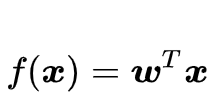
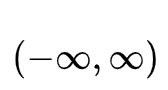
算法选择思路：

首先，我们拿到的数据是用户2天内的信用卡交易数据，这份数据包含很多维度，要解决的问题是预测用户是否会发生信用卡被盗刷。信用卡持卡人是否会发生被盗刷只有两种可能，发生被盗刷或不发生被盗刷。又因为这份数据是打标好的（字段Class是目标列），也就是说它是一个监督学习的场景。于是，我们判定信用卡用户是否会发生被盗刷是一个二元分类问题，意味着可以通过二分类相关的算法来找到具体的解决办法，本项目选用的算法是逻辑斯蒂回归（Logistic Regression）。

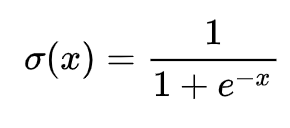
逻辑回顾算法数学背景

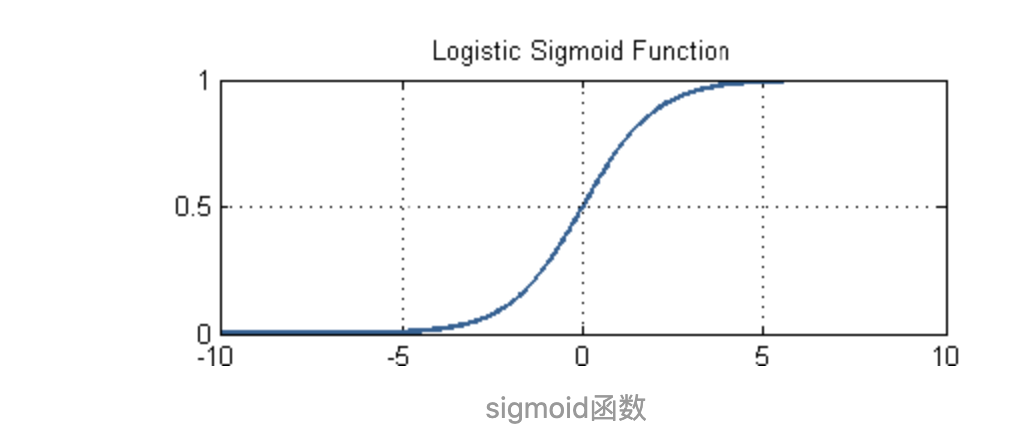
明确了预测目标是标签为A的概率。

我们知道，概率是属于[0,1]区间。但是线性模型  值域是  。

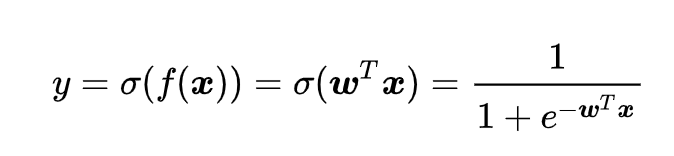
我们不能直接基于线性模型建模。需要找到一个模型的值域刚好在[0,1]区间，同时要足够好用。

于是，选择了我们的sigmoid函数。

它的表达式为：



我们结合sigmoid函数，线性回归函数，把线性回归模型的输出作为sigmoid函数的输入。于是最后就变成了逻辑回归模型：



假设我们已经训练好了一组权值  。只要把我们需要预测的  x 代入到上面的方程，输出的y值就是这个标签为A的概率，我们就能够判断输入数据是属于哪个类别。

分析数据：数据是结构化数据 ，不需要做特征抽象。特征V1至V28是经过PCA处理，而特征Time和Amount的数据规格与其他特征差别较大，需要对其做特征缩放，将特征缩放至同一个规格。在数据质量方面 ，没有出现乱码或空字符的数据，可以确定字段Class为目标列，其他列为特征列。

这份数据是全部打标好的数据，可以通过交叉验证的方法对训练集生成的模型进行评估。70%的数据进行训练，30%的数据进行预测和评估。 现对该业务场景进行总结如下：

根据历史记录数据学习并对信用卡持卡人是否会发生被盗刷进行预测，二分类监督学习场景，选择逻辑斯蒂回归（Logistic Regression）算法。

数据为结构化数据，不需要做特征抽象，但需要做特征缩放。

逻辑回归主要代码：

1. 生成混淆矩阵

from sklearn.linear\_model import LogisticRegression

lg\_clf=LogisticRegression()

lg\_clf.fit(X\_train\_new,Y\_train\_new)

lg\_pred=lg\_clf.predict(X\_test)

cnf\_matrix\_lg = confusion\_matrix(Y\_test,lg\_pred) # 生成混淆矩阵

np.set\_printoptions(precision=2)#精确到两位小数点

2. 利用逻辑回归算法分类

from sklearn.linear\_model import LogisticRegression

para\_logit= {'C': [100,1,10]}#候选参数集

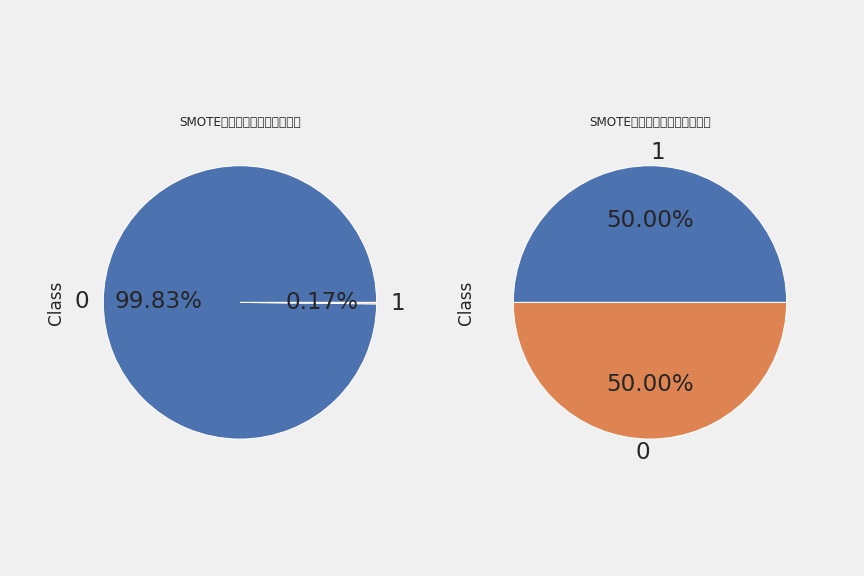
clf=GridSearchCV(LogisticRegression(dual=False),param\_grid=para\_logit,cv=10,iid=False,n\_jobs=-1)#构建分类器，10折交叉验证

clf.fit(X\_train\_new,Y\_train\_new)#使用训练集进行训练

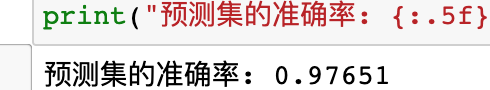
2.5 优化模型

过采样（oversampling），增加正样本使得正、负样本数目接近，然后再进行学习

过采样之前被盗刷所占的比例0.17%

过采样之后被盗刷所占的比例50.00%

3. 训练的准确率和召回率结果：





4. api接口

