XGboost算法甄别某电商的高质量用户

实验目标

通过本案例的学习和课后作业的练习：

1. 通过代码的实现，帮助大家快速理解机器学习整个流程；
2. 能够使用SKlearn实现XGboost算法。

你也可以将本案例相关的 ipynb 学习笔记分享到 AI Gallery Notebook (https://marketplace.huaweicloud.com/markets/aihub/notebook/list/) 版块获得成长值 (https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=9b8d7e7a-a150-449e-ac17-2dcf76d8b492)，分享方法请查看此文档 (https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=8afec58a-b797-4bf9-acca-76ed512a3acb)。

案例内容介绍

XGBoost的全称是eXtreme Gradient Boosting，它是陈天奇等人开发的一个开源机器学习项目，是经过优化的分布式梯度提升库，旨在高效、灵活且可移植，主要是用来解决有监督学习问题。

XGBoost是大规模并行boosting tree的工具，它是目前最快最好的开源 boosting tree工具包，比常见的工具包快10倍以上。在数据科学方面，有大量的Kaggle选手选用XGBoost进行数据挖掘比赛，是各大数据科学比赛的必杀武器；在工业界大规模数据方面，XGBoost的分布式版本有广泛的可移植性，支持在Kubernetes、Hadoop、SGE、MPI、 Dask等各个分布式环境上运行，使得它可以很好地解决工业界大规模数据的问题。

说到XGBoost，不得不提GBDT(Gradient Boosting Decision Tree)。因为XGBoost本质上还是一个GBDT，但是力争把速度和效率发挥到极致，所以叫X (Extreme) GBoosted。包括前面说过，两者都是boosting方法。

本案例推荐的理论学习视频：

- 《AI技术领域课程--机器学习》 XGboost (https://education.huaweicloud.com/courses/course-v1:HuaweiX+CBUCNXE086+Self-paced/courseware/56aaf4a05ebd47f497677c0c4d08a739/0ee5c248c6ce49da9dbabb17ba130cdb/)

注意事项

1. 如果您是第一次使用 JupyterLab，请查看[《ModelArts JupyterLab使用指导》(https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=03676d0a-0630-4a3f-b62c-07fba43d2857)](https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=03676d0a-0630-4a3f-b62c-07fba43d2857)了解使用方法；
2. 如果您在使用 JupyterLab 过程中碰到报错，请参考[《ModelArts JupyterLab常见问题解决办法》(https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=9ad8ce7d-06f7-4394-80ef-4dbf6cfb4be1)](https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=9ad8ce7d-06f7-4394-80ef-4dbf6cfb4be1)尝试解决问题。

实验步骤

1、导入相关库

In [1]:

```
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
from sklearn import datasets
import moxing as mox
%matplotlib inline
```

```
INFO:root:Using MoXing-v1.17.3-

INFO:root:Using OBS-Python-SDK-3.20.7
```

2、读取用户数据

In [2]:

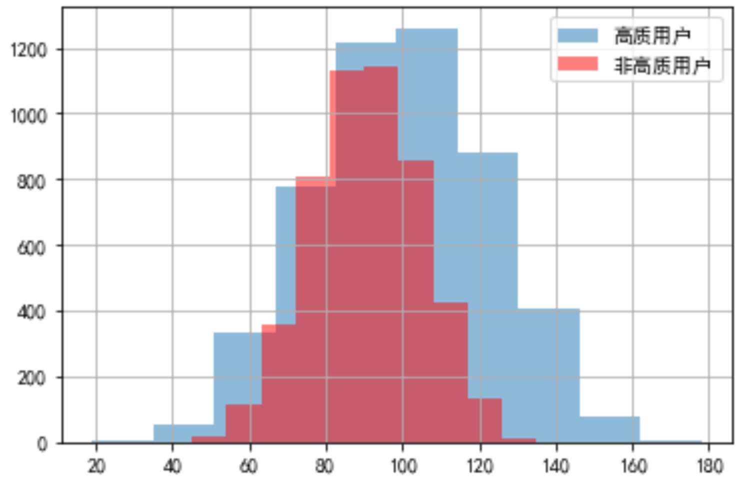
```
if not os.path.exists('data_carrier_svm.csv'):
    mox.file.copy('obs://modelarts-labs-bj4-v2/course/hwc_edu/machine_learning/datasets/XGboost/data_carrier_svm.csv',
                  'data_carrier_svm.csv')
data = pd.read_csv(r'data_carrier_svm.csv', encoding='gbk')
data.head()
```

