trace(tr2,row=2,col=1)
    fig.update_layout(height=400, width=600,
                      title_text=f"Distribution of {i}")
    fig.show()
```



用户入网时间呈现两端高的情况，可能是入网初期有非常有吸引力的促销活动，这些优惠活动吸引了大量对价格敏感导致短期内入网人数增多。随着时间的推移，运营商在网络质量、客户服务等方面建立了良好的口碑，对于一些对稳定性和长期服务有需求的用户来说，会选择长期保留。

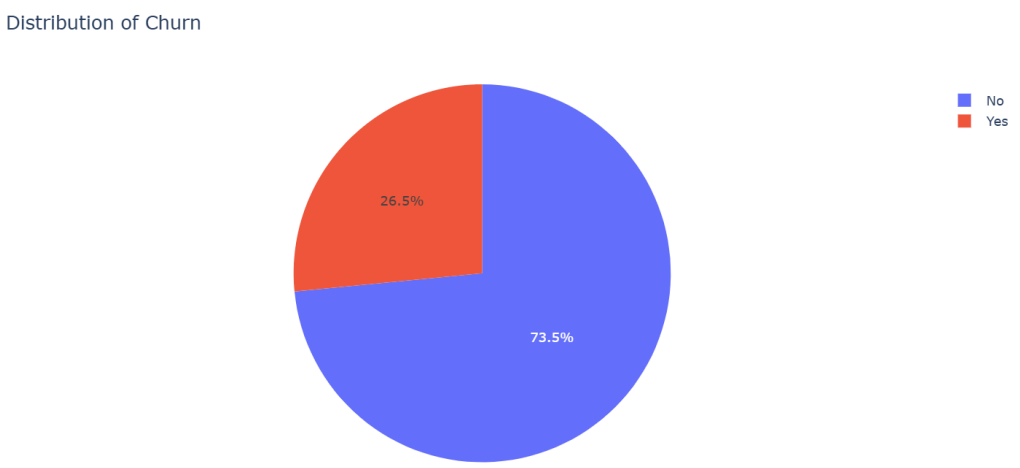




消费金额所出现的波动，有可能是套餐定价策略引起

4. 预测目标变量的分布情况

```
import plotly.express as ex
ex.pie(df,names='Churn',title='Distribution of Churn')
```



可以看出 yes 值（留存）占数据集的 73.5%，NO 值（流失）占数据集的 26.5%存在明显的样本不均衡问题

4. 2 数据预处理

1.转换数值类型

将 MonthlyCharges 月费用的数据类型转换为浮点数类型

```
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
```

2.缺失值处理

```
print("缺失值统计：")
print(df.isnull().sum())
```

```
缺失值统计：
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges       11
Churn               0
dtype: int64
```

发现 TotalCharges 总消费额有 11 个用户数据缺失值为 NaN

对连续属性缺失值的处理方法有多种，列入如使用中位数，众数平均数，

回归填充或删除等操作
但是通过对缺失值列和其他列的观察可以发现缺失值的规律

```
missing_rows = df[df.isnull().any(axis=1)]
nan_rows = df[pd.isna(df['TotalCharges'])]
print(nan_rows[['tenure', 'MonthlyCharges', 'TotalCharges']])
```

打印出包含缺失值的列，可以看出它们的 tenure 使用月数都为 0，推测他们是本月新注册的用户。可能注册时长还没有过一个月周期，因此 TotalCharge 值为 NaN

	tenure	MonthlyCharges	TotalCharges
488	0	52.55	NaN
753	0	20.25	NaN
936	0	80.85	NaN
1082	0	25.75	NaN
1340	0	56.05	NaN
3331	0	19.85	NaN
3826	0	25.35	NaN
4380	0	20.00	NaN
5218	0	19.70	NaN
6670	0	73.35	NaN
6754	0	61.90	NaN

再打印出 tenure 使用月数为 1 的用户数据
可以发现由于经过一个月周期 TotalCharge 变为 MonthlyChargers 月费用的值并且 tenure 变为 1。可以证实我们的猜测

	tenure	MonthlyCharges	TotalCharges
0	1	29.85	29.85
20	1	39.65	39.65
22	1	20.15	20.15
27	1	30.20	30.20
33	1	20.20	20.20
...	...	...	...
6979	1	24.20	24.20
7010	1	74.45	74.45
7016	1	49.95	49.95
7018	1	70.65	70.65
7032	1	75.75	75.75

因此我们使用 MonthlyChargers 的值来填充 TotalCharge 的缺失值并把 tenure 变为 1

```
#将总消费额填充为月消费额
nan_mask = pd.isna(df['TotalCharges'])
df.loc[nan_mask, 'TotalCharges'] = df.loc[nan_mask, 'MonthlyCharges']
df.loc[nan_mask, 'tenure']=1
print(df[nan_mask][['tenure', 'MonthlyCharges', 'TotalCharges']])
```

3.重复值检查

```
df.duplicated().sum()
```

```
0
```

4.数据编码

(1). 处理量纲差异

“MonthlyCharges"、"TotalCharges"量纲差异大通过分析得分我们这里选择使用离散化的方式

用四分位数进行离散

df['MonthlyCharges']=pd.qcut(df['MonthlyCharges'],4,labels=['1','2','3','4'])

df['TotalCharges']=pd.qcut(df['TotalCharges'],4,labels=['1','2','3','4'])

```
]: <bound method NDFrame.head of 0       1
   1       2
   2       2
   3       2
   4       3
          ..
   7038    3
   7039    4
   7040    1
   7041    3
   7042    4
   Name: MonthlyCharges, Length: 7043, dtype: category
   Categories (4, object): ['1' < '2' < '3' < '4']>
```

（2）分类属性编码

由于“No internetserive”的人数占比 OnlineSecurity， OnlineBackup，DeviceProtection，TechSupport，StreamingTV，StreamingTV 占比一致可以判断 No internetserive”不影响预测结果将“No internetserive”并到“No”里面。

df.replace(to_replace='No internet service',value='No',inplace=True)

df.replace(to_replace='No phone service',value='No',inplace=True)

对分类属性进行序列编码

df[x] = LabelEncoder().fit_transform(df[x])

（3）处理样本不均

上采样

```
X_minority = X_train[y_train == 1]
X_majority = X_train[y_train == 0]
y_minority = y_train[y_train == 1]
y_majority = y_train[y_train == 0]
X_minority_upsampled = resample(X_minority, replace=True, n_samples=len(X_majority),
y_minority_upsampled = resample(y_minority, replace=True, n_samples=len(y_majority),
X_balanced = pd.concat([X_majority, X_minority_upsampled])
y_balanced = pd.concat([y_majority, y_minority_upsampled])
```

下采样

```
X_minority = X[y == 1]
X_majority = X[y == 0]
y_minority = y[y == 1]
y_majority = y[y == 0]

X_majority_downsampled = resample(X_majority, replace=False, n_samples=len(X_minority),
y_majority_downsampled = resample(y_majority, replace=False, n_samples=len(y_minority),

X_balanced = pd.concat([X_minority, X_majority_downsampled])
y_balanced = pd.concat([y_minority, y_majority_downsampled])
```

表 X　　数据预处理方案

阶段	预处理目的	预处理方案
1	转换数值类型	使用 to_numeric
2	缺失值处理	寻找规律填充
3	重复值检查	Duplicated 函数
4	处理量纲差异	四分位数离散
5	分类属性编码	LabelEncoder 编码
6	处理样本不均	上采样与下采样

4.3 数据预处理结果展示

缺失值处理

	tenure	MonthlyCharges	TotalCharges
488	1	52.55	52.55
753	1	20.25	20.25
936	1	80.85	80.85
1082	1	25.75	25.75
1340	1	56.05	56.05
3331	1	19.85	19.85
3826	1	25.35	25.35
4380	1	20.00	20.00
5218	1	19.70	19.70
6670	1	73.35	73.35
6754	1	61.90	61.90

11 个缺失值列填充情况

```
print(df.isnull().sum())
```

```
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
dtype: int64
```

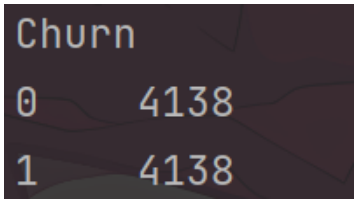
查找缺失值

编码结果：

```
df.head()
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSuppo
0	0	0	1	0	1	0	0	0	0	1	0	
1	1	0	0	0	34	1	0	0	1	0	1	
2	1	0	0	0	2	1	0	0	1	1	0	
3	1	0	0	0	45	0	0	0	1	0	1	
4	0	0	0	0	2	1	0	1	0	0	0	

