客户流失是所有与消费者挂钩行业都会关注的点。因为发展一个新客户是需要一定成本的，一旦客户流失，成本浪费不说，挽回一个客户的成本更大。

今天分享一个电信行业的用户流失分析案例。

所以，电信行业在竞争日益激烈当下，如何挽留更多用户成为一项关键业务指标。为了更好运营用户，这就要求要了解流失用户的特征，分析流失原因，预测用户流失，确定挽留目标用户并制定有效方案。

一、提出问题

1、哪些用户可能会流失?

2、流失概率更高的用户有什么共同特征?

二、理解数据

1、采集数据

本数据集来自 DF ，数据源地址：

https://www.datafountain.cn/dataSets/35/details#

本数据集描述了电信用户是否流失以及其相关信息，共包含 7044 条数据，共 20 个字段，介绍下各个字段：

- 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）
- tenure ： 客户的职位（0-72，共 73 个职位）
- 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 三种）
- InternetService：是否开通互联网服务 （No, DSL 数字网络，fiber optic 光纤网络 三种）
- OnlineSecurity：是否开通网络安全服务 (Yes，No，No internetserive 三种）
- OnlineBackup：是否开通在线备份业务（Yes，No，No internetserive 三种）
- DeviceProtection：是否开通了设备保护业务(Yes，No，No internetserive 三种）
- TechSupport：是否开通了技术支持服务 (Yes，No，No internetserive 三种）
- StreamingTV：是否开通网络电视（Yes，No，No internetserive 三种）
- StreamingMovies：是否开通网络电影（Yes，No，No internetserive 三种）
- 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）
- MonthlyCharges：月费用
- TotalCharges：总费用
- Churn：该用户是否流失（Yes or No）

2、导入数据

In [1]:
```
import numpy as np
import pandas as pd
import os

# 输出文件夹input中存储的文件名
#print(os.listdir(r"F:\data\电信客户流失数据"))
```

In [2]:
```
# 导入相关的包
import matplotlib.pyplot as plt
import seaborn as sns
from pylab import rcParams
import matplotlib.cm as cm

import sklearn
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder                    # 编码转换
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import StratifiedShuffleSplit
```

```
from sklearn.ensemble import RandomForestClassifier          # 随机森林
from sklearn.svm import SVC, LinearSVC                       # 支持向量机
from sklearn.linear_model import LogisticRegression         # 逻辑回归
from sklearn.neighbors import KNeighborsClassifier           # KNN算法
from sklearn.naive_bayes import GaussianNB                   # 朴素贝叶斯
from sklearn.tree import DecisionTreeClassifier              # 决策树分类器
from xgboost import XGBClassifier
from catboost import CatBoostClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier

from sklearn.metrics import classification_report, precision_score, recall_score, f1_score
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import make_scorer
from sklearn.ensemble import VotingClassifier

from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
```

In [3]:
```
# 读取数据文件
telcom=pd.read_csv(r"F:\data\WA_Fn-UseC_-Telco-Customer-Churn.csv")
```

3、查看数据集信息

In [4]:
```
telcom.head(10)
```

Out[4]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	T
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	
5	9305-CDSKC	Female	0	No	No	8	Yes	Yes	Fiber optic	No	...	Yes	

In [5]:
```
# 查看数据集大小
telcom.shape
```

Out[5]: (7043, 21)

In [6]:
```
# 获取数据类型列的描述统计信息
telcom.describe()
```

Out[6]:

	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

三、数据清洗

1、查找缺失值

```
In [7]: # 查找缺失值
        pd.isnull(telcom).sum()
```

```
Out[7]: 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
```

```
In [8]: telcom["Churn"].value_counts()
```

```
Out[8]: No     5174
        Yes    1869
        Name: Churn, dtype: int64
```

数据集中有 5174 名用户没流失，有 1869 名客户流失，数据集不均衡。

2、查看数据类型

```
In [9]: # 查看数据集类型
        telcom.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
customerID          7043 non-null object
gender              7043 non-null object
SeniorCitizen       7043 non-null int64
Partner             7043 non-null object
Dependents          7043 non-null object
tenure              7043 non-null int64
PhoneService        7043 non-null object
MultipleLines       7043 non-null object
InternetService     7043 non-null object
OnlineSecurity      7043 non-null object
OnlineBackup        7043 non-null object
DeviceProtection    7043 non-null object
TechSupport         7043 non-null object
StreamingTV         7043 non-null object
StreamingMovies     7043 non-null object
Contract            7043 non-null object
PaperlessBilling    7043 non-null object
PaymentMethod       7043 non-null object
MonthlyCharges      7043 non-null float64
TotalCharges        7043 non-null object
Churn               7043 non-null object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

TotalCharges 表示总费用，这里为对象类型，需要转换为 float 类型

3、转换类型

```
In [10]: # TotalCharges列转换类型
         '''
         convert_numeric=True表示强制转换数字(包括字符串)，不可转换的值变为NaN
         '''
         telcom['TotalCharges']=telcom['TotalCharges'].convert_objects(convert_numeric=True)
         telcom["TotalCharges"].dtypes
```

