使用 Python 实现对数几率回归模型

一 问题描述

对 Iris 数据集实现对数几率回归模型，先对 Iris 数据集进行分类，通过不同比例的训练集、测试集结果来验证模型的效能。

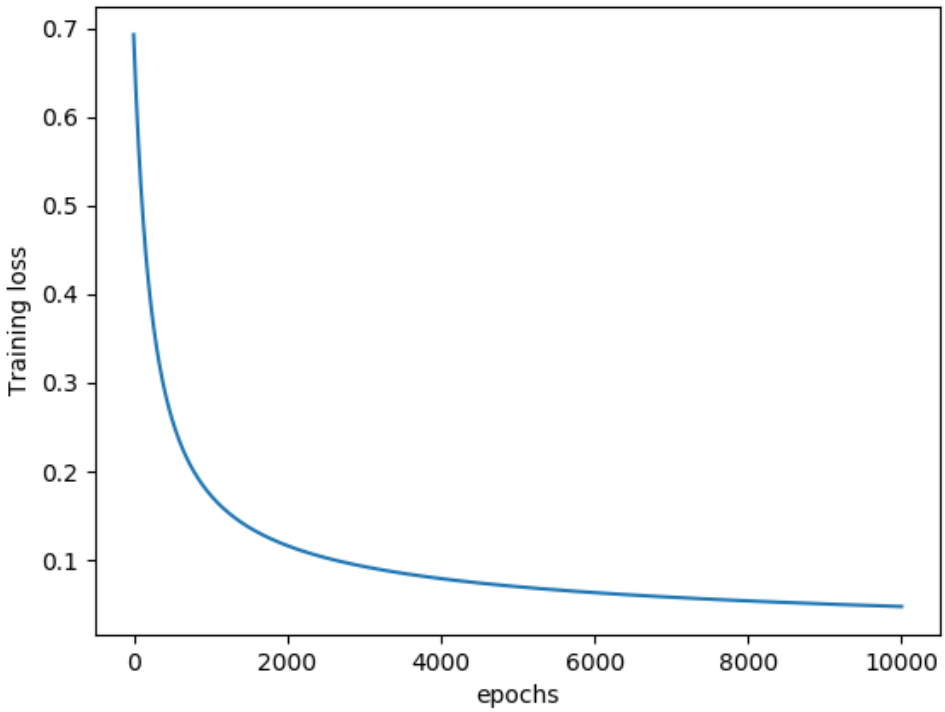
二 数据集描述

利用 sklearn 中 datasets 包对 Iris 数据集进行调用，Iris 数据集中包含 4 种属性以及 3 种品种，为对数据进行可视化呈现，直接选取了第一、二列属性展示，三类数据选择 Setosa 与 Versicolor 两类。