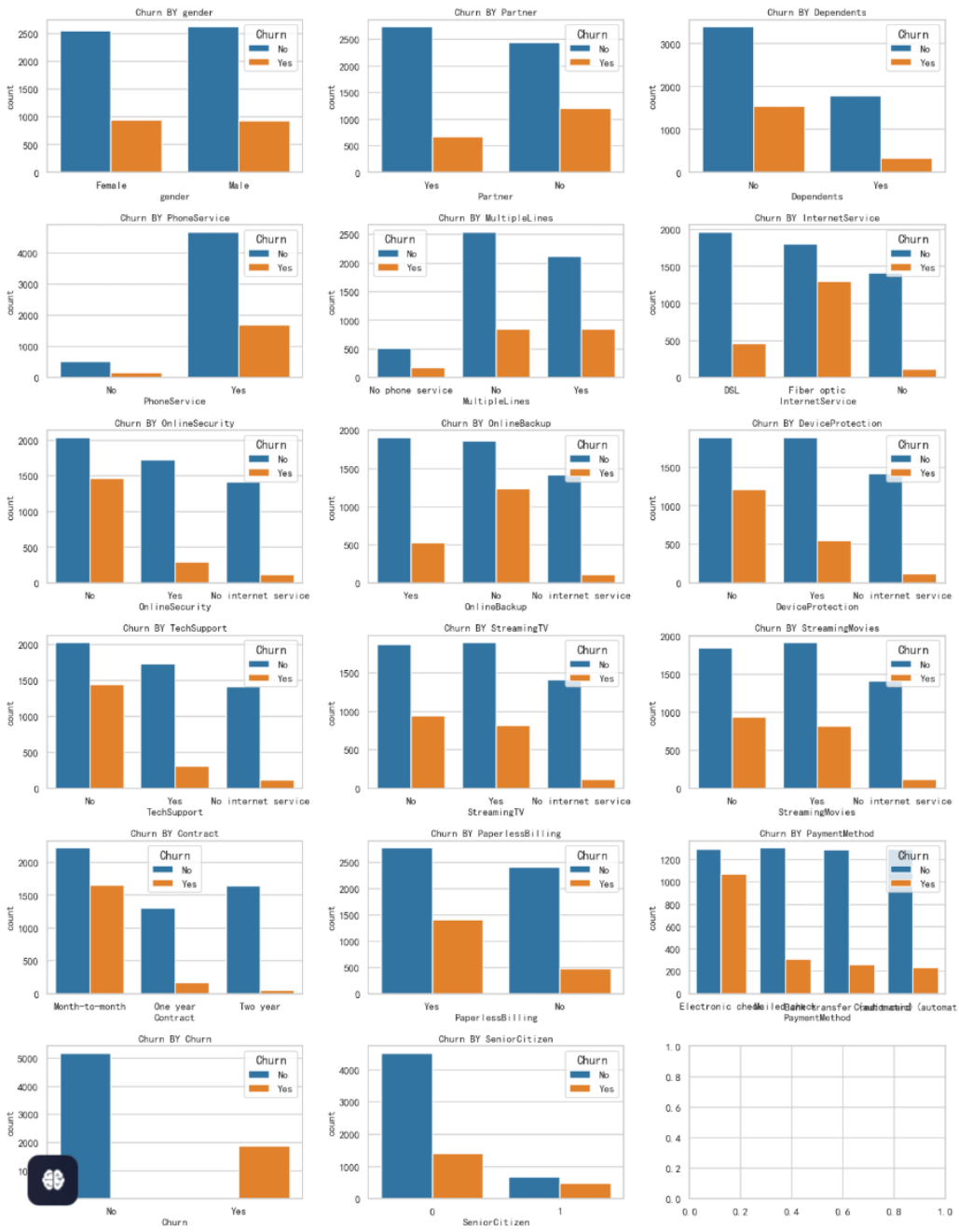
数据平衡结果



4.4 探索性数据分析

（1）查看分类属性每个标签的对应 churn 流失的分布情况

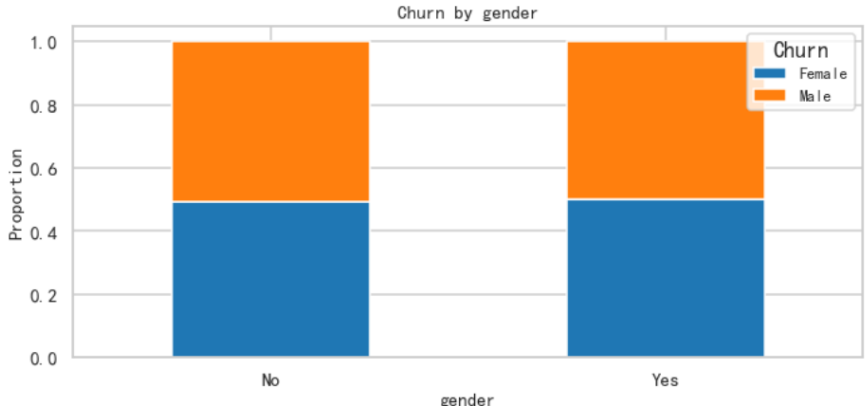
```
figsize = (20, 25)
fig, axes = plt.subplots(6, 3, figsize=figsize)
axes = axes.flatten()

for ax, k in zip(axes, categorical_features):
    sns.countplot(x=k, hue='Churn', data=df, ax=ax)
    ax.legend(title='Churn',fontsize='smaller')
    ax.set_title('Churn BY ' + k)
plt.tight_layout()

# 显示图形
plt.show()
```



由于样本 churn 的用户流失比例极不平衡，很难看出维度对流失用户的作用情况因此我们使用交叉分析的方法使用 pd.crosstab()函数来创建与 churn 的交叉表，

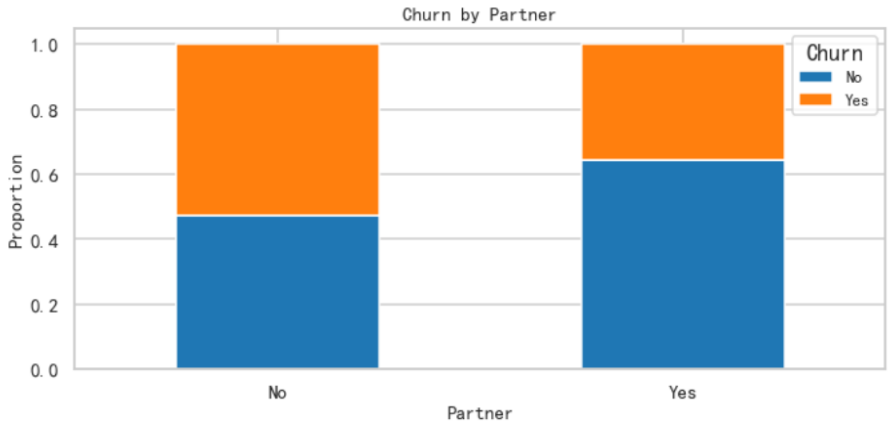
```
for i in categorical_features:
    crosstab = pd.crosstab(df['Churn'], df[i], normalize=0)
    print(crosstab, '\n')
```

