2022 年首届“钉钉杯”大学生大数据挑战赛论文

基于 XGB、LGB、随机森林的 Stacking 诈骗预测模型

摘　　要

随着互联网的迅猛发展以及移动支付的普及，银行卡电信诈骗的问题愈加严峻。本文通对银行卡交易信息数据进行 EDA、特征工程、建立模型、模型调参、模型融合、模型评价，建立了基于 XGB、LGB、随机森林的 Stacking 诈骗预测模型，最终选择 Stackin 作为预测模型，模型准确率为 0.9999987，F-1 值为 0.999885。最后给客户提出建议：**用户如果能够使用 pin 交易，那么将会使欺诈发生的概率趋近于 0；当用户发现当前交易是往期交易的数倍时，用户需要谨慎思考当前是否遭遇了银行卡欺诈行为**。从 Kaggle 获取我们的代码：https://www.kaggle.com/code/zhouhua2022/card-transdata

EDA：导入需要用到的包，读取数据；查看数据基本信息，查看 7 个分类属性数据的分布情况，查看标签在 7 个分类属性条件下的分布情况。

数据预处理：第一步，对数据进行**异常值处理**。查看 3 个连续型属性的分布，发现只有极少的值是异常值，考虑到异常数值对应属性在实际生活中是切实存在的，并且异常值处理前后平均值与方差的变化在 5% 以内，因此不对数据进行异常值处理。第二步，对三个连续型属性数据进行**标准化**。第三步，对 4 个分类属性数据进行**独热编码**。第四步，**选择分类特征**。查看属性的相关系数矩阵，发现属性间相关性小，属性独立性好，故没有进一步对数据降维的必要。第五步，**数据集划分**。对数据集进行划分，训练集与测试集比例为 7:3。

模型建立：使用 LGB,XGB,KNN, 线性 SVC，朴素贝叶斯，决策树随机森林，感知机，逻辑回归，随机剃度下降 **10 个模型**对数据进行拟合并查看在测试集上的准确率，其中决策树，随机森林的准确率达到了 **100%**

模型调参与模型融合：第一步，**模型调参**。选取 XGB、LGB、KNN、决策树、随机森林 5 个准确度最高的模型，使用网格搜索调节参数。第二步，**模型融合**。从调节参数之后的 5 个模型中选出三个表现最好的 XGB、LGB、随机森林建立 Stacking 模型。

模型评价与可视化：计算 6 个模型的 $precision$、$recall$、$f1-score$，画出 6 个模型的混淆矩阵热力图、ROC 曲线，最后根据 F1 分数值选出最优模型为 Stacking 模型。

提出建议：通过标签与分类属性的数据分布情况、标签与分类属性的相关性系数、决策树特征重要性分析，给客户提出合理建议。

关键词：**XGB**；**随机森林**；**Stacking 模型**；**F1 分数值**

目　录

1 绪论

1.1 研究背景

随着互联网的迅猛发展以及移动支付的普及，多账户注册与支付方式变多导致用户信息泄露的情况越来越多，信息泄露导致银行卡交易欺诈行为增多。因此，根据用户交易信息对欺诈行为具有重要的意思。

数据集文件 $card\_transdata.csv$ 中包含了 100 万条用户银行卡交易的信息，其中包括 7 个特征和一列标签，7 个特征包含了用户交易时离家距离，交易数额变化，交易方式等信息，标签指出该笔交易是否存在欺诈行为。本文就是对银行卡交易信息进行分析，提取有效信息，建立模型，从而实现对欺诈行为的检测并对用户提出合理的建议从而减少欺诈行为的发生。

1.2 研究思路

第一步 (EDA)：导入需要用到的包，读取数据；查看数据基本信息，查看 7 个分类属性数据的分布情况，查看标签在 7 个分类属性条件下的分布情况。

第二步 (数据预处理)：首先对数据进行**异常值处理**，通过查看 3 个连续型属性的分布，发现只有极少的值是异常值，考虑到异常数值对应属性在实际生活中是切实存在的，并且异常值处理前后平均值与方差的变化在 5% 以内，因此不对数据进行异常值处理。然后对三个连续型属性数据进行**标准化**。紧接着对 4 个分类属性数据进行独热编码。之后选择分类特征，通过查看属性的相关系数矩阵，发现属性间相关性小，属性独立性好，故没有进一步对数据降维的必要。最后对数据集按 7:3 进行划分。