```
Out[10]: dtype('float64')
```

再次查找缺失值：

```
In [11]: # 再次查找是否存在缺失值
         pd.isnull(telcom["TotalCharges"]).sum()
```

```
Out[11]: 11
```

这里存在 11 个缺失值，由于数量不多我们可以直接删除这些行

4、处理缺失值

```
In [12]: # 删除缺失值所在的行
         telcom.dropna(inplace=True)
         telcom.shape

Out[12]: (7032, 21)
```

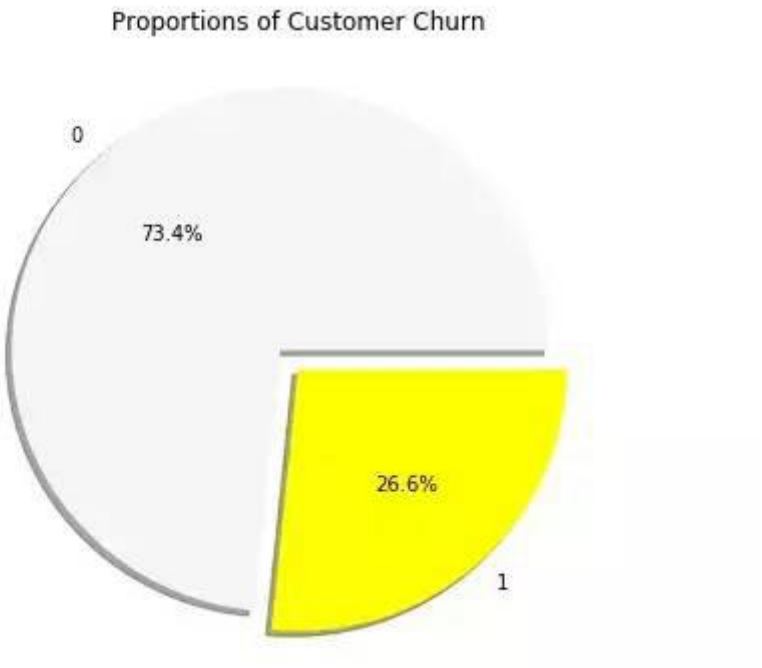
5、数据归一化处理

```
In [13]: # 数据归一化处理
         # 对Churn 列中的值 Yes和 No分别用 1和 0替换，方便后续处理
         telcom['Churn'].replace(to_replace = 'Yes', value = 1,inplace = True)
         telcom['Churn'].replace(to_replace = 'No', value = 0,inplace = True)
         telcom['Churn'].head()

Out[13]: 0    0
         1    0
         2    1
         3    0
         4    1
         Name: Churn, dtype: int64

In [14]: telcom['Churn'].replace(to_replace='Yes', value=1, inplace=True)
         telcom['Churn'].replace(to_replace='No',  value=0, inplace=True)
         telcom['Churn'].head()

Out[14]: 0    0
         1    0
         2    1
         3    0
         4    1
         Name: Churn, dtype: int64
```

四、数据可视化呈现

1、查看流失客户占比

In [15]:
```
# 查看流失客户占比
'''
画饼图参数:
labels   (每一块)饼图外侧显示的说明文字
explode  (每一块)离开中心距离
startangle   起始绘制角度,默认图是从x轴正方向逆时针画起,如设定=90则从y轴正方向画起
shadow   是否阴影
labeldistance label  绘制位置,相对于半径的比例, 如<1则绘制在饼图内侧
autopct   控制饼图内百分比设置,可以使用format字符串或者format function
        '%1.1f'指小数点前后位数(没有用空格补齐)
pctdistance 类似于labeldistance,指定autopct的位置刻度
radius   控制饼图半径
'''
churnvalue=telcom["Churn"].value_counts()
labels=telcom["Churn"].value_counts().index