gender	Female	Male
Churn		
No	0.492656	0.507344
Yes	0.502408	0.497592



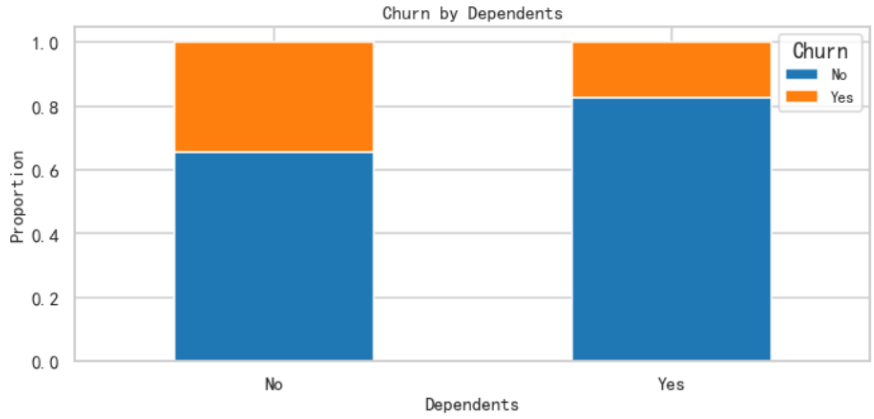
Gender：性别对用户流失留存影响不大

Partner	No	Yes
Churn		
No	0.471782	0.528218
Yes	0.642055	0.357945



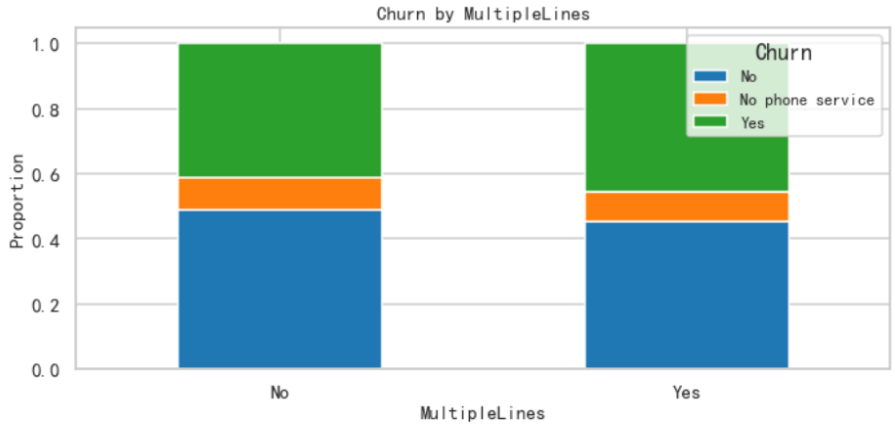
Partner：单身用户更容易流失

Dependents	No	Yes
Churn		
No	0.655199	0.344801
Yes	0.825575	0.174425



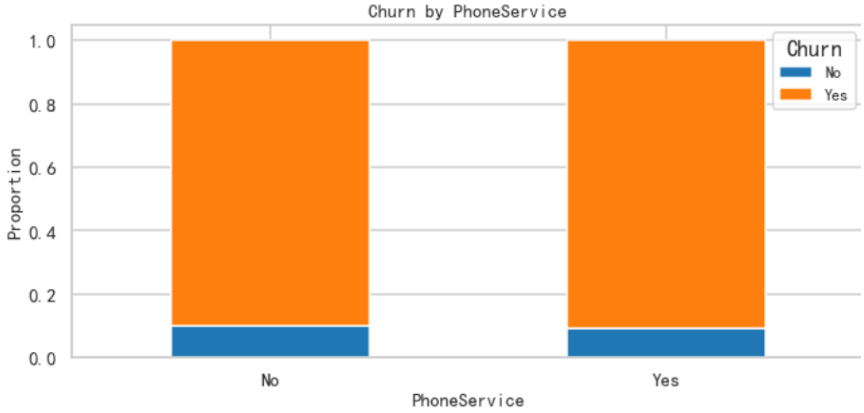
Dependents：经济不独立的用户更容易流失

MultipleLines	No	No phone service	Yes
Churn			
No	0.491109	0.098956	0.409934
Yes	0.454254	0.090958	0.454789



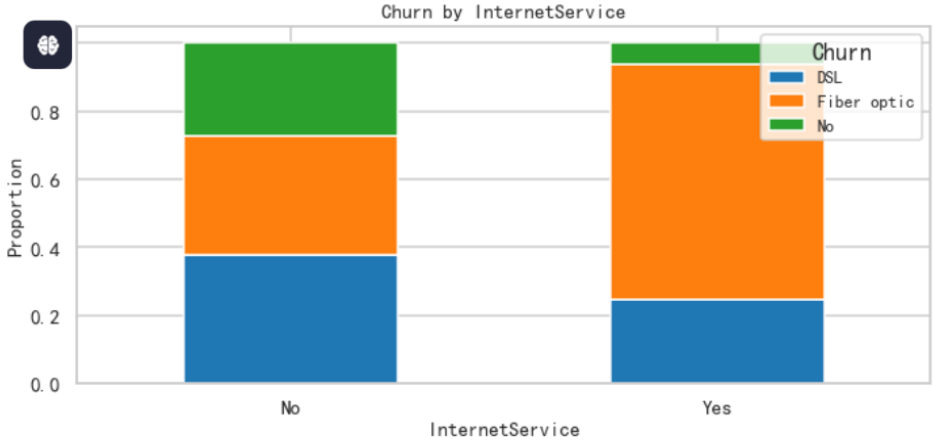
MultipleLines：是否开通 MultipleLines 对用户流失留存影响不大

PhoneService	No	Yes
Churn		
No	0.098956	0.901044
Yes	0.090958	0.909042



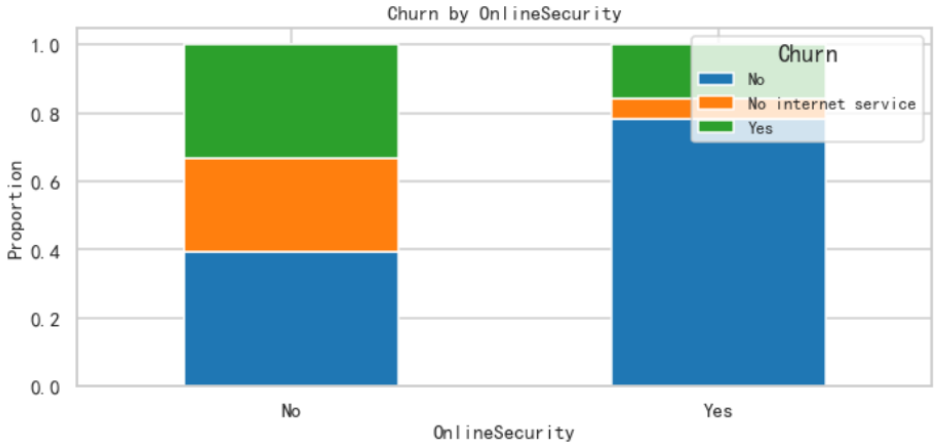
Phoneservice：是否开通 Phoneservice 对用户流失留存影响不大

InternetService	DSL	Fiber optic	No
Churn			
No	0.379204	0.347700	0.273096
Yes	0.245586	0.693954	0.060460



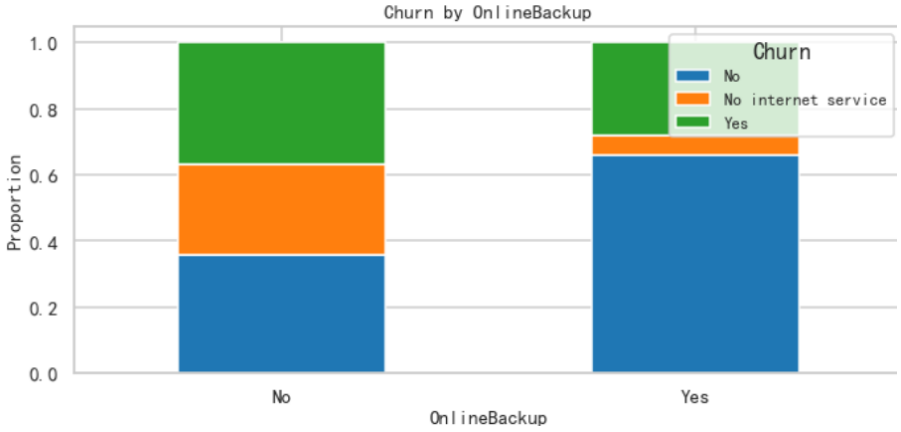
Internerservice：办理了 Fiber opti 的客户容易流失

OnlineSecurity	No	No internet service	Yes
Churn			
No	0.393699	0.273096	0.333204
Yes	0.781701	0.060460	0.157838



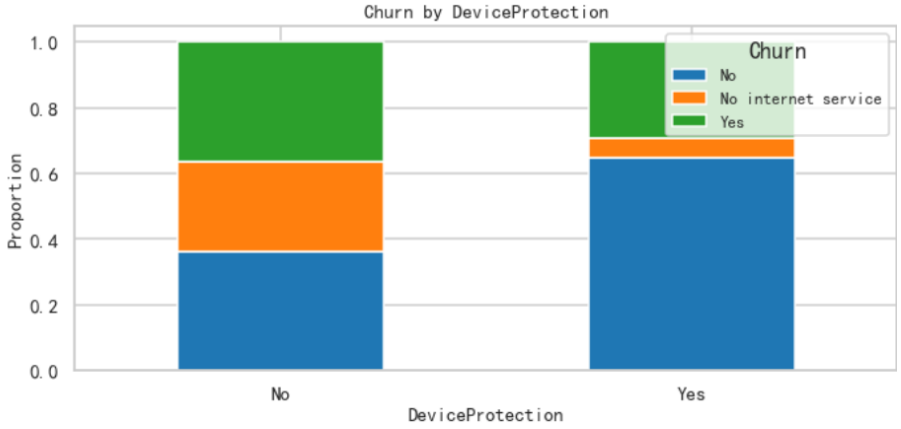
OnlineSecurity：没开通的客户容易流失。

OnlineBackup	No	No internet service	Yes
Churn			
No	0.358523	0.273096	0.368380
Yes	0.659711	0.060460	0.279829