Out[2]:

	用户标识符	网络类型	类目1消费金额	类目2消费金额	类目3消费金额	类目4消费金额	类目5消费金额	类目6消费金额	类目7消费金额	类目7消费金额.1	总消费次数	账户余额	是否高质用户
0	66069	3G	70	97	395	13	64	168	59	465	7	36	0
1	64410	3G	94	79	366	35	59	182	70	542	13	66	0
2	60110	3G	92	99	390	44	134	219	8	548	8	110	1
3	69600	4G	131	87	391	0	128	180	63	498	4	30	1
4	64683	4G	74	104	397	35	112	258	68	614	15	18	1

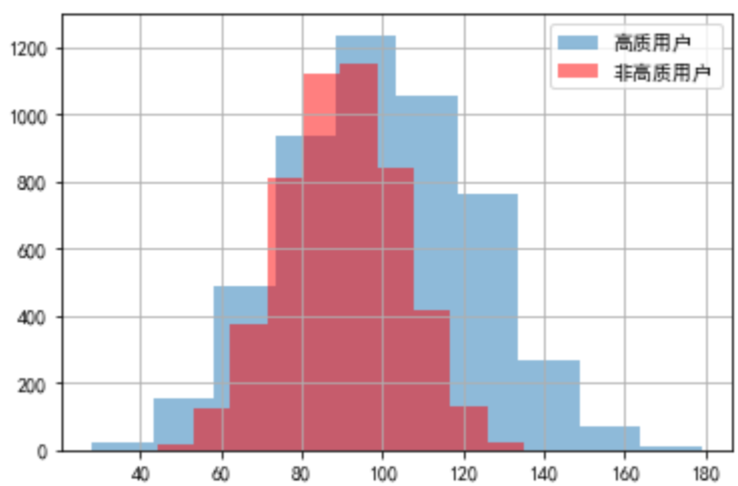
In [3]:
```
# 不同用户的类目1消费金额分布情况对比
cond = data['是否高质用户'] == 1
data[cond]['类目1消费金额'].hist(alpha=0.5, label='高质用户')
data[~cond]['类目1消费金额'].hist(color='r', alpha=0.5, label='非高质用户
')
plt.legend()
```

Out[3]: <matplotlib.legend.Legend at 0x7fa122095160>



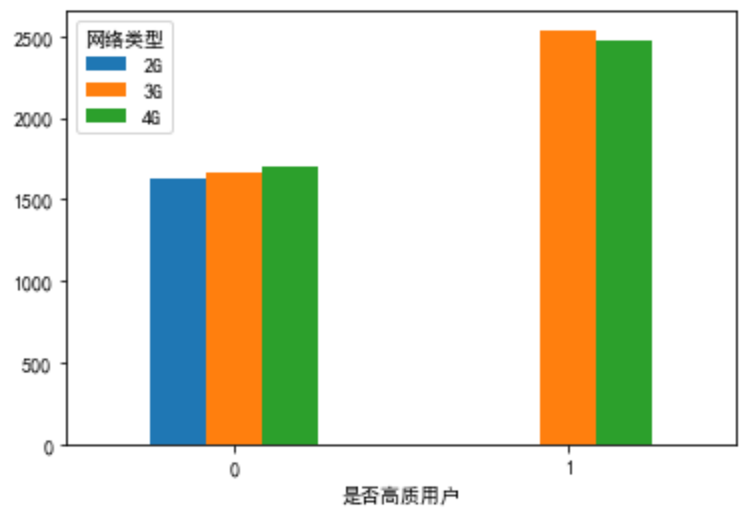
In [4]:
```
# 不同用户的类目2消费金额分布情况对比
cond = data['是否高质用户'] == 1
data[cond]['类目2消费金额'].hist(alpha=0.5, label='高质用户')
data[~cond]['类目2消费金额'].hist(color='r', alpha=0.5, label='非高质用户
')
plt.legend()
```

Out[4]: <matplotlib.legend.Legend at 0x7fa1033c5518>



In [5]:
```
# 不同用户的网络类型情况对比
grouped = data.groupby(['是否高质用户', '网络类型'])['用户标识符'].count().unstack()
grouped.plot(kind='bar', alpha=1.0, rot=0)
```

Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x7fa100e7c208>

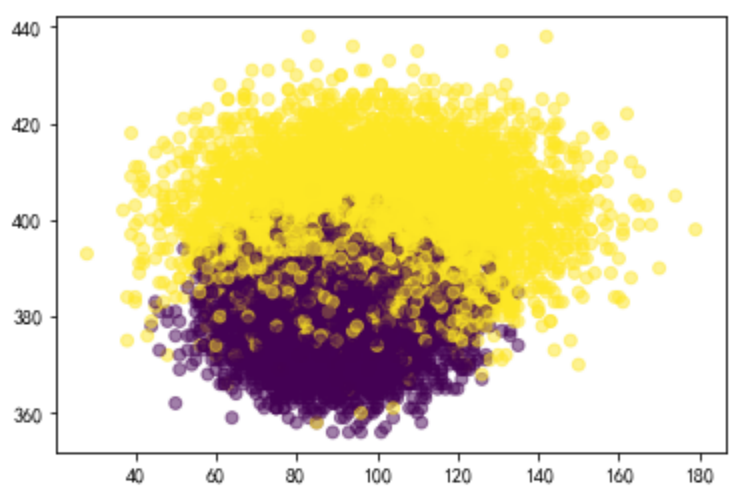


In [6]:
```
data['是否高质用户'].value_counts()
```

Out[6]:
```
1    5003
0    4997
Name: 是否高质用户, dtype: int64
```

In [7]:
```
# 生成数据可视化
y = data.loc[:, '是否高质用户']
plt.scatter(data.loc[:, '类目2消费金额'], data.loc[:, '类目3消费金额'], c=y, alpha=0.5)
```

Out[7]: <matplotlib.collections.PathCollection at 0x7fa100d6a6d8>



3、数据预处理

In [8]:
```
# 类别特征编码
X = data.loc[:, '网络类型':'账户余额']
y = data.loc[:, '是否高质用户']
print('The shape of X is {0}'.format(X.shape))
print('The shape of y is {0}'.format(y.shape))
```

```
The shape of X is (10000, 11)

The shape of y is (10000,)
```

In [9]:
```
X.head()
```

Out[9]:

	网络类型	类目1消费金额	类目2消费金额	类目3消费金额	类目4消费金额	类目5消费金额	类目6消费金额	类目7消费金额	类目7消费金额.1	总消费次数	账户余额
0	3G	70	97	395	13	64	168	59	465	7	36
1	3G	94	79	366	35	59	182	70	542	13	66
2	3G	92	99	390	44	134	219	8	548	8	110
3	4G	131	87	391	0	128	180	63	498	4	30
4	4G	74	104	397	35	112	258	68	614	15	18

In [10]:
```
from sklearn.preprocessing import OneHotEncoder


def service_mapping(cell):
    if cell == '2G':
        return 2
    elif cell == '3G':
        return 3
    elif cell == '4G':
        return 4


# 将网路类型的string型值映射为整数型
service_map = X['网络类型'].map(service_mapping)
service = pd.DataFrame(service_map)
# service_df
# 使用OncHotEncoder转化类型特征为0/1编码的多维特征
enc = OneHotEncoder()
service_enc = enc.fit_transform(service).toarray()
service_enc

# 0/1编码的多维特征的名称
service_names = enc.active_features_.tolist()
service_newname = [str(x) + 'G' for x in service_names]

service_df = pd.DataFrame(service_enc, columns=service_newname)
service_df.head()
X_enc = pd.concat([X, service_df], axis=1).drop('网络类型', axis=1)
X_enc.head()
```

```
/home/ma-user/anaconda3/envs/XGBoost-Sklearn/lib/python3.6/site-packa
ges/sklearn/preprocessing/_encoders.py:363: FutureWarning: The handli
ng of integer data will change in version 0.22. Currently, the catego
ries are determined based on the range [0, max(values)], while in the
future they will be determined based on the unique values.

If you want the future behaviour and silence this warning, you can sp
ecify "categories='auto'".

In case you used a LabelEncoder before this OneHotEncoder to convert
the categories to integers, then you can now use the OneHotEncoder di
rectly.

  warnings.warn(msg, FutureWarning)

/home/ma-user/anaconda3/envs/XGBoost-Sklearn/lib/python3.6/site-packa
ges/sklearn/utils/deprecation.py:77: DeprecationWarning: Function act
ive_features_ is deprecated; The ``active_features_`` attribute was d
eprecated in version 0.20 and will be removed 0.22.

  warnings.warn(msg, category=DeprecationWarning)
```

