**Kaggle电信用户流失分类**

1）样本数据分析

该实例数据同样来自Kaggle，它的每一条数据为一个用户的信息，共有21个有效字段，其中，最后一个字段Churn标志该用户是否流失。运用numpy和pandas等工具对数据进行初步分析，尽量理解特征之间的关系。

* 分析实验要求
* 分析实验数据的完整性、重复性
* 分析实验数据各特征之间的关联关系

电信用户流失各特征含义

|  |
| --- |
| customerID：用户ID；  gender：性别（Female & Male）；  SeniorCitizen：老年用户（1表示是，0表示不是）；  Partner：伴侣用户（Yes or No）；  Dependents：亲属用户（Yes or No）；  tenure：在网时长（0-72月）；  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）。 |

**# 观察是否有重复值**

df.customerID.duplicated().sum()

**# 观察特征的取值情况**

df.nunique()

2）用户画像

运用各类分析工具对流失用户和非流失用户进行特征分析，说明流失用户和非流失用户的特点。

* 分析各特征与用户是否流失的关系
* 综合上述关系，给出流失用户和非流失用户的特点

### #流失用户与非流失用户的个人信息对比：

import matplotlib.pyplot as plt

import seaborn as sns

fig, axes = plt.subplots(2, 2, figsize=(10, 8))

sns.countplot(x='gender', data=df, hue='Churn', ax=axes[0][0])

sns.countplot(x='SeniorCitizen', data=df, hue='Churn', ax=axes[0][1])

sns.countplot(x='Partner', data=df, hue='Churn', ax=axes[1][0])

sns.countplot(x='Dependents', data=df, hue='Churn', ax=axes[1][1])

### #流失用户与非流失用户的服务订阅信息对比

### plt.rc('font', family='SimHei')

### plt.title("在网时长密度图")

### ax1 = sns.kdeplot(df[df['Churn'] == 'Yes']['tenure'], color='r', linestyle='-', label='Churn:Yes')

### ax1 =sns.kdeplot(df[df['Churn'] == 'No']['tenure'], color='b', linestyle='--', label='Churn:No')

3）分类算法建模及分析

运用各类分类算法对数据进行建模，包括：逻辑回归模型、决策树模型、随机森林模型。并用准确率和AUC指标对它们的预测效果进行评估。

* 编码、特征提取
* 数据预处理
* 学习分类算法的应用方法
* 对样本数据进行建模
* 用准确度指标和AUC指标评估模型
* 记录并分析实验结果

### #编码，提取特征

df\_clu = df.drop(['Unnamed: 0', 'customerID', 'Churn'], axis=1)

labels = df['Churn']

**# 二值对象型特征转换成数值型**

df\_clu['gender'] = df\_clu['gender'].replace('Male', 1).replace('Female', 0)

df\_clu['Partner'] = df\_clu['Partner'].replace('Yes', 1).replace('No', 0)

df\_clu['Dependents'] = df\_clu['Dependents'].replace('Yes', 1).replace('No', 0)

df\_clu['PhoneService'] = df\_clu['PhoneService'].replace('Yes', 1).replace('No', 0)

df\_clu['PaperlessBilling'] = df\_clu['PaperlessBilling'].replace('Yes', 1).replace('No', 0)

labels = labels.replace('Yes', 1).replace('No', 0)

**# 离散的，可用距离度量的对象型特征转化为数值型**

df\_clu['Contract'] = df\_clu['Contract'].replace("Month-to-month", 1).replace("One year", 12).replace("Two year", 24)

**# 离散的，不宜用距离度量的特征用one-hot编码**

df\_clu = pd.get\_dummies(df\_clu)

df\_clu.info()

df\_clu.max()

**# 数据归一化**

df\_clu['tenure'] = ( df\_clu['tenure'] - df\_clu['tenure'].min() )/( df\_clu['tenure'].max() - df\_clu['tenure'].min() )

df\_clu['Contract'] =( df\_clu['Contract'] - df\_clu['Contract'].min() )/( df\_clu['Contract'].max() - df\_clu['Contract'].min() )

df\_clu['MonthlyCharges']=(df\_clu['MonthlyCharges']-df\_clu['MonthlyCharges'].min())/(df\_clu['MonthlyCharges'].max()-df\_clu['MonthlyCharges'].min() )

df\_clu['TotalCharges']=(df\_clu['TotalCharges']-df\_clu['TotalCharges'].min() )/( df\_clu['TotalCharges'].max() - df\_clu['TotalCharges'].min() )

**# 将数据集分成训练集和验证集**

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, classification\_report, roc\_auc\_score

X\_train, X\_test, y\_train, y\_test = train\_test\_split(df\_clu, labels, test\_size=0.3, random\_state = 1026)