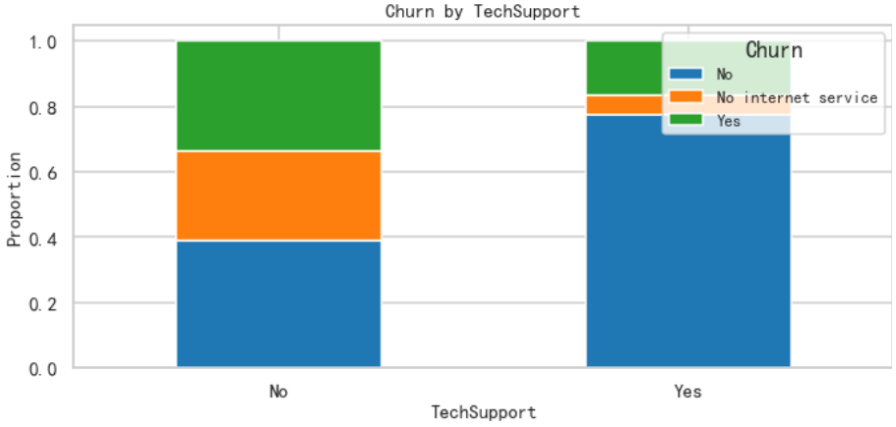
OnlineBackup：没开通在线备份服务的客户容易流失。

DeviceProtection	No	No internet service	Yes
Churn			
No	0.364128	0.273096	0.362775
Yes	0.647940	0.060460	0.291600



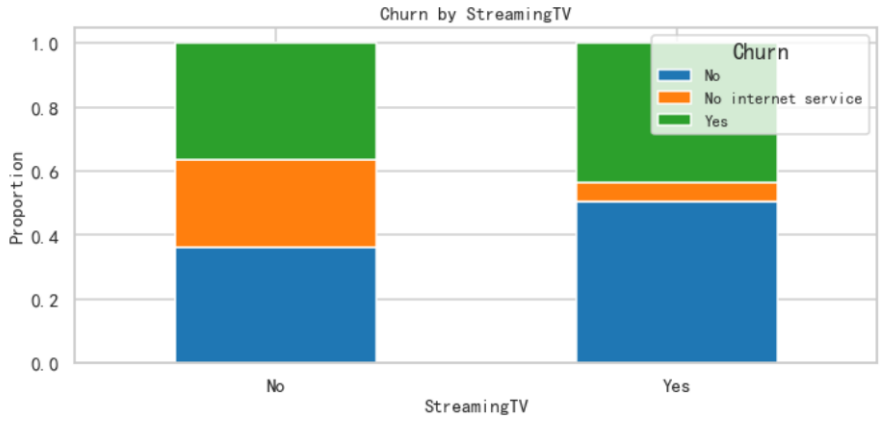
DeviceProtection：没开通设备保护业务的用户比较容易流失

TechSupport	No	No internet service	Yes
Churn			
No	0.391767	0.273096	0.335137
Yes	0.773676	0.060460	0.165864



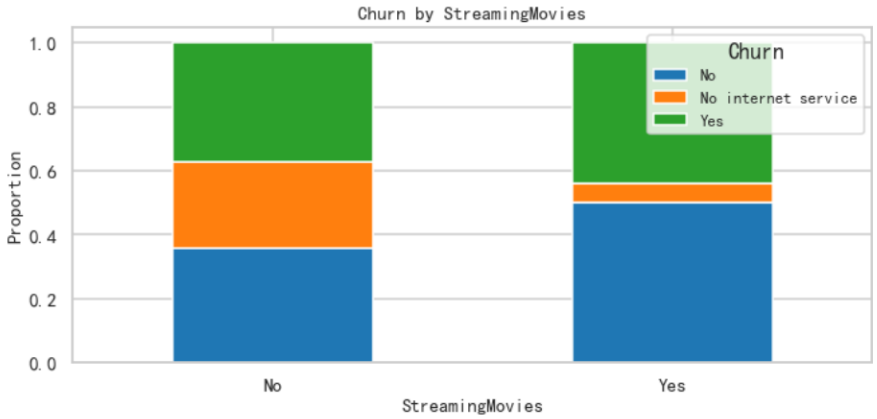
TechSupport：没开通技术支持服务的用户容易流失。

StreamingTV	No	No internet service	Yes
Churn			
No	0.361036	0.273096	0.365868
Yes	0.504013	0.060460	0.435527



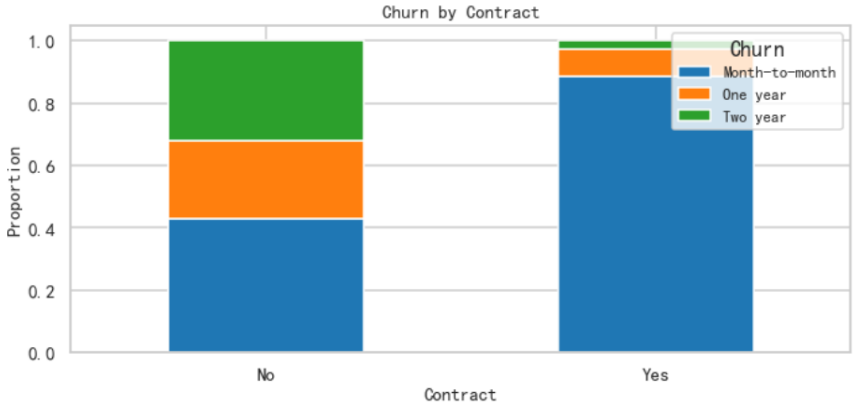
StreamingTV：没有开通网络电视服务的用户更容易流失，但较不显著

StreamingMovies	No	No internet service	Yes
Churn			
No	0.356977	0.273096	0.369927
Yes	0.501873	0.060460	0.437667



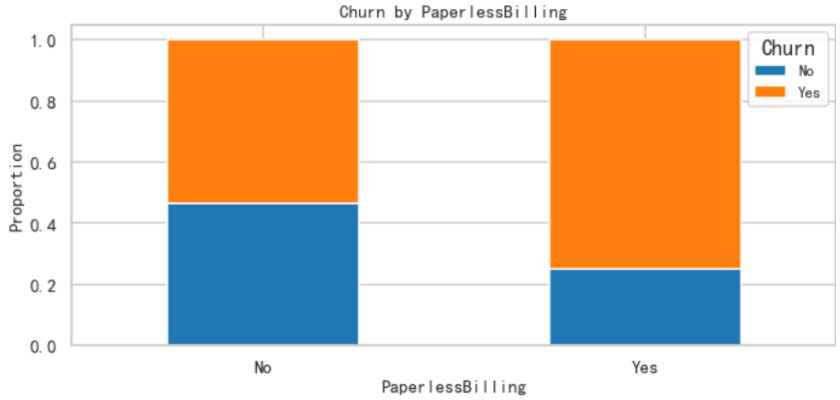
StreamingMovies：没有开通网络电视服务的用户更容易流失，但较不显著

Contract	Month-to-month	One year	Two year
Churn			
No	0.429068	0.252609	0.318322
Yes	0.885500	0.088818	0.025682



Contract：月份签订合同的用户最容易流失。签约越久越不容易流失

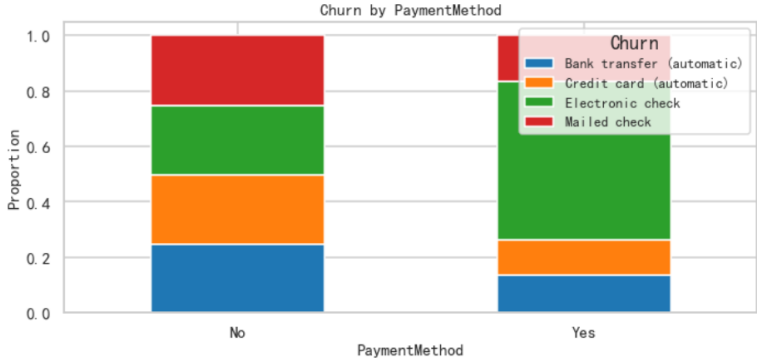
PaperlessBilling	No	Yes
Churn		
No	0.464438	0.535562
Yes	0.250936	0.749064