三 实验结果

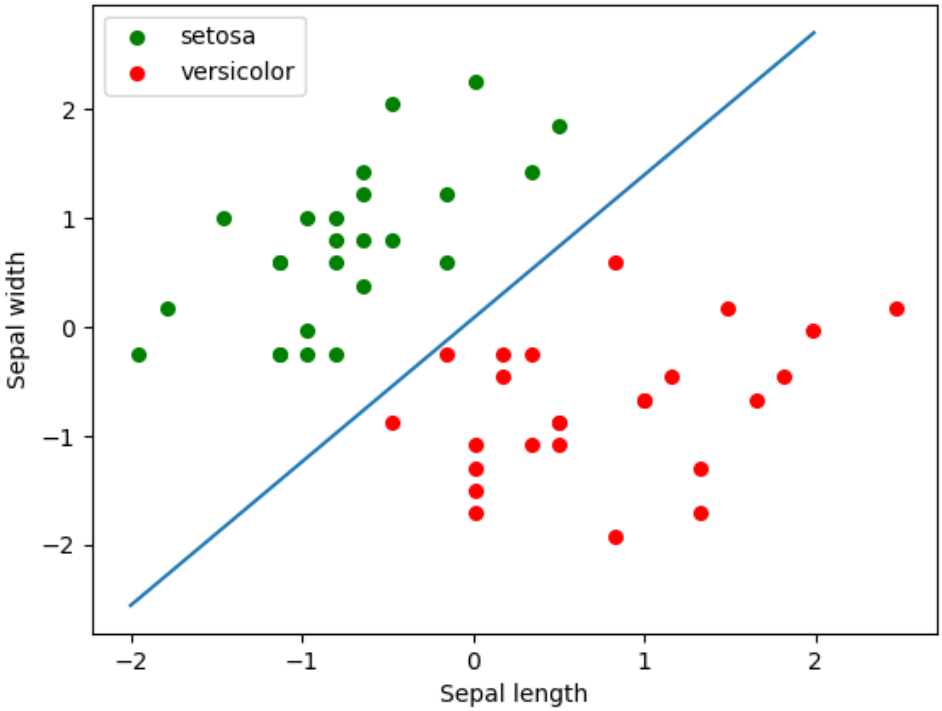
3 . 1 将数据集的 50% 作为训练集，50% 作为测试集

图 **1**　模型损失函数的变化曲线图



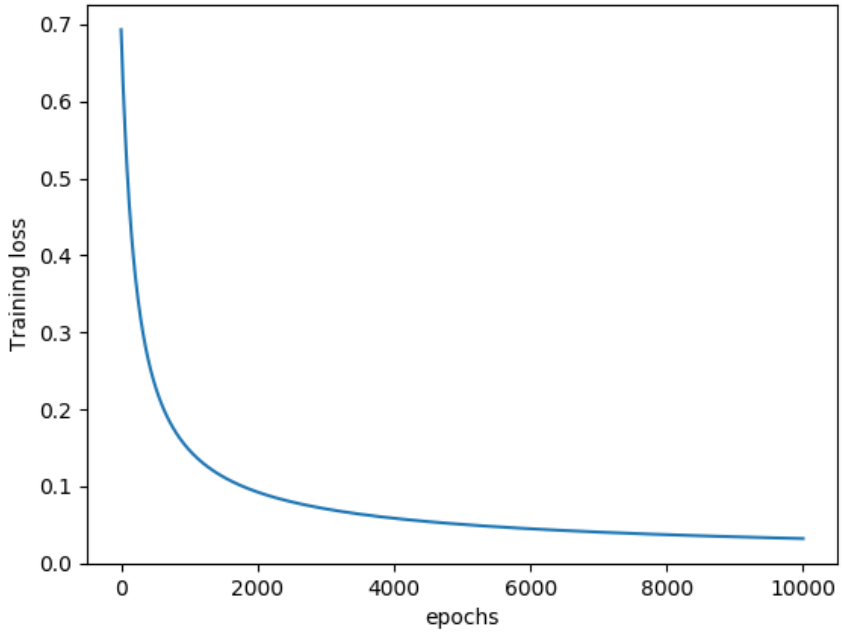
准确度：1.0

图 **2**　测试集数据的可视化与决策边界



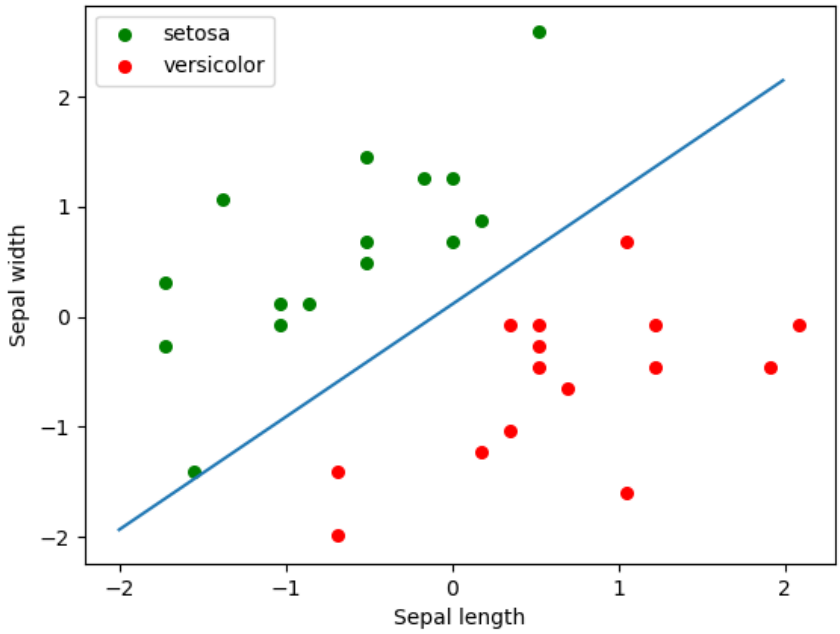
3.2 将数据集的 70% 作为训练集，30% 作为测试集

图 **3**　模型损失函数的变化曲线图



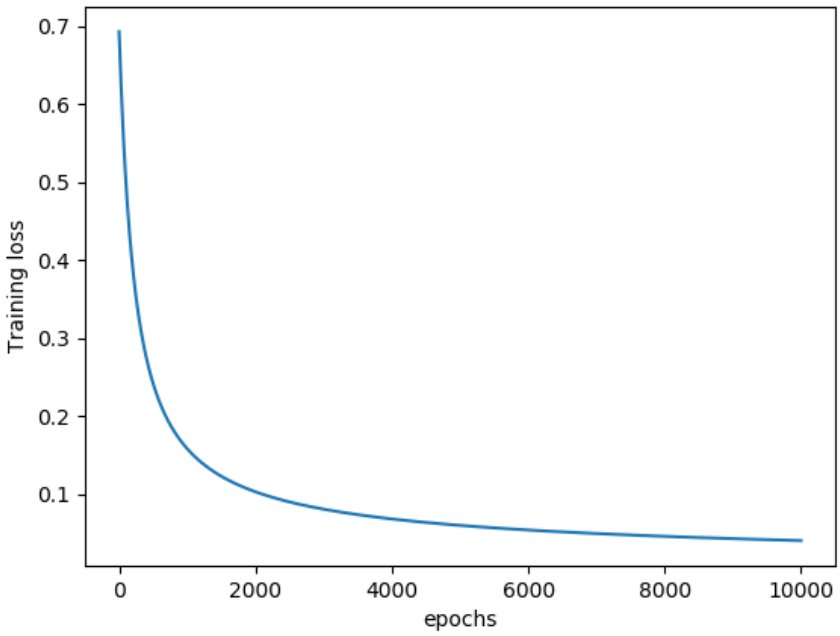
准确度：1.0