rcParams["figure.figsize"]=6,6
plt.pie(churnvalue,labels=labels,colors=["whitesmoke","yellow"], explode=(0.1,0),autopct='%1.1f%%', shadow=True)
plt.title("Proportions of Customer Churn")
plt.show()
```

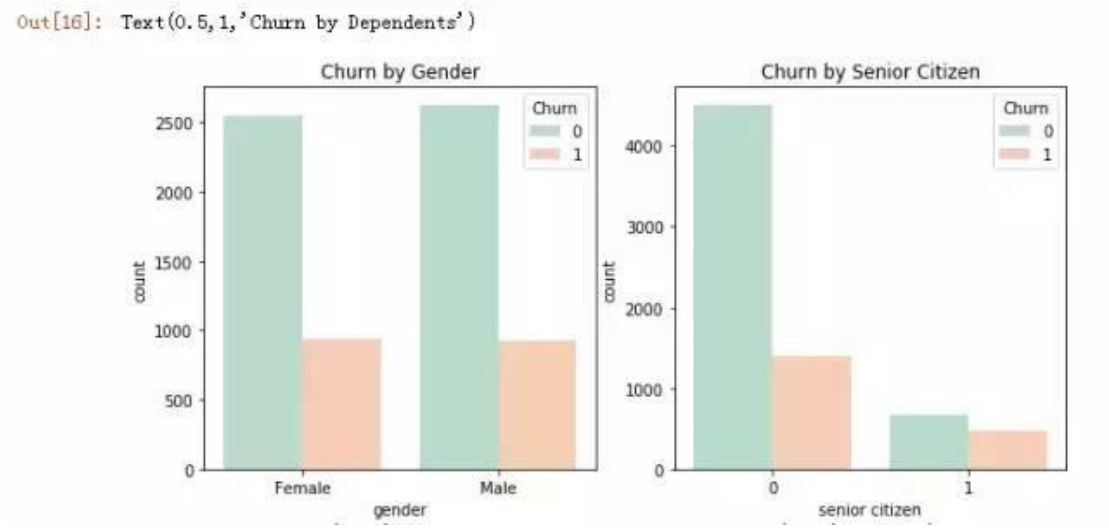


由图中结果可以看出，流失客户占整体客户的 26.6%。

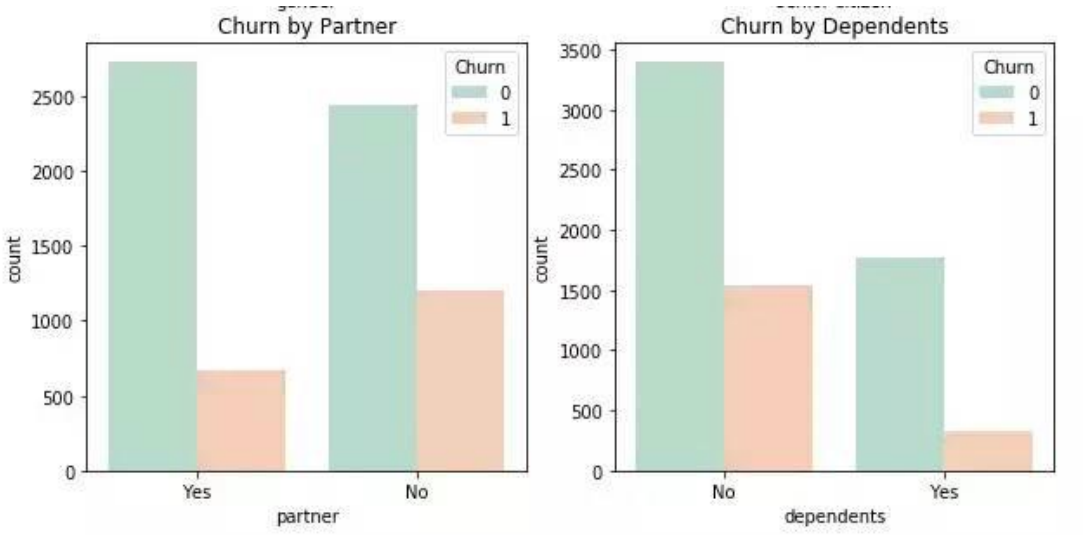
2、性别、老年人、配偶、亲属对流客户流失率的影响

In [16]:
```
# 性别、老年人、配偶、亲属对流客户流失率的影响
f, axes = plt.subplots(nrows=2, ncols=2, figsize=(10,10))

plt.subplot(2,2,1)
gender=sns.countplot(x="gender",hue="Churn",data=telcom,palette="Pastel2") # palette参数表示设置颜色，这里设置为主题色Pastel2
plt.xlabel("gender")
plt.title("Churn by Gender")

plt.subplot(2,2,2)
seniorcitizen=sns.countplot(x="SeniorCitizen",hue="Churn",data=telcom,palette="Pastel2")
plt.xlabel("senior citizen")
plt.title("Churn by Senior Citizen")

plt.subplot(2,2,3)
partner=sns.countplot(x="Partner",hue="Churn",data=telcom,palette="Pastel2")
plt.xlabel("partner")
plt.title("Churn by Partner")