PaperlessBilling ：开通电子账单的用户较容易流失

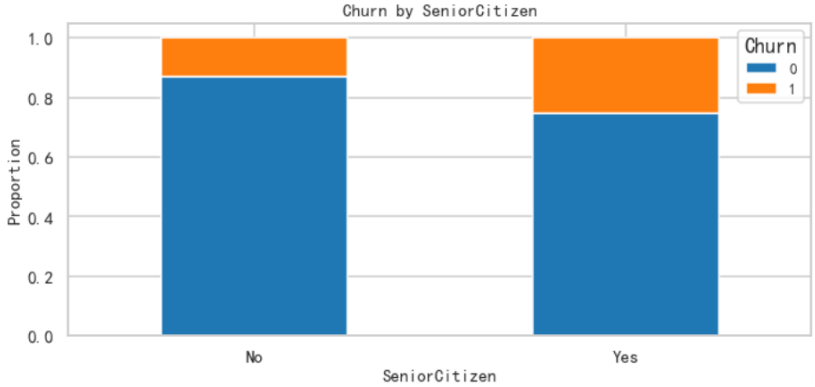
PaymentMethod	Bank transfer (automatic)	Credit card (automatic)	\
Churn			
No	0.248550	0.249324	
Yes	0.138042	0.124131	

PaymentMethod	Electronic check	Mailed check
Churn		
No	0.250097	0.252029
Yes	0.573034	0.164794



PaymentMethod：使用电子支票支付的人更容易流失

SeniorCitizen	0	1
Churn		
No	0.871279	0.128721
Yes	0.745318	0.254682



SeniorCitizen 分析：年轻用户在流失、留存人数占比都高

五、建模分析

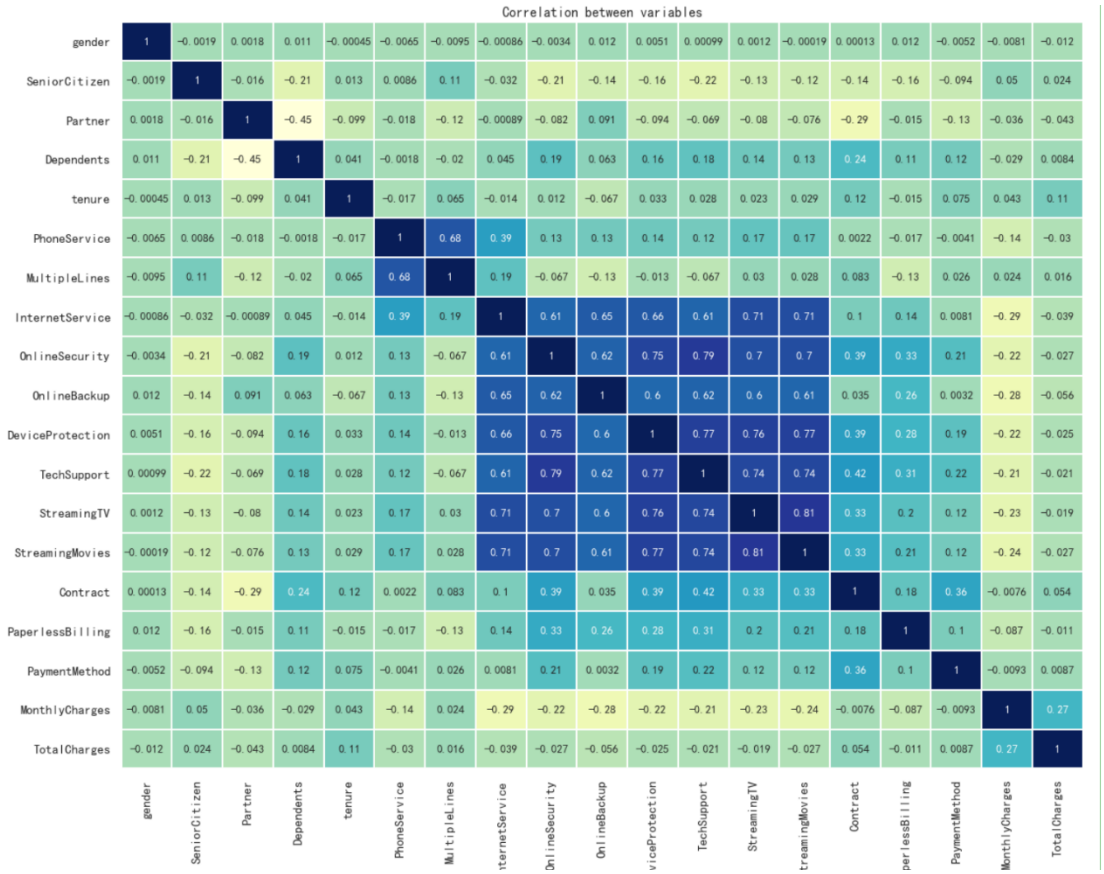
5.1 数据划分

对数据集进行二八拆分为测试集与训练集
X_train, X_val,y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

5.2 绘制热力图观察变量之间的相关性

打印相关性矩阵绘制热力图

```
corr_df = feature.apply(lambda x: pd.factorize(x)[0])
corr_df.head()
corr=corr_df.corr()
corr
```

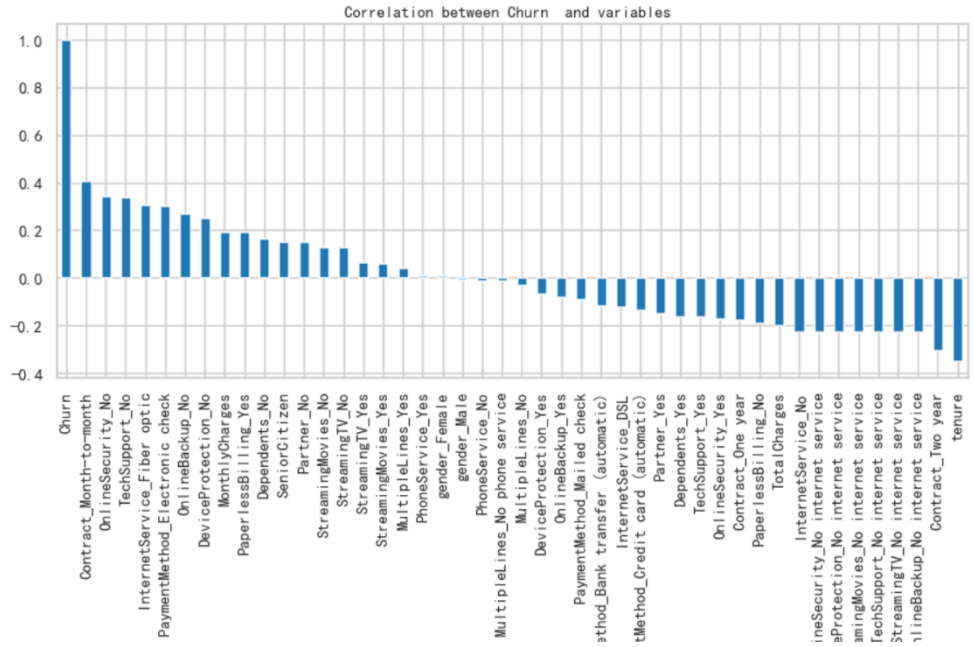


从热力图来看，互联网服务、网络安全、在线备份、设备维护服务、技术支持服务、开通网络电视服务、开通网络电影之间相关性很强，且是正相关。说明在电信业务中这些维度具有一定联系

5.3 绘制 churn 与变量关系图

```
plt.figure(figsize=(15,6))
df_onehot.corr()['Churn'].sort_values(ascending=False).plot(kind='bar')
plt.title('Correlation between Churn  and variables ')
```

```
Out[17]: Text(0.5, 1.0, 'Correlation between Churn  and variables ')
```



从图看 gender（性别）、PhoneService（电话服务）相关性几乎为 0，两个维度在特征选择时可以忽略。

5.4 特征选择

使用随机森林选择进行特征选择，并获取特征选择后的获取选择的特征索引，得到新的训练集

```
selector =
SelectFromModel(RandomForestClassifier(n_estimators=100,
random_state=42), threshold="median")
selector.fit(X_train_preprocessed, y_balanced)
```

特征选择结果

['tenure', 'MonthlyCharges', 'TotalCharges', 'gender', 'Partner', 'InternetService', 'OnlineSecurity', 'Contract', 'PaperlessBilling', 'PaymentMethod']

5.5 模型选择与参数调优

选择定义以下分类模型进行参数调优

```
xgb_model = XGBClassifier(eval_metric='logloss', scale_pos_weight=len(y[y
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
logreg_model = LogisticRegression(random_state=42, max_iter=1000)
svc_model = SVC(probability=True, random_state=42)
gb_model = GradientBoostingClassifier(random_state=42)
ada_model = AdaBoostClassifier(n_estimators=100, random_state=42)
```

然后对每模型设计参数网格进行超优调参

```
xgb_param_grid = {
    'n_estimators': [50, 100, 150],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 4, 5],
    'subsample': [0.6, 0.8, 1.0],
    'alpha':[0.05,0.1,0.2],
    'reg_lambda':[1.0,5.0,10.0],
    'min_child_weight':[3,5,7],
    'colsample_bytree': [0.6, 0.8, 1.0],
}

# 随机森林参数网格
rf_param_grid1 = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
```