第三步 (模型建立)：使用 LGB,XGB,KNN, 线性 SVC，朴素贝叶斯，决策树随机森林，感知机，逻辑回归，随机剃度下降 10 个模型对数据进行拟合并查看在测试集上的准确率。

第四步 (模型调参与模型融合)：首先进行模型调参，选取 XGB、LGB、KNN、决策树、随机森林 5 个准确度最高的模型，使用网格搜索调节参数。然后进行模型融合，使用调节参数之后的 5 个模型建立 Stacking 模型。

第五步 (模型评价与可视化)：计算 6 个模型的 $precision$、$recall$、$f1-score$，画出 6 个模型的混淆矩阵热力图、ROC 曲线，最后根据 F1 分数值选出最优模型为 Stacking 模型。

第六步 (提出建议)：通过标签与分类属性的数据分布情况、标签与分类属性的相关性系数、决策树特征重要性分析，给客户提出合理建议。

具体研究思路如下：

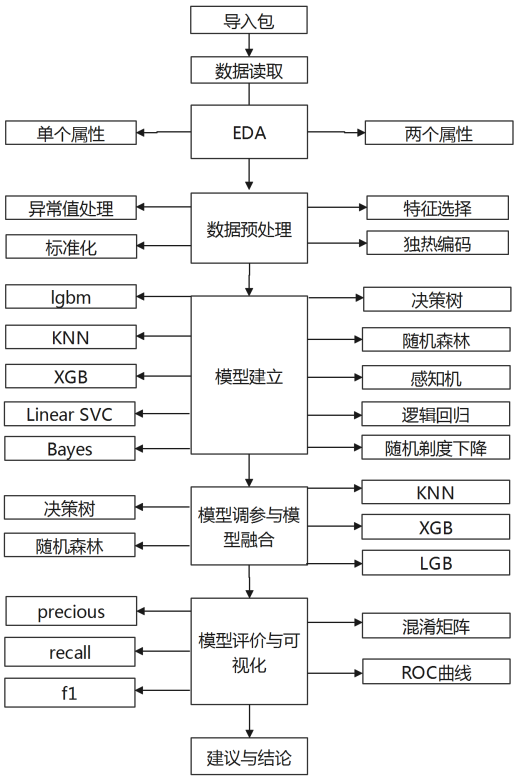

图 1.1 论文研究思路

2 EDA

2.1 导入包

```
#导入常用包
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import random as rnd
%matplotlib inline

#导入机器学习包
from sklearn.linear_model import LogisticRegression    #逻辑回归
from sklearn.svm import SVC, LinearSVC                 #SVC
from sklearn.ensemble import RandomForestClassifier    #随机森林
from sklearn.neighbors import KNeighborsClassifier     #KNN
from sklearn.naive_bayes import GaussianNB             #贝叶斯
from sklearn.linear_model import Perceptron            #感知机
from sklearn.linear_model import SGDClassifier         #SGD
from sklearn.tree import DecisionTreeClassifier        #决策树
```

图 2.1 导入常用的包

首先导入需要的包，其中常用包包括了 numpy，pandas，matplotlib，seabornscikit-learn 的包包括了逻辑回归，SVC，随机森林，KNN，朴素贝叶斯，感知机，SGD 和决策树，本文中用于训练并预测信用卡欺诈行为是否发生。

2.2 数据读取

```
df=pd.read_csv('../input/credit-card-fraud/card_transdata.csv')
```