plt.subplot(2,2,4)
dependents=sns.countplot(x="Dependents",hue="Churn",data=telcom,palette="Pastel2")
plt.xlabel("dependents")
plt.title("Churn by Dependents")
```

Out[16]:
```
Text(0.5,1,'Churn by Dependents')
```



性别、老年人占比结果



配偶、亲属占比结果

可以看出，男性与女性用户之间的流失情况基本没有差异，而在老年用户中流失占比明显比非老年用户更高，在所有数据中未婚与已婚人数基本持平，但未婚中流失人数比已婚中的流失人数高出了快一倍，从经济独立情况来看，经济未独立的用户流失率要远远高于经济独立的用户。

3、提取特征

In [17]:
```
# 提取特征
charges=telcom.iloc[:,1:20]
# 对特征进行编码
'''
离散特征的编码分为两种情况：
1、离散特征的取值之间没有大小的意义，比如color：[red,blue],那么就使用one-hot编码
2、离散特征的取值有大小的意义，比如size:[X,XL,XXL],那么就使用数值的映射{X:1,XL:2,XXL:3}
'''
corrDf = charges.apply(lambda x: pd.factorize(x)[0])
corrDf .head()
```

Out[17]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	1	1	1	0	1	1	1	0	0	0	1	1	1	1	1
2	1	0	1	0	2	1	1	0	1	0	0	0	0	0	0	0	1	2	2
3	1	0	1	0	3	0	0	0	1	1	1	1	0	0	1	1	2	3	3
4	0	0	1	0	2	1	1	1	0	0	0	0	0	0	0	0	0	4	4

4、构造相关性矩阵

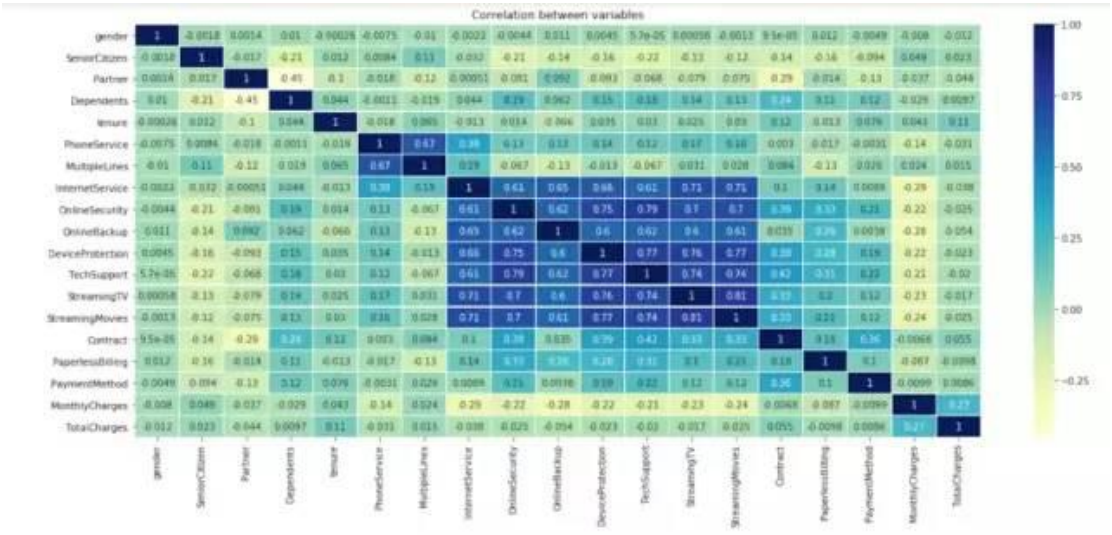
In [18]:
```
# 构造相关性矩阵
corr = corrDf.corr()
corr
```

Out[18]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	Devic
gender	1.000000	-0.001819	0.001379	0.010349	-0.000265	-0.007515	-0.010284	-0.002236	-0.004365	0.011081	
SeniorCitizen	-0.001819	1.000000	-0.016957	-0.210550	0.012240	0.008392	0.113769	-0.032160	-0.210546	-0.144782	
Partner	0.001379	-0.016957	1.000000	-0.452269	-0.100513	-0.018397	-0.118037	-0.000513	-0.081078	0.091536	
Dependents	0.010349	-0.210550	-0.452269	1.000000	0.044138	-0.001078	-0.019178	0.044030	0.188889	0.061970	
tenure	-0.000265	0.012240	-0.100513	0.044138	1.000000	-0.017864	0.064580	-0.012924	0.014436	-0.066232	
PhoneService	-0.007515	0.008392	-0.018397	-0.001078	-0.017864	1.000000	0.674824	0.387266	0.125544	0.129432	
MultipleLines	-0.010284	0.113769	-0.118037	-0.019178	0.064580	0.674824	1.000000	0.186852	-0.066695	-0.130978	
InternetService	-0.002236	-0.032160	-0.000513	0.044030	-0.012924	0.387266	0.186852	1.000000	0.607412	0.650084	
OnlineSecurity	-0.004365	-0.210546	-0.081078	0.188889	0.014436	0.125544	-0.066695	0.607412	1.000000	0.621270	

5、使用热地图显示相关系数

In [19]:
```
# 使用热地图显示相关系数
'''
heatmap     使用热地图展示系数矩阵情况
linewidths  热力图矩阵之间的间隔大小
annot       设定是否显示每个色块的系数值
'''
plt.figure(figsize=(20,16))
ax = sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns,
                 linewidths=0.2, cmap="YlGnBu",annot=True)
plt.title("Correlation between variables")
```



结论：

从上图可以看出，互联网服务、网络安全服务、在线备份业务、设备保护业务、技术支持服务、网络电视和网络电影之间存在较强的相关性，多线业务和电话服务之间也有很强的相关性，并且都呈强正相关关系。

6、使用 one-hot 编码