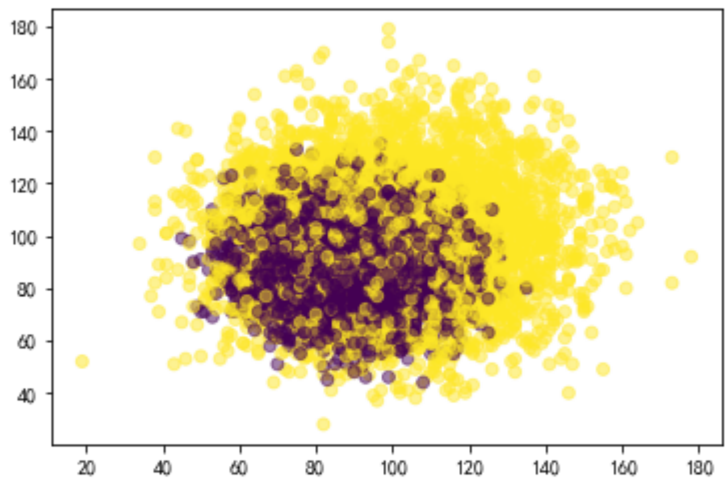
Out[10]:

	类目1消费金额	类目2消费金额	类目3消费金额	类目4消费金额	类目5消费金额	类目6消费金额	类目7消费金额	类目7消费金额.1	总消费次数	账户余额	2G	3G	4G
0	70	97	395	13	64	168	59	465	7	36	0.0	1.0	0.0
1	94	79	366	35	59	182	70	542	13	66	0.0	1.0	0.0
2	92	99	390	44	134	219	8	548	8	110	0.0	1.0	0.0
3	131	87	391	0	128	180	63	498	4	30	0.0	0.0	1.0
4	74	104	397	35	112	258	68	614	15	18	0.0	0.0	1.0

4、训练过程准备

In [11]:
```
from sklearn import metrics
from sklearn.model_selection import train_test_split, GridSearchCV, cro
ss_val_score
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier

# 分割训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X_enc, y, test_size
=0.2, random_state=112)
print('The shape of X_train is {0}'.format(X_train.shape))
print('The shape of X_test is {0}'.format(X_test.shape))
print('The shape of y_train is {0}'.format(y_train.shape))
print('The shape of y_test is {0}'.format(y_test.shape))

# 生成数据可视化
plt.scatter(X_train.iloc[:, 0], X_train.iloc[:, 1], c=y_train, alpha=0.
5)


# 交叉验证
def modelfit(alg, X_train, y_train, performCV=True, printFeatureImporta
nce=True, cv_folds=5):
    alg.fit(X_train, y_train)
    # Predict training set:
    train_predictions = alg.predict(X_train)
    train_predprob = alg.predict_proba(X_train)[:, 1]
    # Perform cross-validation:here the author calculate cross-validate
d AUC
    if performCV:
        cv_score = cross_val_score(alg, X_train, y_train, cv=cv_folds,
scoring='roc_auc')
    # Print model report:
    print("\nModel Report")
    print("Accuracy (Train): %3.4f" % metrics.accuracy_score(y_train.va
lues, train_predictions))
    ## IMPORTANT: first argument is true values, second argument is pre
dicted probabilities
    print("AUC Score (Train): %f" % metrics.roc_auc_score(y_train, trai
n_predprob))

    if performCV:
        print("CV Score : Mean - %.7g | Std - %.7g | Min - %.7g | Max -
%.7g" \
              % (np.mean(cv_score), np.std(cv_score), np.min(cv_score),
np.max(cv_score)))

    # Print Feature Importance:
    if printFeatureImportance:
        feat_imp = pd.Series(alg.feature_importances_, X_train.columns.
tolist()).sort_values(ascending=True)
        feat_imp.plot(kind='barh', title='Feature Importances')
        plt.ylabel('Feature Importance Score')
        _ = plt.xlabel('Relative importance')
```

```
The shape of X_train is (8000, 13)

The shape of X_test is (2000, 13)

The shape of y_train is (8000,)

The shape of y_test is (2000,)
```