图 2.2 读取数据集

使用 *pandas* 的函数读取数据集，并命名为 *df*。

2.3 查看数据基本信息

通过查看数据行数列数，数据集前五行，基本信息等，获取到以下基本信息：

- 没有缺失值；
- 数据形状为 (1000000,8), 包含 100 万数据、8 个属性；
- 8 个属性之间的相关性接近于 0，属性的独立性好，具有优良特征；
- 8 个属性中 $distance\_from\_home$、$distance\_from\_last\_transaction$、$ratio\_to\_median\_purcha$ 为连续型数据；$repeat\_retailer$、$used\_chip$、$used\_pin\_number$、$online\_order$ 是取值为 0 或 1 的特征；$fraud$ 是取值为 0 或 1 的标签。

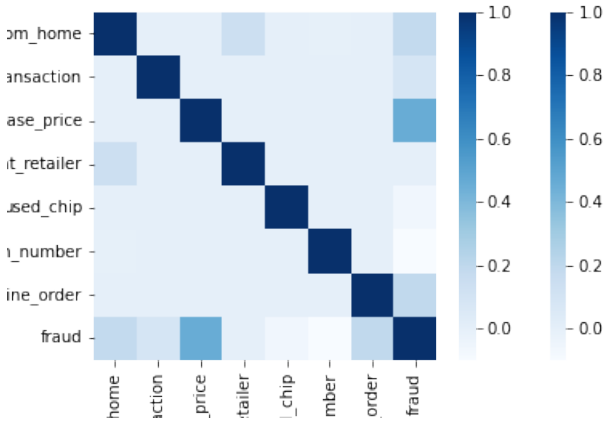

图 2.3 属性相关性系数矩阵热力图

2.4 单个属性数据查看