```
In [20]: # 使用one-hot编码
         tel_dummies = pd.get_dummies(telcom.iloc[:,1:21])
         tel_dummies.head()
```

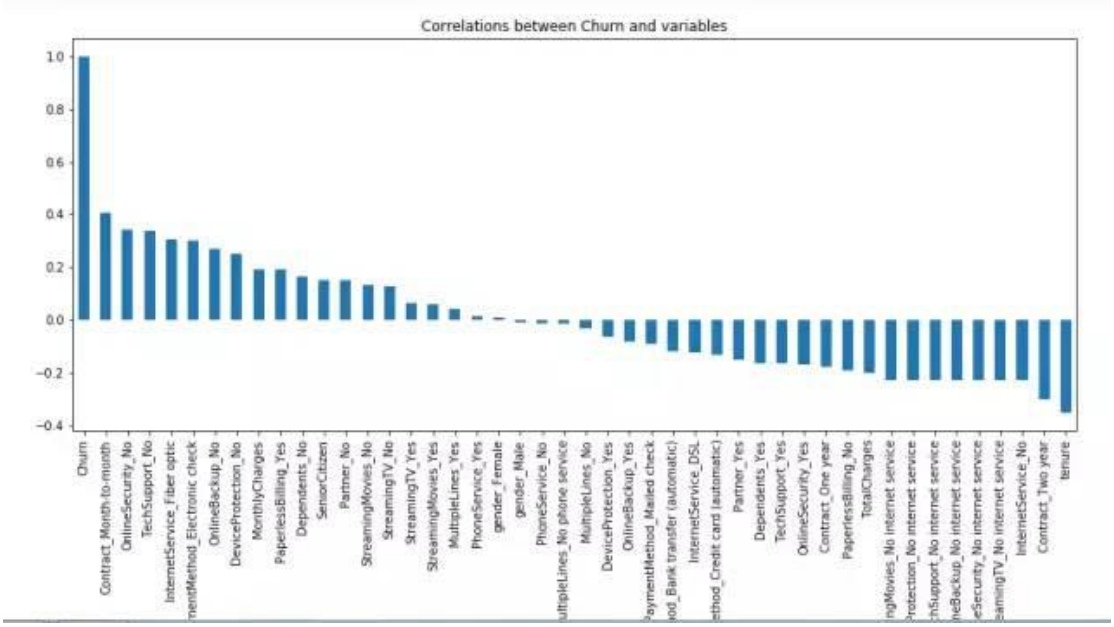
Out[20]:

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges	Churn	gender_Female	gender_Male	Partner_No	Partner_Yes	Dependents_No	...	StreamingMovies_Y
0	0	1	29.85	29.85	0	1	0	0	1	1	...	
1	0	34	56.95	1889.50	0	0	1	1	0	1	...	
2	0	2	53.85	108.15	1	0	1	1	0	1	...	
3	0	45	42.30	1840.75	0	0	1	1	0	1	...	
4	0	2	70.70	151.65	1	1	0	1	0	1	...	

5 rows × 46 columns

7、电信用户是否流失与各变量之间的相关性

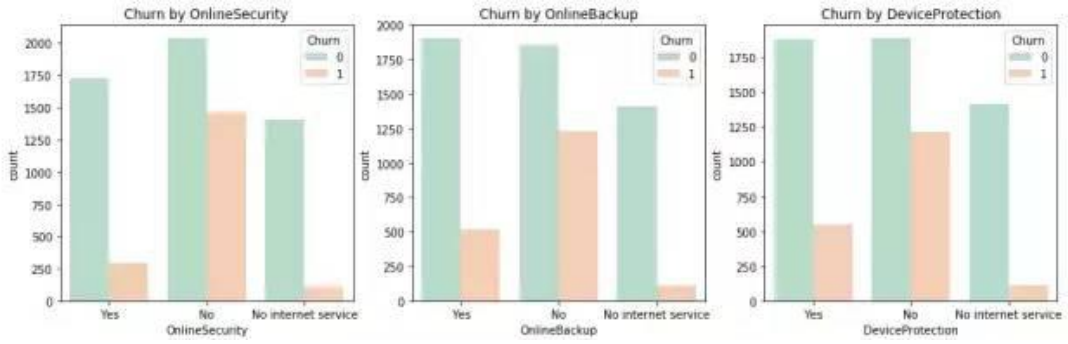
```
In [21]: # 电信用户是否流失与各变量之间的相关性
         plt.figure(figsize=(15,8))
         tel_dummies.corr()['Churn'].sort_values(ascending = False).plot(kind='bar')
         plt.title("Correlations between Churn and variables")
```

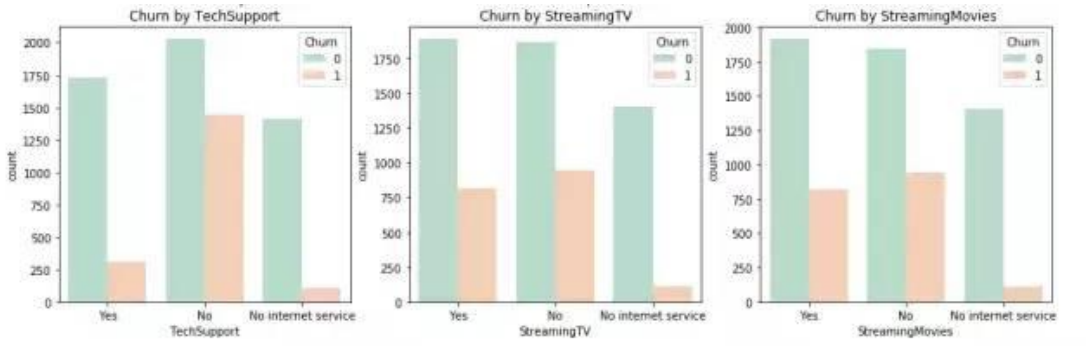


由图上可以看出，变量 gender 和 PhoneService 处于图形中间，其值接近于 0 ，这两个变量对电信客户流失预测影响非常小，可以直接舍弃。

8、网络安全服务、在线备份业务、设备保护业务、技术支持服务、网络电视、网络电影和无互联网服务对客户流失率的影响

