信用卡欺诈分析

三个目标：

1、信用卡欺诈属于二分类问题，即使用 sklearn 中的逻辑回归分类；

2、欺诈交易在所有交易中的比例很小，对于这种数据不平衡的情况，采用什么样的模型评估标准会更准确；

3、对信用卡欺诈分析的项目，进行数据可视化与模型效果评估。

一、构建 sklearn 中的逻辑回归分类

sklearn 里的 LogisticRegression() 函数构建逻辑回归分类器，常用构造参数如下：

1、penalty：惩罚项，取值为 l1 或 l2，默认为 l2。当模型参数满足高斯分布的时候，使用 l2，当模型参数满足拉普拉斯分布的时候，使用 l1；

2、solver：代表的是逻辑回归损失函数的优化方法。有 5 个参数可选，分别为 liblinear、lbfgs、newton-cg、sag 和 saga。默认为 liblinear，适用于数据量小的数据集，当数据量大的时候可以选用 sag 或 saga 方法。

3、max_iter：算法收敛的最大迭代次数，默认为 10。

4、n_jobs：拟合和预测的时候 CPU 的核数，默认是 1，也可以是整数，如果是 -1 则代表 CPU 的核数。

创建好后，使用 fit 函数拟合，使用 predict 函数预测。

二、模型评估指标

通常使用准确率 (accuracy)来评估模型，它指的是分类器正确分类的样本数与总体样本数之间的比例。这个指标对大部分的分类情况是有效的，不过当分类结果严重不平衡的时候，准确率很难反应模型的好坏。

此时，**应当更加注重特殊类别的识别。**

数据预测的四种情况：TP、FP、TN、FN。

TP：预测为正，判断正确；

FP：预测为正，判断错误；

TN：预测为负，判断正确；

FN：预测为负，判断错误。

样本总数 =TP+FP+TN+FN

预测正确的样本数为 TP+TN

即 准确率 Accuracy = (TP+TN)/(TP+TN+FN+FP)

精确率 P = TP/ (TP+FP)，在所有判断为真的个数中，真正为真的比例。

召回率 R = TP/ (TP+FN)，也称为查全率，被正确识别出来的个数与真的总数的比例。

F1指标综合了精确率和召回率，可以更好地评估模型的好坏。有：

F1 = 2 x P x R / (P + R)

$$F1 = 2 \times \frac{P \times R}{P + R}$$

F1 作为精确率 P 和召回率 R 的调和平均，数值越大代表模型的结果越好。

数据集说明：

包括了 2013 年 9 月份两天时间内的信用卡交易数据，284807 笔交易中，一共有 492 笔是欺诈行为。输入数据一共包括了 28 个特征 V1，V2，……V28 对应的取值，以及交易时间 Time 和交易金额 Amount。为了保护数据隐私，我们不知道 V1 到 V28 这些特征代表的具体含义，只知道这 28 个特征值是通过 PCA 变换得到的结果。另外字段 Class 代表该笔交易的分类，Class=0 为正常（非欺诈），Class=1 代表欺诈。我们的目标是针对这个数据集构建一个信用卡欺诈分析的分类器，采用的是逻辑回归。从数据中你能看到欺诈行为只占到了 492/284807=0.172%，数据分类结果的分布是非常不平衡的，因此我们不能使用准确率评估模型的好坏，而是需要统计 F1 值（综合精确率和召回率）。

项目流程分析如下：

1、载入数据；

2、探索数据，划分数据集，测试集与训练集；

3、V1-V28 的特征值都经过 PCA 的变换，但是其余的两个字段，Time 和 Amount 还需要进行规范化。Time 字段和交易本身是否为欺诈交易无关，因此我们不作为特征选择，只需要对 Amount 做数据规范化；
4、创建逻辑回归分类器，然后传入训练集数据进行训练，并传入测试集预测结果，将预测结果与测试集的结果进行比对。这里的模型评估指标用到了精确率、召回率和 F1 值。同时我们将精确率 - 召回率进行了可视化呈现。

定义了 plot_confusion_matrix 函数对混淆矩阵进行可视化。
使用了 sklearn 中的 precision_recall_curve 函数，通过预测值和真实值来计算精确率 - 召回率曲线。precision_recall_curve 函数会计算在不同概率阈值情况下的精确率和召回率。最后定义 plot_precision_recall 函数，绘制曲线。