绘制 7 个特征属性的数据分布图如下：

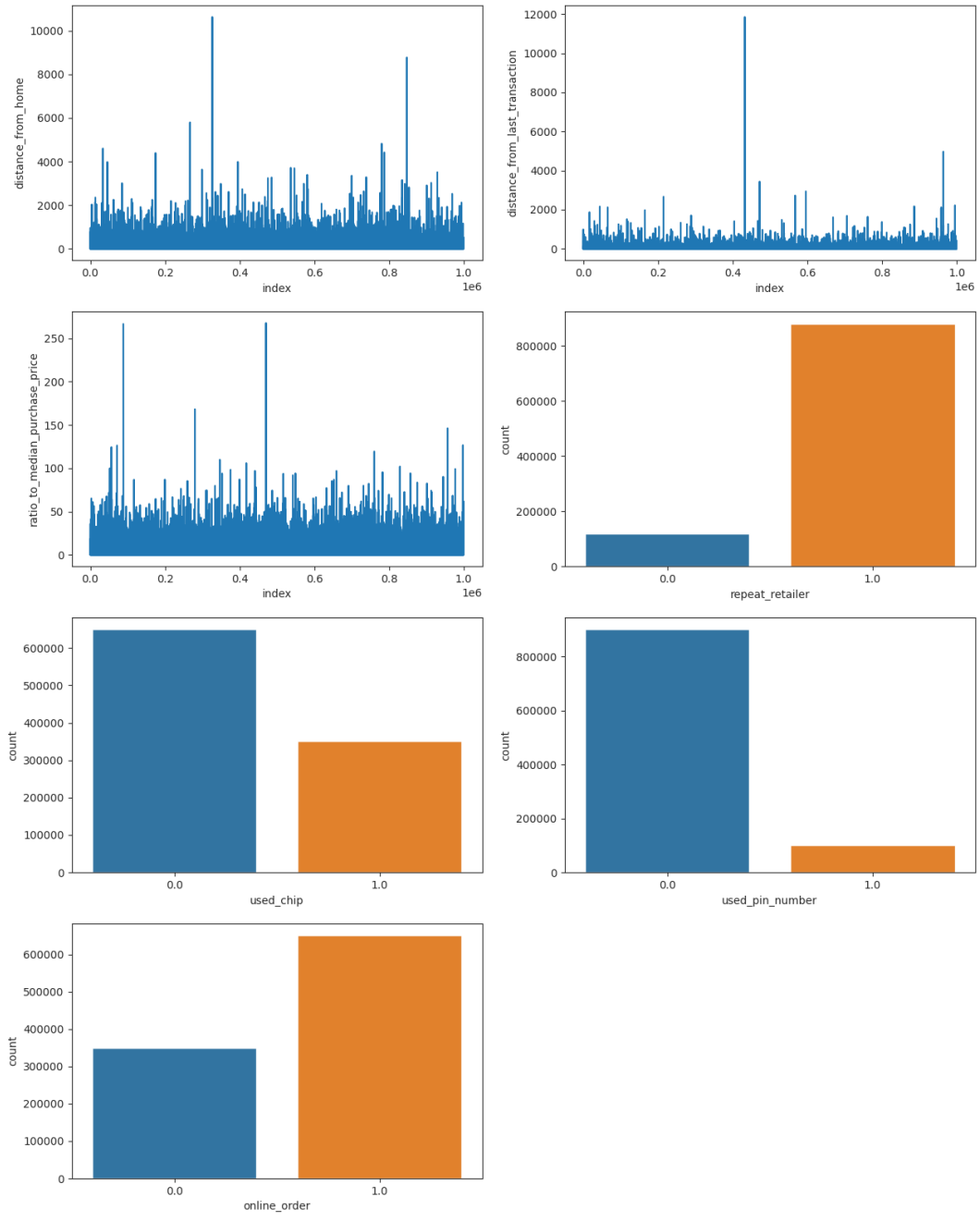

图 2.4 数据分布图

通过上图，我们得到以下结论：

- $distance\_from\_home$ 中只有少部分的值超出 2000，大部分值在 2000 以下；
- $distance\_from\_last\_transaction$ 中极少的值超出 1000，大部分值在 1000 以下；

- *ratio_to_median_purchase_price* 中只有极少超出 100，大部分值在 100 以下；
- 连续型属性只有极少的值发生了严重偏离；推断是出国，境外交易等因素导致；
- *repeat_retailer* 图显示，大部分的交易发生在同一个账户；
- *used_chip* 图显示，用银行卡交易的订单要比不用银行卡交易的订单多一些；
- *used_pin_number* 图显示，大部分的交易不会使用 PIN 码进行交易；
- *online_order* 图显示, 在线交易订单比非在线订单多