```
In [22]: # 网络安全服务、在线备份业务、设备保护业务、技术支持服务、网络电视、网络电影和无互联网服务对客户流失率的影响
covariables=["OnlineSecurity", "OnlineBackup", "DeviceProtection", "TechSupport", "StreamingTV", "StreamingMovies"]
fig,axes=plt.subplots(nrows=2,ncols=3,figsize=(16,10))
for i, item in enumerate(covariables):
    plt.subplot(2,3,(i+1))
    ax=sns.countplot(x=item,hue="Churn",data=telcom,palette="Pastel2",order=["Yes","No","No internet service"])
    plt.xlabel(str(item))
    plt.title("Churn by "+ str(item))
    i=i+1
plt.show()
```



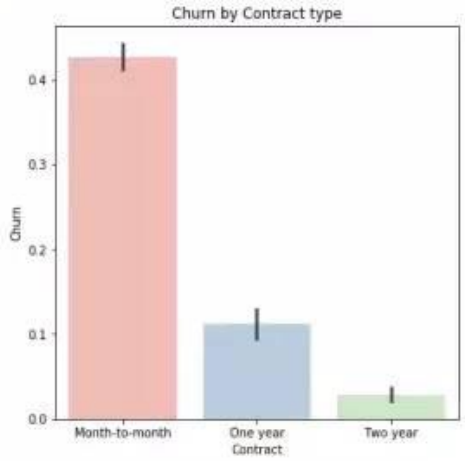


由上图可以看出，在网络安全服务、在线备份业务、设备保护业务、技术支持服务、网络电视和网络电影六个变量中，没有互联网服务的客户流失率值是相同的，都是相对较低。

这可能是因为以上六个因素只有在客户使用互联网服务时才会影响客户的决策，这六个因素不会对不使用互联网服务的客户决定是否流失产生推论效应。

9、签订合同方式对客户流失率的影响

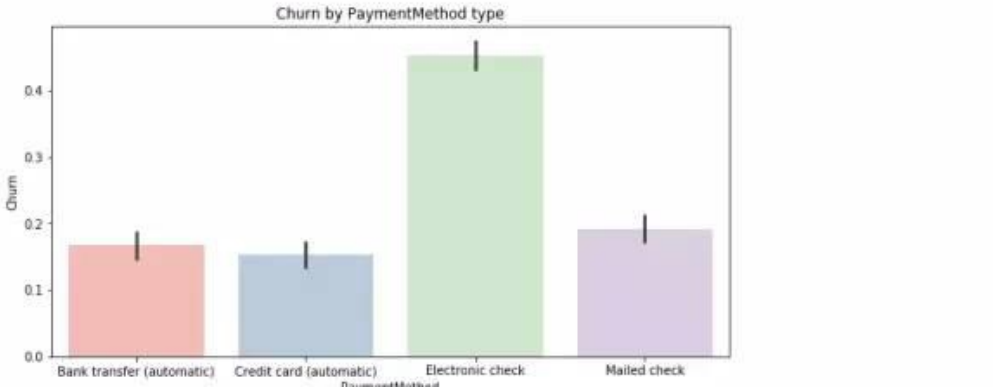
```
In [23]: # 签订合同方式对客户流失率的影响
         sns.barplot(x="Contract",y="Churn", data=telcom, palette="Pastel1", order= ['Month-to-month', 'One year', 'Two year'])
         plt.title("Churn by Contract type")
Out[23]: Text(0.5,1,'Churn by Contract type')
```



由图上可以看出，签订合同方式对客户流失率影响为：按月签订 > 按一年签订 > 按两年签订，这可能表明，设定长期合同对留住现有客户更有效。

10、付款方式对客户流失率的影响

```
In [24]: # 付款方式对客户流失率的影响
plt.figure(figsize=(10,5))
sns.barplot(x="PaymentMethod",y="Churn", data=telcom, palette="Pastel1",
            order= ['Bank transfer (automatic)', 'Credit card (automatic)', 'Electronic check','Mailed check'])
plt.title("Churn by PaymentMethod type")

Out[24]: Text(0.5,1,'Churn by PaymentMethod type')
```



由图上可以看出，在四种支付方式中，使用 Electronic check 的用户流流失率最高，其他三种支付方式基本持平，因此可以推断电子账单在设计上影响用户体验。

五、数据预处理

由前面结果可知，CustomerID 表示每个客户的随机字符，对后续建模不影响，我这里选择删除 CustomerID 列；gender 和 PhoneService 与流失率的相关性低，可直接忽略。

```
In [26]: telcomvar=telcom.iloc[:,2:20]
telcomvar.drop('PhoneService',axis=1, inplace=True)

# 提取ID
telcom_id = telcom['customerID']

telcomvar.head()
```

Out[26]:

	SeniorCitizen	Partner	Dependents	tenure	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	Strea
0	0	Yes	No	1	No phone service	DSL	No	Yes	No	No	No	
1	0	No	No	34	No	DSL	Yes	No	Yes	No	No	
2	0	No	No	2	No	DSL	Yes	Yes	No	No	No	
3	0	No	No	45	No phone service	DSL	Yes	No	Yes	Yes	No	
4	0	No	No	2	No	Fiber optic	No	No	No	No	No	

对客户的职位、月费用和总费用进行去均值和方差缩放，对数据进行标准化：

```
In [27]: # 对客户的职位、月费用和总费用进行去均值和方差缩放，对数据进行标准化
         """
         标准化数据，保证每个维度的特征数据方差为1，均值为0，使得预测结果不会被某些维度过大的特征值而主导。
         """
         scaler = StandardScaler(copy=False)
         # fit_transform()的作用就是先拟合数据，然后转化它将其转化为标准形式
         scaler.fit_transform(telcomvar[['tenure','MonthlyCharges','TotalCharges']])