这里我使用 RandomizedSearchCV 随机搜索在给定的超参数范围内最优的超参数组合。

random_search=RandomizedSearchCV(estimator=rf_model,param_distributions=rf_param_grid, n_iter=1, cv=skf, scoring='roc_auc', n_jobs=1)

estimator 为使用的分类模型，param_distributions 为超参数取值范围 scoring='roc_auc' 设定评估指标为 ROC AUC

使用 StratifiedKFold 进行交叉验证

```
random_search.fit(X_train, y_train)
print("Best parameters:", random_search.best_params_)
```

对定义好的参数搜索模型进行训练并输出最优参数

5.6 评估策略

（1）使用训练好的最优模型对测试集进行预测

```
y_val_pred = best_model.predict(X_val)
y_val_proba = best_model.predict_proba(X_val)[:, 1]
```

（2）调用混淆矩阵、AUC 对模型进行评估

```
from sklearn.metrics import classification_report,
accuracy_score, precision_score, recall_score,
roc_auc_score, \
    roc_curve, auc

print(classification_report(y_val, y_val_pred))
accuracy = accuracy_score(y_val, y_val_pred)
precision = precision_score(y_val, y_val_pred)
recall = recall_score(y_val, y_val_pred)
```

（3）绘制 AUC 曲线

```
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='blue', lw=2, label='ROC曲线 (AUC = {:.2f})'.form
plt.plot([0, 1], [0, 1], color='red', linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('假阳性率 (False Positive Rate)')
plt.ylabel('真正率 (True Positive Rate)')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc='lower right')
plt.grid()
plt.show()
```

5.7 运行结果展示与分析

（1）先分后采样与先采样后分操作对比

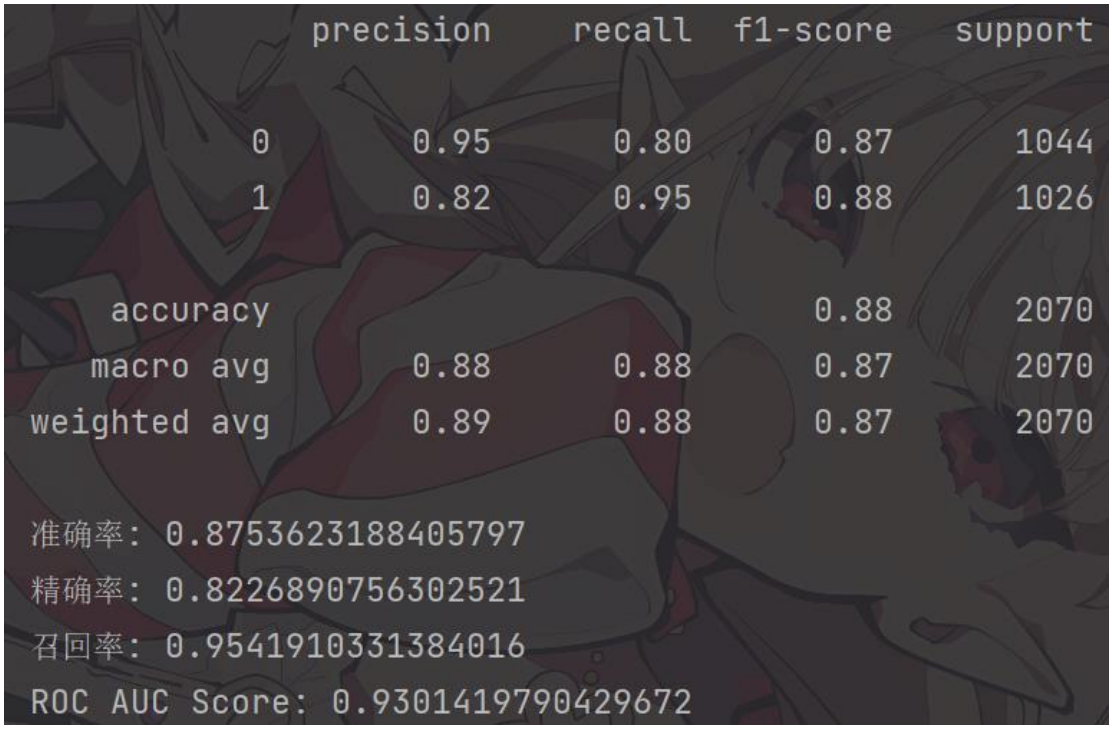


图 1 先采样后分评估结果

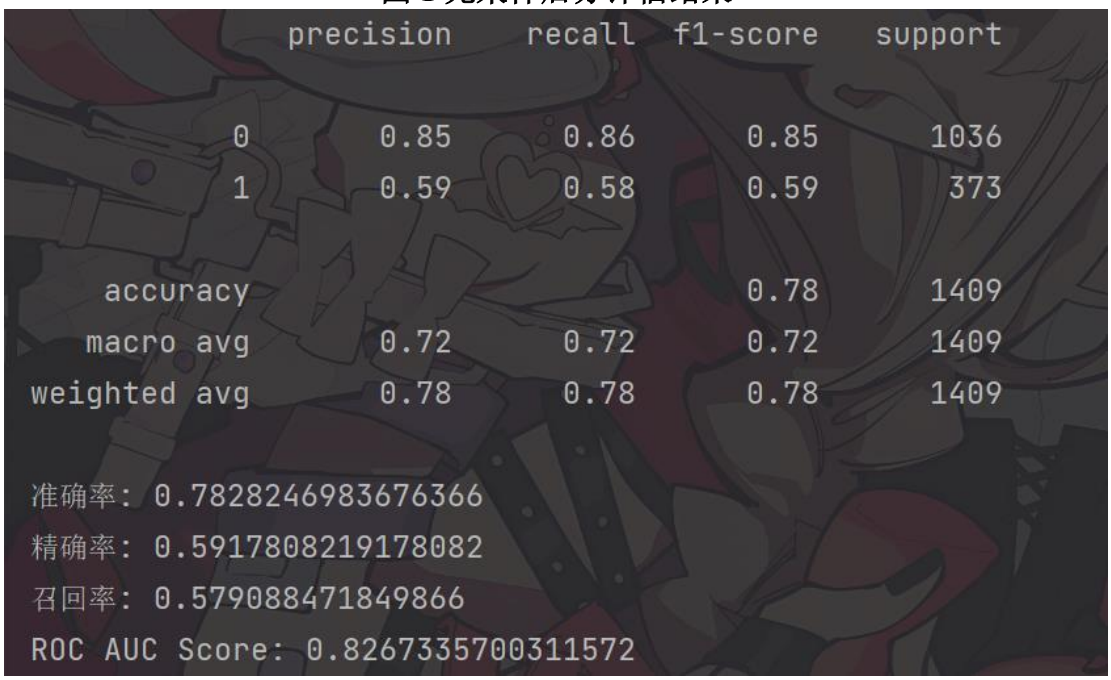


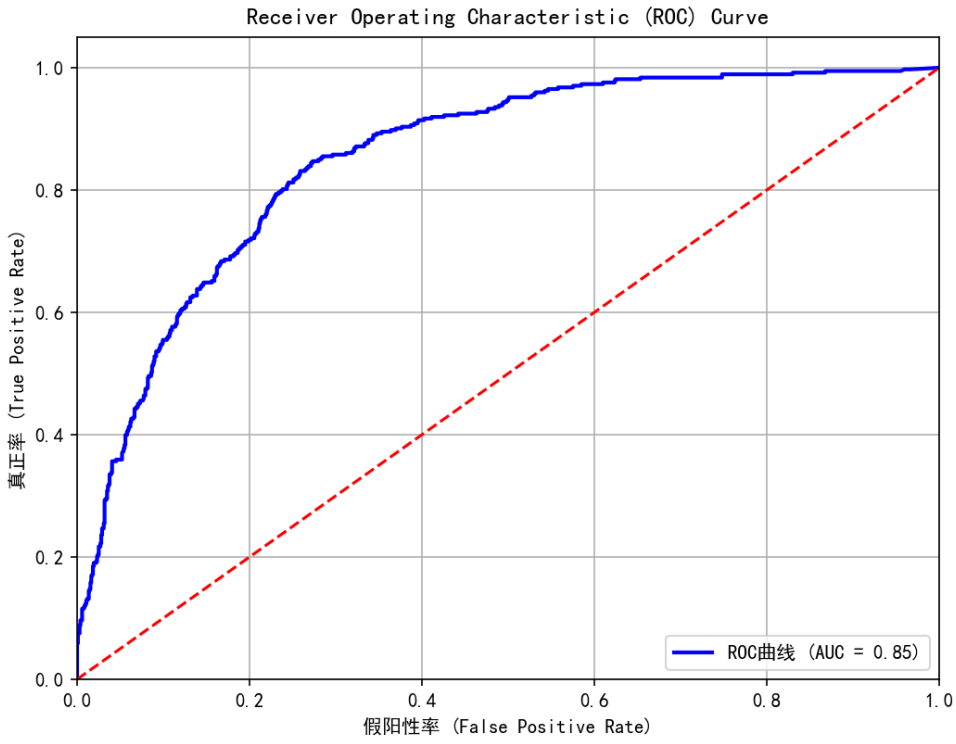
图 2 先分后采样评估结果

从图 1 与图 2 的对比中我们可以看出采样与拆分数据集的操作顺序对预测结果的影响巨大，其原因是因为先采样后拆分的数据集里包含了许多我们使用上采样方法生成的样本造成了样本的污染