2.5 标签关于分类属性的分布

分别绘制标签在 4 个分类属性条件下的分布图如下：

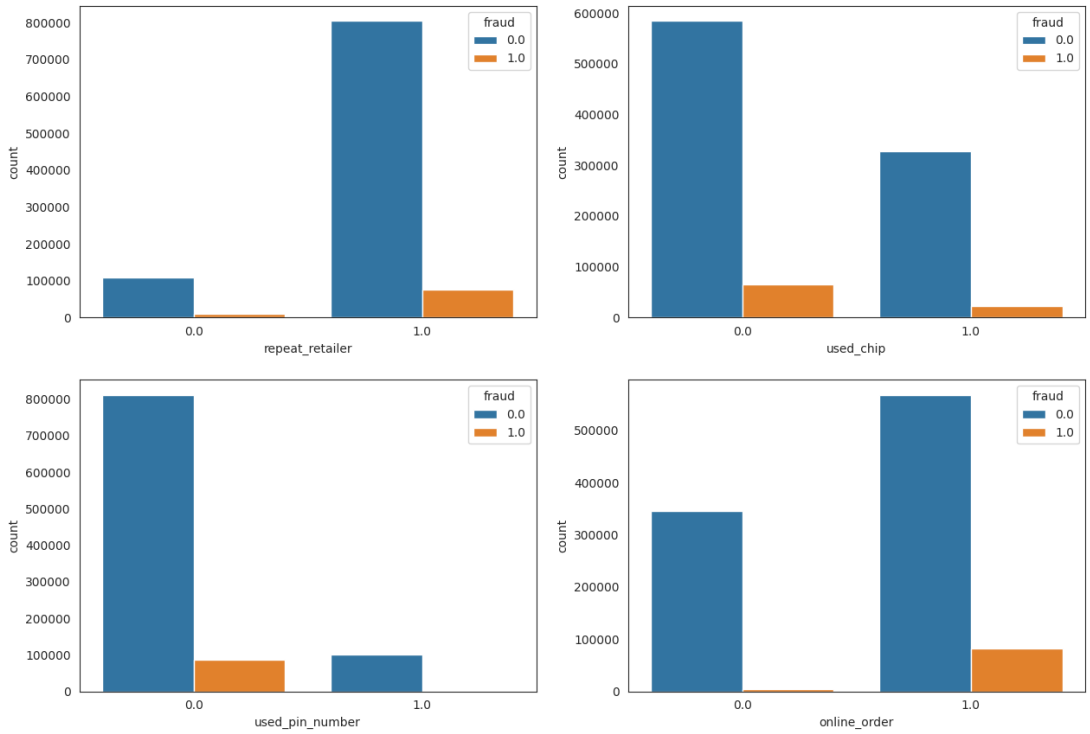

图 2.5 标签关于分类属性的分布

- *fraud* 关于 *repeat_retailer* 数据分布图显示，交易发生在同一个账户与不发生在同一个账户发生欺诈行为与不发生欺诈行为的比例相近，也就是说交易发不发生在同一个账户对是否欺诈的影响不大。

- $fraud$ 关于 $used\_chip$ 数据分布图显示，用银行卡交易与不用银行卡交易发生欺诈行为与不发生欺诈行为的比例相近，也就是说是否用银行卡交易对是否欺诈的影响不大。

- $used\_pin\_number$ 图显示，发生欺诈行为的大都是不用 pin 交易的，用 pin 交易的没有发生欺诈行为。

- $online\_order$ 图显示, 在线交易发生欺诈行为的概率要比非在线交易发生欺诈行为的概率高得多。

3 数据预处理

3.1 异常值处理

根据前面的分析，3 个连续型属性的分布，发现只有极少的值是异常值，考虑到异常数值对应属性在实际生活中是切实存在的，并且异常值处理前后平均值与方差的变化在 5% 以内，因此不对数据进行异常值处理。

3.2 标准化

对三个连续型属性进行标准化。标准化代码和结果如图：

```
from sklearn.preprocessing import StandardScaler
df.iloc[:,0:3]=StandardScaler().fit_transform(df.iloc[:,0:3])
df.head(3)
```

	distance_from_home	distance_from_last_transaction	ratio_to_median_purchase_price	repeat_retailer	used_chip	used_pin_number	online_order	fraud
0	0.477882	-0.182849	0.043491	1.0	1.0	0.0	0.0	0.0
1	-0.241607	-0.188094	-0.189300	1.0	0.0	0.0	0.0	0.0
2	-0.329369	-0.163733	-0.498812	1.0	0.0	0.0	1.0	0.0

图 3.1 标准化

3.3 独热编码

对四个分类属性进行独热编码，代码和结果如图：

```
df.insert (4 ,'repeat_retailer_hot', (df['repeat_retailer']==0).astype('float32')),df.insert (6 ,'used_chip_hot', (df['used_chip']==0).astype('float32'))
df.insert (8 ,'used_pin_number_hot', (df['used_pin_number']==0).astype('float32')),df.insert (10 ,'online_order_hot', (df['online_order']==0).astype('float32'))
df.head(3)
```

	distance_from_home	distance_from_last_transaction	ratio_to_median_purchase_price	repeat_retailer	repeat_retailer_hot	used_chip	used_chip_hot	used_pin_number	used_pin_number_hot	online_order	online_order_hot	fraud
0	0.477882	-0.182849	0.043491	1.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0
1	-0.241607	-0.188094	-0.189300	1.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0
2	-0.329369	-0.163733	-0.498812	1.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0

图 3.2 独热编码

3.4 特征选择

根据图2.3的相关系数矩阵热力图以及相关系数矩阵，发现属性间相关性小，属性独立性好，故没有进一步对数据降维的必要。因此选择所有特征进行分类。

4 模型建立

4.1 数据集划分

按照训练集：测试集为 7:3 的比例对数据集进行划分，代码和结果如图：

```
from sklearn.model_selection import train_test_split
x=df.iloc[:,0:11]
y=df.iloc[:,[11]]
train_x,test_x,train_y,test_y=train_test_split(x,y,test_size=0.3,random_state=420)
[train_x.shape,train_y.shape,test_x.shape,test_y.shape]
```

```
[(700000, 11), (700000, 1), (300000, 11), (300000, 1)]
```