图 **4**　测试集数据的可视化与决策边界



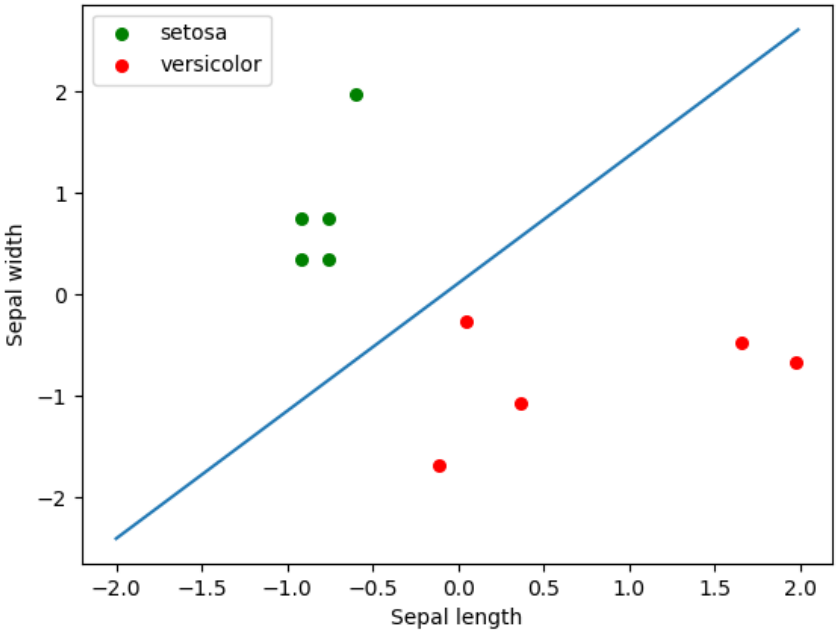
3 . 3 将数据集的 90% 作为训练集，10% 作为测试集

图 **5**　模型损失函数的变化曲线图



准确度：1.0

图 **6**　测试集数据的可视化与决策边界

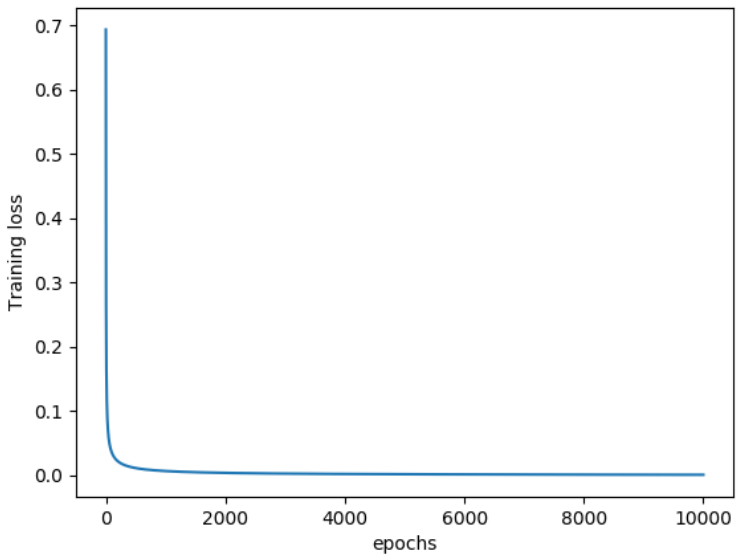


四 实验结果分析

4.1 损失函数变化曲线分析

在梯度下降的训练步长中，刚开始将步长设置为 0.01，发现损失函数曲线下降速度过快，因此对差值取平均来调整步长。

图 **7**　步长取 **0.01** 时损失函数曲线