**#建模并验证**

例：

# 逻辑回归模型

from sklearn.linear\_model import LogisticRegression

**源代码：**

import numpy as np

import pandas as pd

df = pd.read\_csv('E:\研一\机器学习\telecom\_users.csv')

df.customerID.duplicated().sum() # 观察是否有重复值

df.nunique() # 观察特征的取值情况

#流失用户与非流失用户的个人信息对比：

import matplotlib.pyplot as plt

import seaborn as sns

fig, axes = plt.subplots(2, 2, figsize=(10, 8))

sns.countplot(x='gender', data=df, hue='Churn', ax=axes[0][0])

sns.countplot(x='SeniorCitizen', data=df, hue='Churn', ax=axes[0][1])

sns.countplot(x='Partner', data=df, hue='Churn', ax=axes[1][0])

sns.countplot(x='Dependents', data=df, hue='Churn', ax=axes[1][1])

#流失用户与非流失用户的服务订阅信息对比

plt.rc('font', family='SimHei')

plt.title("在网时长密度图")

ax1 = sns.kdeplot(df[df['Churn'] == 'Yes']['tenure'], color='r', linestyle='-', label='Churn:Yes')

ax1 =sns.kdeplot(df[df['Churn'] == 'No']['tenure'], color='b', linestyle='--', label='Churn:No')

#编码，提取特征

df\_clu = df.drop(['Unnamed: 0', 'customerID', 'Churn'], axis=1)

labels = df['Churn']

df\_clu

# 二值对象型特征转换成数值型

df\_clu['gender'] = df\_clu['gender'].replace('Male', 1).replace('Female', 0)

df\_clu['Partner'] = df\_clu['Partner'].replace('Yes', 1).replace('No', 0)

df\_clu['Dependents'] = df\_clu['Dependents'].replace('Yes', 1).replace('No', 0)

df\_clu['PhoneService'] = df\_clu['PhoneService'].replace('Yes', 1).replace('No', 0)

df\_clu['PaperlessBilling'] = df\_clu['PaperlessBilling'].replace('Yes', 1).replace('No', 0)

labels = labels.replace('Yes', 1).replace('No', 0)

# 离散的，可用距离度量的对象型特征转化为数值型

df\_clu['Contract'] = df\_clu['Contract'].replace("Month-to-month", 1).replace("One year", 12).replace("Two year", 24)

# 离散的，不宜用距离度量的特征用one-hot编码

df\_clu = pd.get\_dummies(df\_clu)

# 归一化

df\_clu['tenure'] = ( df\_clu['tenure'] - df\_clu['tenure'].min() )/( df\_clu['tenure'].max() - df\_clu['tenure'].min() )

df\_clu['Contract'] =( df\_clu['Contract'] - df\_clu['Contract'].min() )/( df\_clu['Contract'].max() - df\_clu['Contract'].min() )

df\_clu['MonthlyCharges']=(df\_clu['MonthlyCharges']-df\_clu['MonthlyCharges'].min())/(df\_clu['MonthlyCharges'].max()-df\_clu['MonthlyCharges'].min() )

df\_clu['TotalCharges']=(df\_clu['TotalCharges']-df\_clu['TotalCharges'].min() )/( df\_clu['TotalCharges'].max() - df\_clu['TotalCharges'].min() )

df\_clu

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, classification\_report, roc\_auc\_score

X\_train, X\_test, y\_train, y\_test = train\_test\_split(df\_clu, labels, test\_size=0.3, random\_state = 1026)

# 逻辑回归模型

from sklearn.linear\_model import LogisticRegression

log = LogisticRegression()

log.fit(X\_train, y\_train)

y\_pred1 = log.predict(X\_test)

print('逻辑回归的精确度为：', accuracy\_score(y\_test, y\_pred1))

print('逻辑回归的auc为：', roc\_auc\_score(y\_test, y\_pred1, average='micro'))

print('分类报告：\n', classification\_report(y\_test, y\_pred1))

# 决策树模型

from sklearn.tree import DecisionTreeClassifier

tree = DecisionTreeClassifier()

tree.fit(X\_train, y\_train)

y\_pred2 = tree.predict(X\_test)

print('决策树的精确度为：', accuracy\_score(y\_test, y\_pred2))

print('决策树的auc为：', roc\_auc\_score(y\_test, y\_pred2, average='micro'))

print('分类报告：\n', classification\_report(y\_test, y\_pred2))

# 随机森林模型

from sklearn.ensemble import RandomForestClassifier

forests = RandomForestClassifier()

forests.fit(X\_train, y\_train)

y\_pred3 = forests.predict(X\_test)

print('随机森林的精确度为：', accuracy\_score(y\_test, y\_pred3))

print('随机森林的auc为：', roc\_auc\_score(y\_test, y\_pred3, average='micro'))

print('分类报告：\n', classification\_report(y\_test, y\_pred3))

**运行（测试）过程及结果：**