5、模型训练及预测

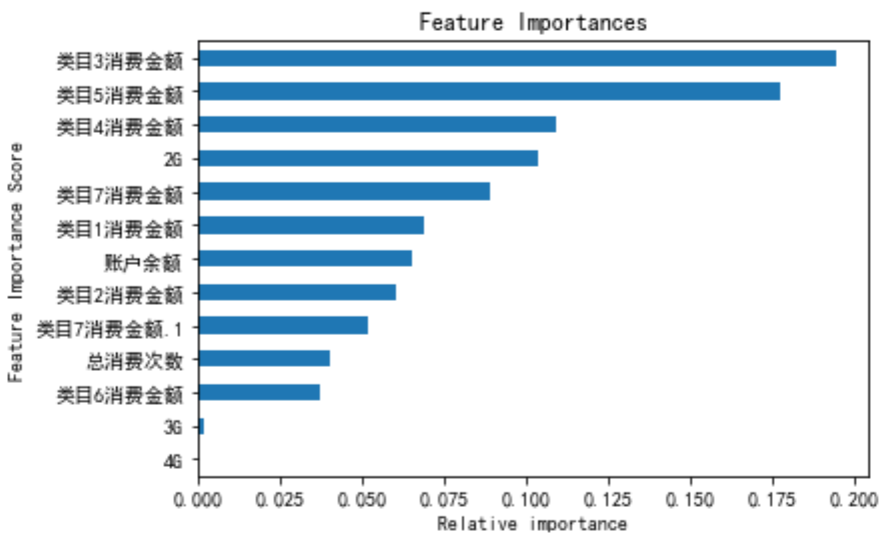
In [12]:
```
from pylab import mpl

mpl.rcParams['font.sans-serif'] = ['SimHei']  # 指定默认字体
mpl.rcParams['axes.unicode_minus'] = False  # 解决保存图像是负号'-'显示为方块的问题

# 模型实例化
clf0 = XGBClassifier()
# 在训练集上训练模型
clf0.fit(X_train, y_train)

# 在测试集上预测
y_pred = clf0.predict(X_test)

# 计算准备；率
score = metrics.accuracy_score(y_test, y_pred)
print('The accuracy score of the model is: {0}'.format(score))

# 查看混淆举证
metrics.confusion_matrix(y_test, y_pred)
```

```
The accuracy score of the model is: 0.986
```

Out[12]:
```
array([[1032,    6],
       [  22,  940]])
```

In [13]:
```
# 模型实例化
clf0 = XGBClassifier()

modelfit(clf0, X_train, y_train)
```

```
Model Report

Accuracy (Train): 0.9854

AUC Score (Train): 0.999031

CV Score : Mean - 0.9975615 | Std - 0.00049871 | Min - 0.9968794 | Ma
x - 0.9982451
```



小结

通过最终的实验分析，可以看到模型的分类准确率可以达到0.98，可以根据实际业务指标来判断模型是否达到了需要，可以通过后续的多种策略（超参数选择，模型剪枝等）进行模型性能的提升。

以上是 XGboost 的实现方法，受限于篇幅原因，本案例未完全覆盖 XGboost 的全部操作，欢迎你将更全面的 XGboost 学习笔记分享到 AI Gallery Notebook (https://marketplace.huaweicloud.com/markets/aihub/notebook/list/) 版块获得成长值 (https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=9b8d7e7a-a150-449e-ac17-2dcf76d8b492)，分享方法请查看此文档 (https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=8afec58a-b797-4bf9-acca-76ed512a3acb)。

作业

请你利用本实验中学到的知识点，完成以下编程题：

1. 请你尝试修改 XGBClassifier() 函数的 max_depth（树的深度）参数的不同取值，看看该参数的修改对模型会有怎样的影响。 (https://marketplace.huaweicloud.com/markets/aihub/notebook/detail/?id=e8c39443-1bcb-4cbf-8df3-310168616a60)
2. 请你尝试修改 XGBClassifier() 函数的 n_estimators（决策树的个数） 参数的不同取值，看看该参数的修改对模型会有怎样的影响。 (https://marketplace.huaweicloud.com/markets/aihub/notebook/detail/?id=925f9e3a-1a3f-4ff7-b2d8-74a20b43ab35)
3. 请你尝试修改 XGBClassifier() 函数的所有可调参数的不同取值，看看不同参数的不同取值组合，对模型会有怎样的影响。 (https://marketplace.huaweicloud.com/markets/aihub/notebook/detail/?id=d34e9687-dcbd-4a21-89c6-c32627110868)