Out[27]: array([[-1.28024804, -1.16169394, -0.99419409],
                [ 0.06430269, -0.26087792, -0.17979982],
                [-1.23950408, -0.36392329, -0.95964911],
                ...,
                [-0.87280842, -1.17000405, -0.85451414],
                [-1.15801615,  0.31916782, -0.87209546],
                [ 1.36810945,  1.35793167,  2.01234407]])

In [28]: # transform()的作用是通过找中心和缩放等实现标准化
         telcomvar[['tenure','MonthlyCharges','TotalCharges']]=scaler.transform(telcomvar[['tenure','MonthlyCharges','TotalCharges']])
```

使用箱线图查看数据是否存在异常值：

```
In [29]: # 使用箱线图查看数据是否存在异常值
         plt.figure(figsize = (8,4))
         numbox = sns.boxplot(data=telcomvar[['tenure','MonthlyCharges','TotalCharges']], palette="Set2")
         plt.title("Check outliers of standardized tenure, MonthlyCharges and TotalCharges")

Out[29]: Text(0.5,1,'Check outliers of standardized tenure, MonthlyCharges and TotalCharges')
```

由以上结果可以看出，在三个变量中不存在明显的异常值。

查看对象类型字段中存在的值：

In [30]:
```
# 查看对象类型字段中存在的值
def uni(columnlabel):
    print(columnlabel,'--' ,telcomvar[columnlabel].unique())  # unique函数去除其中重复的元素，返回唯一值

telcomobject=telcomvar.select_dtypes(['object'])
for i in range(0,len(telcomobject.columns)):
    uni(telcomobject.columns[i])
```

```
Partner -- ['Yes' 'No']
Dependents -- ['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)']
```

综合之前的结果来看，在六个变量中存在 No internet service，即无互联网服务对客户流失率影响很小，这些客户不使用任何互联网产品，因此可以将 No internet service 和 No 是一样的效果，可以使用 No 替代 No internet service。

In [31]:
```
# 替换值
telcomvar.replace(to_replace='No internet service', value='No', inplace=True)
telcomvar.replace(to_replace='No phone service', value='No', inplace=True)
for i in range(0,len(telcomobject.columns)):
    uni(telcomobject.columns[i])
```

```
Partner -- ['Yes' 'No']
Dependents -- ['No' 'Yes']
MultipleLines -- ['No' 'Yes']
InternetService -- ['DSL' 'Fiber optic' 'No']
OnlineSecurity -- ['No' 'Yes']
OnlineBackup -- ['Yes' 'No']
DeviceProtection -- ['No' 'Yes']
TechSupport -- ['No' 'Yes']
StreamingTV -- ['No' 'Yes']
StreamingMovies -- ['No' 'Yes']
Contract -- ['Month-to-month' 'One year' 'Two year']
PaperlessBilling -- ['Yes' 'No']
PaymentMethod -- ['Electronic check' 'Mailed check' 'Bank transfer (automatic)'
 'Credit card (automatic)']
```

使用 Scikit-learn 标签编码,将分类数据转换为整数编码：

In [32]:
```
# 使用Scikit-learn标签编码，将分类数据转换为整数编码
def labelencode(columnlabel):
    telcomvar[columnlabel] = LabelEncoder().fit_transform(telcomvar[columnlabel])

for i in range(0,len(telcomobject.columns)):
    labelencode(telcomobject.columns[i])

for i in range(0,len(telcomobject.columns)):
    uni(telcomobject.columns[i])
```

```
Partner -- [1 0]
Dependents -- [0 1]
MultipleLines -- [0 1]
InternetService -- [0 1 2]
OnlineSecurity -- [0 1]
OnlineBackup -- [1 0]
DeviceProtection -- [0 1]
TechSupport -- [0 1]
StreamingTV -- [0 1]
StreamingMovies -- [0 1]
Contract -- [0 1 2]
PaperlessBilling -- [1 0]
PaymentMethod -- [2 3 0 1]
```

六、构建模型

1、建立训练数据集和测试数据集

In [33]:
```
"""
我们需要将数据集拆分为训练集和测试集以进行验证。
由于我们所拥有的数据集是不平衡的，所以最好使用分层交叉验证来确保训练集和测试集都包含每个类样本的保留人数。
交叉验证函数StratifiedShuffleSplit，功能是从样本数据中随机按比例选取训练数据（train）和测试数据（test）
参数 n_splits是将训练数据分成train/test对的组数，可根据需要进行设置，默认为10
参数test_size和train_size是用来设置train/test对中train和test所占的比例
参数 random_state控制是将样本随机打乱
"""
X=telcomvar
y=telcom["Churn"].values

sss=StratifiedShuffleSplit(n_splits=5, test_size=0.2, random_state=0)
print(sss)
print("训练数据和测试数据被分成的组数：",sss.get_n_splits(X,y))
```

```
StratifiedShuffleSplit(n_splits=5, random_state=0, test_size=0.2,
            train_size=None)
训练数据和测试数据被分成的组数： 5
```

In [34]:
```
# 建立训练数据和测试数据
for train_index, test_index in sss.split(X, y):
    print("train:", train_index, "test:", test_index)
    X_train,X_test=X.iloc[train_index], X.iloc[test_index]
    y_train,y_test=y[train_index], y[test_index]
```