图 4.1 数据集划分

4.2 建立模型并选择调参模型

使用 LGB,XGB,KNN, 线性 SVC，朴素贝叶斯，决策树，随机森林，感知机，逻辑回归，随机剃度下降 10 个模型对数据进行拟合并查看在测试集上的准确率如图：

	Model	Score
2	Random Forest	100.000000
7	Decision Tree	100.000000
0	KNN	99.850000
8	LGB	99.797333
9	XGB	99.797333
4	Perceptron	96.050000
5	Stochastic Gradient Decent	95.970000
1	Logistic Regression	95.830000
6	Linear SVC	94.710000
3	Naive Bayes	64.920000

图 4.2 模型得分

其中决策树，随机森林的准确率达到了 **100%**，LGB,XGB,KNN 均有较高的准确率。

5 模型调参与模型融合

5.1 模型调参

选用 LGB,XGB,KNN，决策树，随机森林评分最高的 5 个模型进行调参，使用 5 折交叉验证，使用网格搜索对参数进行优化，其中随机森林的调参代码和结果如下：

```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier(random_state=0)
tuned_parameters = [{ 'n_estimators':range(25,75,1),"criterion":["gini","entropy"]}]
clf = GridSearchCV(estimator=rfc,param_grid=tuned_parameters, cv=5, n_jobs=-1)
clf.fit(x, y)
print('Best parameters:')
print(clf.best_params_)
print(clf.best_score_)

# Best parameters:
criterion_best='gini'
n_estimators_best= 35
scores_best=0.999989
```

图 5.1 调参代码

选取得分最高的参数作为最终模型，参数调节前后模型的变化如下图：

模型	调节前五折交叉得分	调节后五折交叉得分	最优参数1	最优参数2
随机森林	0.999968	0.999999	criterion='gini'	n_estimators= 35
决策树	0.999970	0.999990	max_depth_best= 7	
KNN	0.998609	0.998834	n_neighbors = 3	
XGB	0.999912	0.999986	max_depth=3	n_estimatores=0
LGB	0.998096	0.999987	max_depth=5	n_estimatores= 0

图 5.2 调参结果

5.2 模型融合

选取调参后表现最高的 LGB,XGB, 随机森林搭建 Stacking 模型，代码如下：

```
from sklearn.ensemble import StackingClassifier
sta = StackingClassifier(estimators=[('rfc', rfc),('xgb',xgb ),('lgbm', lgbm)])
sta.fit(train_x, train_y)
y_pred_sta = sta.predict(test_x)
```

图 5.3 Stacking 模型

6 模型评价与可视化

根据上面的 6 个模型，求出模型 5 折交叉验证准确率，precious，recall，F-1 分数值如下：

模型	五折交叉得分	precious	recall	F-1
随机森林	0.999999	1	0.999809	0.999905
决策树	0.999990	0.999923	0.999771	0.999847
KNN	0.998834	0.994485	0.991221	0.992851
XGB	0.999986	1	0.999885	0.999943
LGB	0.999987	1	0.999771	0.999885
Stacking		1	0.999771	0.999885