（2）优化使用管道形式进行数据处理与训练（上采样）

```
preprocessor_selected = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_features_selected),
        ('cat', categorical_transformer, categorical_features_selected)
    ])
```

```
              precision    recall  f1-score   support

           0       0.89      0.79      0.84      1036
           1       0.56      0.74      0.63       373

    accuracy                           0.77      1409
   macro avg       0.72      0.76      0.74      1409
weighted avg       0.80      0.77      0.78      1409

准确率: 0.7743080198722498
精确率: 0.5553319919517102
召回率: 0.739946380697051
ROC AUC Score: 0.8501700187357024
```



AUC 曲线

可以看出在使用道进行数据处理与训练后模型评估分数上升

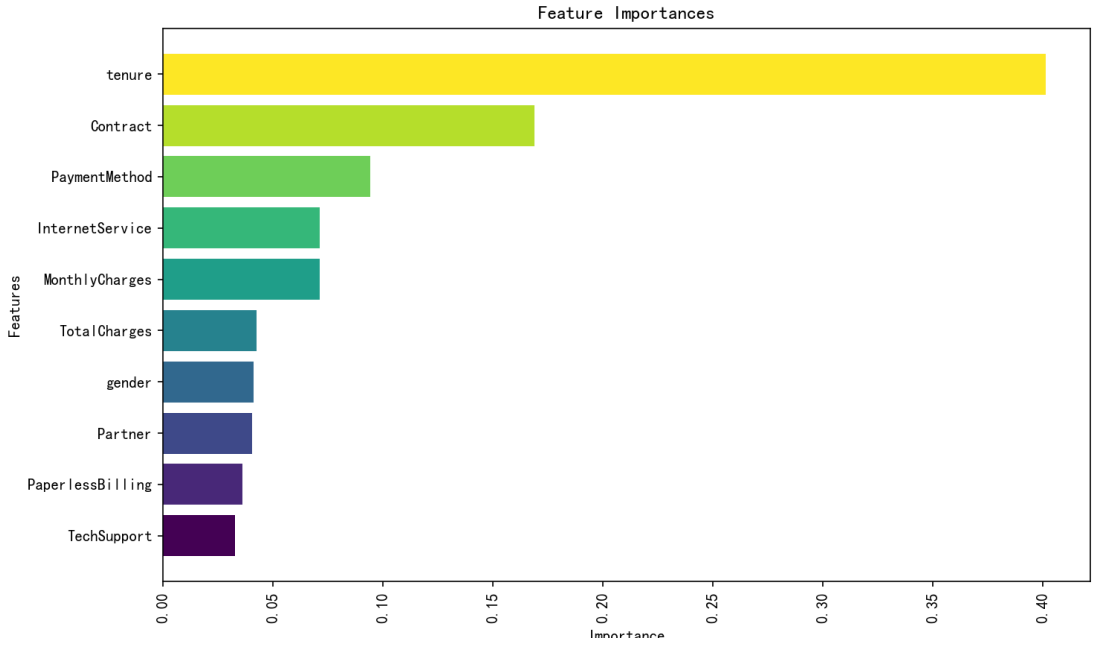
（3）数据处理与训练（下采样）

	precision	recall	f1-score	support
0	0.76	0.73	0.75	369
1	0.75	0.78	0.76	379
accuracy			0.75	748
macro avg	0.75	0.75	0.75	748
weighted avg	0.75	0.75	0.75	748

我们使用下采样对数据平衡后模型对 1 流失用户的识别率得到大幅提升，但对 0 的识别率下降

（4）使用最优模型输出特征重要性并画图

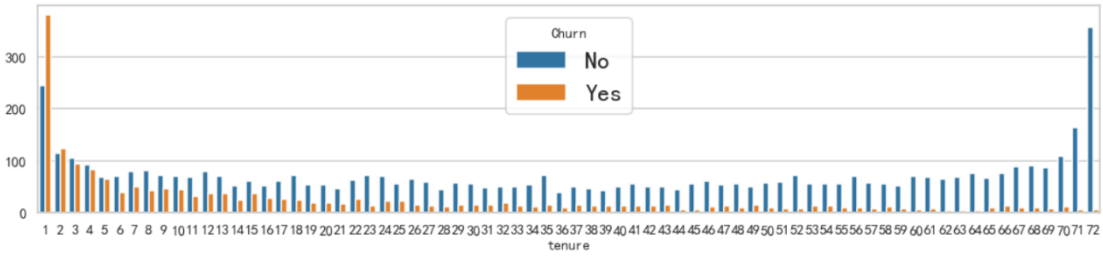
```
feature_importances = best_model.feature_importances_
sorted_indices = np.argsort(feature_importances)
sorted_importances = feature_importances[sorted_indices]
sorted_features = X_selected.columns[sorted_indices]
```



可以看出'tenure', 'MonthlyCharges', 'TotalCharges', 'Partner', 'InternetService', , 'Contract', 'PaperlessBilling', 'PaymentMethod'这些特征对客户是否流失的影响很大