```
train: [3780 1588 2927 ..., 3956 6130 6814] test: [5126 2423 2498 ..., 6703 6618 6010]
train: [6916 6953 5388 ..., 6156 3262 3471] test: [4097 4734 2309 ..., 1278 1724 5508]
train: [1218 2877 3756 ...,  848 4568 6967] test: [ 133 1822 5303 ..., 3150 5611 4569]
train: [2552 4723 2055 ..., 4030 2165 1994] test: [ 233  438 4434 ..., 4625 1121 3422]
train: [4040 1561 6463 ..., 2550 6727 4009] test: [4581 3898 3153 ..., 2095 1765 2249]
```

In [35]:
```
# 输出数据集大小
print('原始数据特征：', X.shape,
      '训练数据特征：',X_train.shape,
      '测试数据特征：',X_test.shape)

print('原始数据标签：', y.shape,
      '    训练数据标签：',y_train.shape,
      '    测试数据标签：',y_test.shape)
```

```
原始数据特征： (7032, 17) 训练数据特征： (5625, 17) 测试数据特征： (1407, 17)
原始数据标签： (7032,)    训练数据标签： (5625,)    测试数据标签： (1407,)
```

2、选择机器学习算法

In [36]:
```
# 使用分类算法，这里选用10种分类算法
Classifiers=[["Random Forest",RandomForestClassifier()],
             ["Support Vector Machine",SVC()],
             ["LogisticRegression",LogisticRegression()],
             ["KNN",KNeighborsClassifier(n_neighbors=5)],
             ["Naive Bayes",GaussianNB()],
             ["Decision Tree",DecisionTreeClassifier()],
             ["AdaBoostClassifier", AdaBoostClassifier()],
             ["GradientBoostingClassifier", GradientBoostingClassifier()],
             ["XGB", XGBClassifier()],
             ["CatBoost", CatBoostClassifier(logging_level='Silent')]
]
```

3、训练模型

In [37]:
```
Classify_result=[]
names=[]
prediction=[]
for name,classifier in Classifiers:
    classifier=classifier
    classifier.fit(X_train,y_train)
    y_pred=classifier.predict(X_test)
    recall=recall_score(y_test,y_pred)
    precision=precision_score(y_test,y_pred)
    class_eva=pd.DataFrame([recall,precision])
    Classify_result.append(class_eva)
    name=pd.Series(name)
    names.append(name)
    y_pred=pd.Series(y_pred)
    prediction.append(y_pred)
```

4、评估模型

召回率（recall）的含义是：原本为对的当中，预测为对的比例（值越大越好，1 为理想状态）

精确率、精度（precision）的含义是：预测为对的当中，原本为对的比例（值越大越好，1 为理想状态）

F1 分数（F1-Score）指标综合了 Precision 与 Recall 的产出的结果

F1-Score 的取值范围从 0 到 1 的，1 代表模型的输出最好，0 代表模型的输出结果最差。

In [38]:
```
# 评估模型

names=pd.DataFrame(names)
names=names[0].tolist()
result=pd.concat(Classify_result,axis=1)
result.columns=names
result.index=["recall","precision","f1score"]
result
```

Out[38]:

	Random Forest	Support Vector Machine	LogisticRegression	KNN	Naive Bayes	Decision Tree	AdaBoostClassifier	GradientBoostingClassifier	XGB	CatBoost
recall	0.427807	0.459893	0.548128	0.478610	0.740642	0.513369	0.548128	0.545455	0.542781	0.542781
precision	0.594796	0.696356	0.652866	0.561129	0.552894	0.498701	0.672131	0.673267	0.678930	0.695205
f1score	0.497667	0.553945	0.595930	0.516595	0.633143	0.505929	0.603829	0.602659	0.603269	0.609610

综上所述，在 10 种分类算法中朴素贝叶斯（Naive Bayes）的 F1 分数最大为 63.31%，所以使用朴素贝叶斯模型效果最好。

七、实施方案

7、实施方案

In [62]:
```
# 预测数据集特征（由于没有提供预测数据集，这里选取后10行作为需要预测的数据集）
pred_X = telcomvar.tail(10)

# 提取customerID
pre_id = telcom_id.tail(10)

# 使用朴素贝叶斯方法，对预测数据集中的生存情况进行预测
model = GaussianNB()
model.fit(X_train,y_train)
pred_y = model.predict(pred_X)

# 预测结果
predDf = pd.DataFrame({'customerID':pre_id, 'Churn':pred_y})
predDf
```

Out[62]:

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

八、结论

通过上述分析，我们可以大致勾勒出容易流失的用户特征：

老年用户与未婚且经济未独立的青少年用户更容易流失。

电话服务对用户的流失没有直接的影响。

提供的各项网络服务项目能够降低用户的流失率。

签订合同越久，用户的留存率越高。

采用 electronic check 支付的用户更易流失。

针对上述诊断结果，可有针对性的对此提出建议：

推荐老年用户与青少年用户采用数字网络，且签订 2 年期合同（可以各种辅助优惠等营销手段来提高 2 年期合同的签订率），若能开通相关网络服务可增加用户粘性，因此可增加这块业务的推广，同时考虑改善电子账单支付的用户体验。

最后，分享源码：
https://paste.ubuntu.com/p/tsZjWrVvY9/

数据源地址：

https://www.datafountain.cn/datasets/35