图 6.1 模型评价指标

分别绘制 6 个模型在测试集上的混淆矩阵如图：

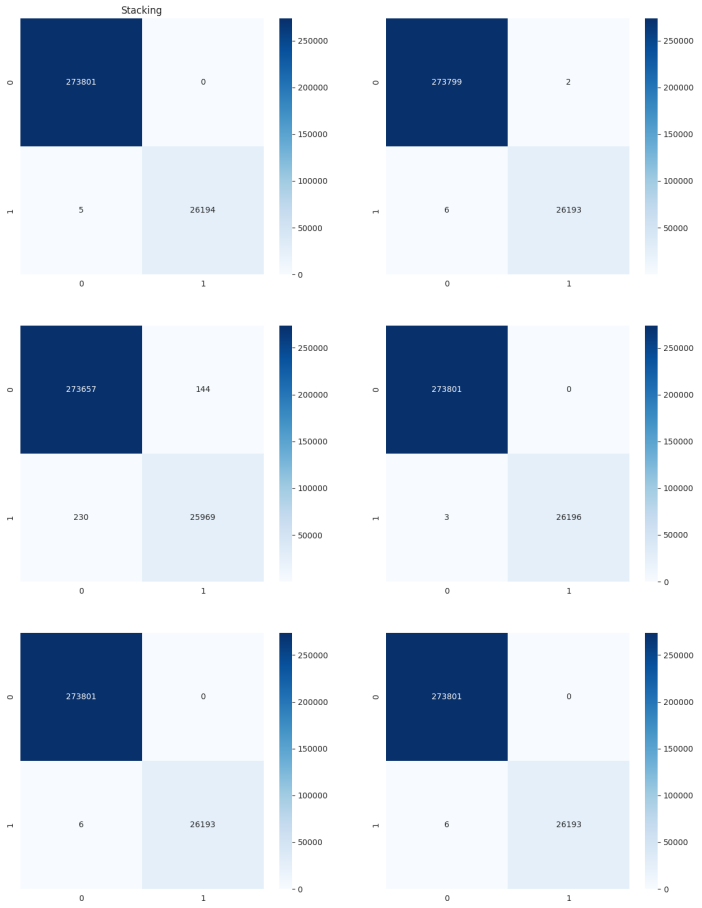

图 6.2 混淆矩阵

通过绘制 6 个图的 ROC 曲线发现，LGB,XGB,KNN，决策树，随机森林，Stacking 的模型的 ROC 曲线均相和下图相同：

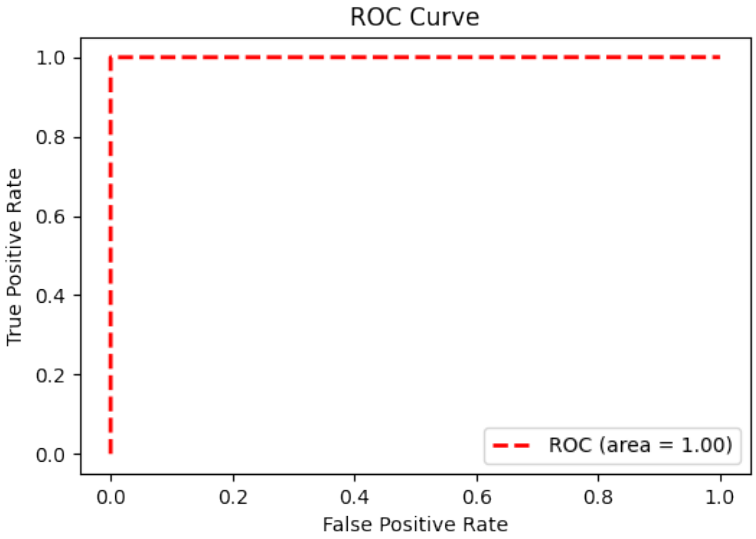

图 6.3 ROC 曲线

从以上的模型可以看出，模型之间的差别已经不大，根据五折交叉得分和 F1 分数值最终选择 Stackin 作为最终模型，模型准确率为 0.9999987，F-1 值为 0.999885。

7 建议与结论

7.1 建议

根据 fraud 与各个属性的相关性系数以及结合本文 EDA 部分的结论，我们提出一下建议：用户如果能够使用 pin 交易，那么将会使欺诈发生的概率趋近于 0；当用户发现当前交易是往期交易的数倍时，用户需要谨慎思考当前是否遭遇了银行卡欺诈行为。

7.2 结论

最终选择 Stackin 作为最终模型，模型准确率为 0.9999987，F-1 值为 0.999885。

参考文献

[1] 周志华. 机器学习 [J]. 清华大学出版社, 2016, 8(28): 1–415.