（5）数值属性特征流失分布分析

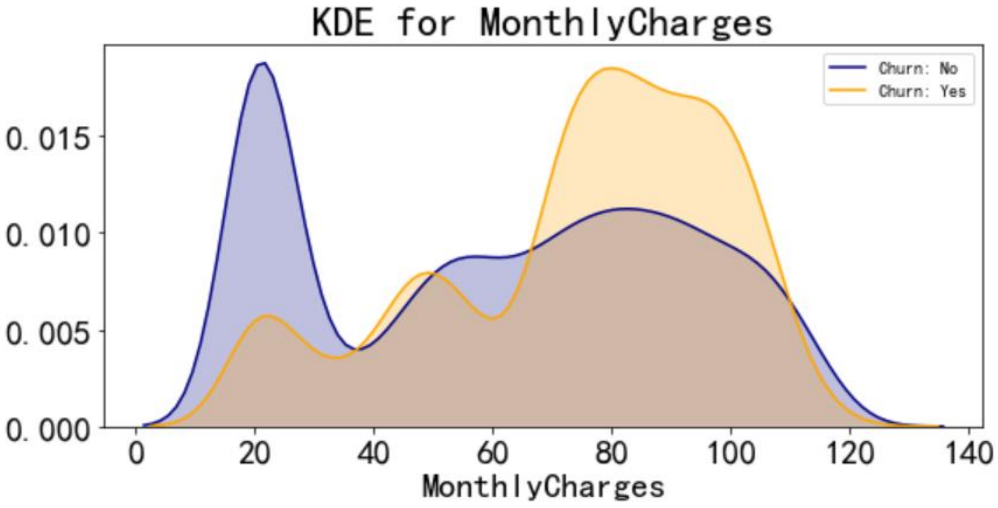
1.针对重要性最高的 tenure 特征查看其 churn 变量的分布情况



由图可以看出在任期 1-5 之间客户的流失量较大，也就是说刚开通电信业务 1 至 5 个月的用户容易流失，并且随着 tenure 任期的增加流失的概率越来越小

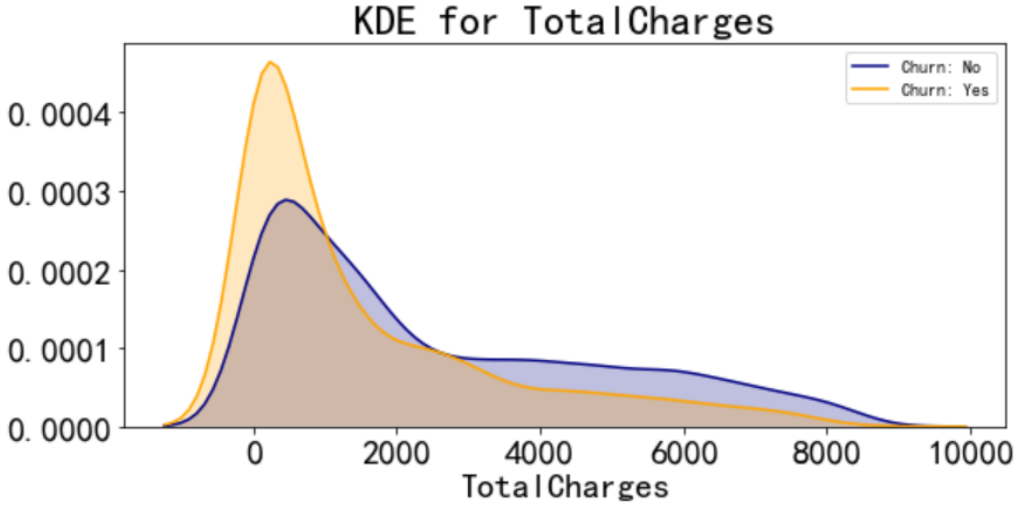
2.使用核密度估计展现 churn 特征的分布情况

```
def kdeplot(feature,xlabel):
    plt.figure(figsize=(9, 4))
    plt.title("KDE for {0}".format(feature))
    ax0 = sns.kdeplot(df[df['Churn'] == 'No'][feature].dropna(), color= 'navy', label= 'Churn: No', shade='True')
    ax1 = sns.kdeplot(df[df['Churn'] == 'Yes'][feature].dropna(), color= 'orange', label= 'Churn: Yes',shade='True')
    plt.xlabel(xlabel)
    #设置字体大小
    plt.rcParams.update({'font.size': 20})
    plt.legend(fontsize=10)
for i in numeric_features:
    kdeplot(i,i)
plt.show()
```



MonthlyCharges 核密度估计图

如图可以看出月消费在 70 至 110 之间的客户更容易流失



TotalCharges 核密度估计图

如图可以看出总消费在 1300 以下的客户更容易流失

5.8 分析总结

一、数据分析结果

对于分类属性：

Partner：单身用户更容易流失

OnlineSecurity：没开通的客户容易流失。

OnlineBackup：没开通在线备份服务的客户容易流失。

DeviceProtection：没开通设备保护业务的用户比较容易流失

TechSupport：没开通技术支持服务的用户容易流失。

StreamingTV：没有开通网络电视服务的用户更容易流失，但较不显著

StreamingMovies：没有开通网络电视服务的用户更容易流失，但较不显著

Contract：月份签订合同的用户最容易流失。签约越久越不容易流失

PaperlessBilling ：开通电子账单的用户较容易流失

PaymentMethod：使用电子支票支付的人更容易流失

SeniorCitizen 分析：年轻用户在流失、留存人数占比都高

Dependents：经济不独立的用户更容易流失

Internerservice：办理了 Fiber opti 的客户容易流失

对于数值属性

1.任期 1-5 之间客户的流失量较大，也就是说刚开通电信业务 1 至 5 个月的用户容易流失，并且随着 tenure 任期的增加流失的概率越来越小

2.月消费在 70 至 110 之间的客户更容易流失

3.总消费在 1300 以下的客户更容易流失

二、业务建议：

1. **针对不同用户属性优化服务套餐与推广策略**

在线安全（OnlineSecurity）、在线备份（OnlineBackup）、设备保护（DeviceProtection）、技术支持（TechSupport）服务方面：加强对这些增值服务功能和优势的宣传推广，通过短信、APP 推送、客服介绍等多种方式向未开

通用户详细说明服务能为其带来的保障，如数据安全、设备维修便利等，鼓励用户开通，可提供限时免费体验等优惠活动吸引用户尝试。

合同签订（Contract）：针对签订短期合同（尤其是按月签订合同）的用户，在合同即将到期前，主动联系用户，提供更有吸引力的长期合同优惠套餐，比如给予一定的费用折扣、额外的流量或通话时长赠送等，引导用户签订较长期限的合同，降低因合同短期化导致的高流失风险。

老年用户与经济不独立用户（SeniorCitizen、Dependents）：

针对老年用户，简化套餐内容和业务办理流程，提供专门的老年服务热线，客服人员用更耐心、通俗易懂的方式解答疑问和处理问题；推出适合老年人的优惠套餐，比如包含亲情通话时长、养生类资讯服务等，提升老年用户的使用体验和留存率。对于经济不独立的用户，设计价格亲民、性价比高的基础套餐，同时可与相关机构合作，推出学生套餐（针对学生群体等经济不独立用户），包含学习类应用流量优惠、学习资源免费获取等权益，满足其核心需求，减少流失

2. **基于用户任期和消费情况的服务优化**

任期方面（tenure）：对于新开通电信业务 1 - 5 个月的用户，在这个关键阶段加强客户关怀，定期发送使用指南、业务介绍、优惠活动等信息，帮助用户更好地了解和使用电信服务；设立新用户专属客服团队，快速响应和解决新用户遇到的问题，提高新用户的满意度，降低早期流失风险。随着任期增加，持续为老用户提供差异化的福利，如根据不同任期阶段给予不同等级的积分奖励、优先参与新业务体验等，激励老用户长期留存。

消费情况方面：

月消费（70 - 110 之间）与总消费（1300 以下）的客户：分析这类用户的消费结构，识别其主要使用的业务和未充分利用的业务，针对性地调整套餐内容，如对于月消费接近上限但流量不够用的用户，推荐升级流量包并给予一定的价格优惠；对于总消费较低的用户，推出小额消费套餐升级计划，以少量费用增加更多实用服务，提高用户消费体验，避免因性价比不高而流失。

三、营销策略

精准营销活动

基于用户画像的个性化推荐：综合用户的各类属性（如单身与否、是否开通增值服务、合同期限、年龄、消费情况等）以及行为数据，构建详细的用户画像，利用大数据算法实现精准营销。例如，向单身且月消费在 70 - 110 的年轻用户推荐包含社交流量优惠、在线娱乐服务（如 StreamingTV 等）开通优惠的套餐组合；向有经济不独立用户的家庭推荐家庭共享套餐，整合通话、流量等资源，并给予设备保护等增值服务的折扣优惠。

营销时机把握：根据用户任期情况，在用户开通业务后的第 1 个月、第 3 个月、第 5 个月等关键流失风险节点，推送有吸引力的优惠活动，如赠送话费、流量、限时免费开通增值服务等，帮助用户顺利度过高流失风险期；对于合同即将到期的用户，提前 1 - 2 个月进行针对性的续约营销，推出专属的续约优惠套餐，强调续约的好处，如费用减免、服务升级等。

套餐组合与捆绑销售策略

增值服务与基础套餐捆绑：将容易导致用户流失的未开通增值服务（如 OnlineSecurity、OnlineBackup 等）与基础套餐进行合理捆绑销售，设置不同档次的捆绑套餐，以整体更优惠的价格吸引用户选择，比如购买包含一定通话时

长和流量的基础套餐，额外加少量费用即可开通多项增值服务，让用户感受到更高的性价比，提高用户对整体服务的依赖度。

六、创新性

1.在数据处理时没有使用预估值填充，而是按照业务逻辑进行分析填充
2.对比了采样与拆分数据集的操作顺序对预测结果的影响
3.使用管道来进行数据处理与训练
4．使用核密度估计展现了数值特征的样本分布情况

七、项目总结

深入了解了电信用户流失的关键因素，通过严谨的数据挖掘流程，清晰地识别出不同用户属性和消费行为与流失率之间的紧密联系，为精准营销和服务优化提供了坚实的数据支撑。
在数据处理和分析方法上积累了丰富经验，创新性地应用多种技术手段，如基于业务逻辑的填充方法、对比实验、管道技术以及核密度估计等，提升了项目团队的数据处理能力和分析水平，拓展了数据分析的思路和方法库。
模型的解释性方面还可以进一步加强。虽然使用了随机森林等算法进行特征选择和模型构建，但对于一些复杂的模型结果，向业务人员和非技术人员解释时仍存在一定困难，需要在后续项目中探索更直观、易懂的模型解释方法，以便更好地推动数据分析成果的落地应用。
改进模型优化与评估：
持续探索和尝试其他先进的机器学习和数据挖掘算法，结合本项目的业务特点和数据特征，寻找更优的模型组合和参数设置，进一步提高流失预测的准确性和稳定性。同时，加强对模型的实时监测和动态调整能力，随着业务的发展和数据的变化，及时更新和优化模型，以适应不断变化的市场环境和用户需求。
完善模型评估指标体系，除了常用的 ROC AUC 等指标外，引入更多能够反映业务实际效益和用户体验的评估指标，如用户留存率的实际提升效果、营销策略的投资回报率等，从多个维度全面评估模型的性能和价值，为项目决策提供更全面、准确的依据。