取了平均后后见结果中的损失函数曲线可以发现，损失函数越来越低，说明迭代次数的不断增加使准确率越来越高，10000 次时接近最低。

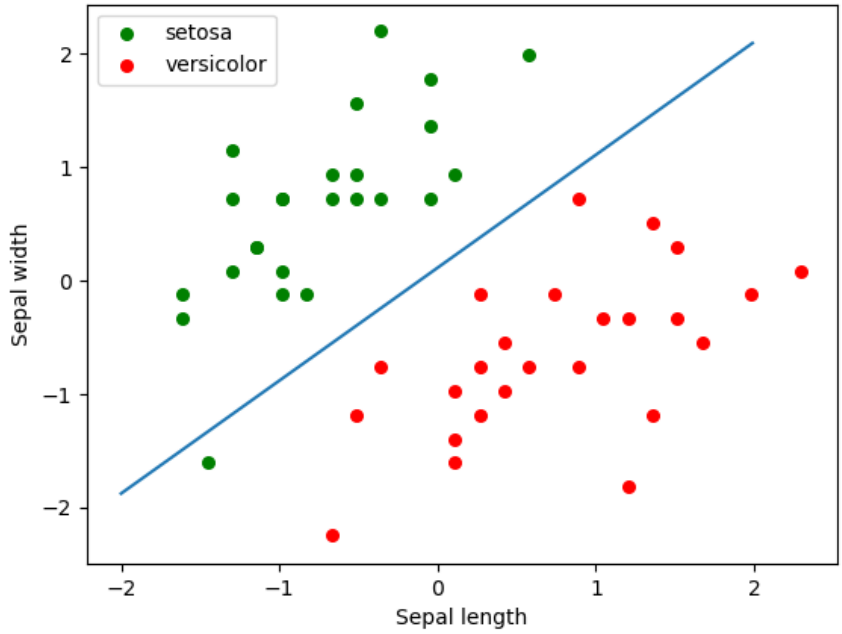
4.2 对预测结果准确度率的分析

在三种情况中，随机划分训练集与测试集，因此每运行一次，得到的对数几率模型不一样，预测结果以及结果的准确度也会发生变化。

经过多次运行可以发现，分类正确率变化并没有太大差距，符合预料，可以认为模型的效能很高。

例如在将数据集的 50% 作为训练集，50% 作为测试集的情况中：

图 **8**　测试集数据的可视化与决策边界



此时正确率为 0.98，也是符合预期范围的。同样对另两种情况也都进行了多次运行，结果均符合预期范围。