重庆大学大数据与软件学院

上 机 报 告

|  |  |
| --- | --- |
| 上机名称 | 线性回归和对数几率回归 |
| 课程名称 | 机器学习 |

**开课实验室：DS-1501 2024 年 9月 24 日**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 姓 名 | | 張和俊 | 学号 | 20220664 | 成 绩 |  |
| 上机（项目）名称 | | | 线性回归和对数几率回归 | | 指导教师 | 王悦阳 |
| 教师评语 | 教师签名：王悦阳  年 月 日 | | | | | |

一、 实验目的

实现线性回归模型和对数几率回归模型

二、实验内容

1、（用python）实现线性回归算法

（1）导入数据集（Olympics）

（2）利用公式 (1) 和 (2) 求出直线方程

（3）可视化（画出原始数据点云、直线）

（4）评价两种方法的优劣（运行时间、目标函数等）

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

2、用 LogisticRegression 模型做二分类任务

（1）导入数据集并划分训练集/测试集（乳腺癌数据集，sklearn库自带）

（2）构建对数几率回归模型（调库）

（3）计算评估指标：AUC（调库）

（4）可视化（ROC曲线，参考Experimental\_Results中的ROC）

3、用 LogisticRegression 模型做多分类任务

（1）导入数据集并划分训练集/测试集（鸢尾花数据集，sklearn库自带）

（2）构建对数几率回归模型（调库）

（3）计算评估指标：accuracy, precision, recall, F1-score（调库）

（4）可视化（决策边界，参考Experimental\_Results中的LogisticRegression）

4、在二分类和多分类任务中手动计算评估指标（选做）

二分类：AUC

多分类：accuracy, precision, recall, F1-score

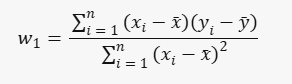
开发环境、数据集与代码参考内部资料

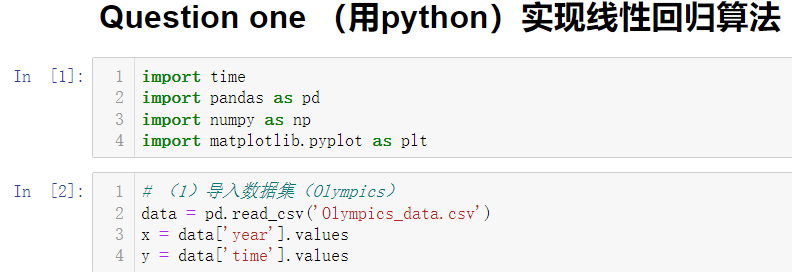
三、实验结果、实验步骤及实验过程原始记录(有截图和简要说明)

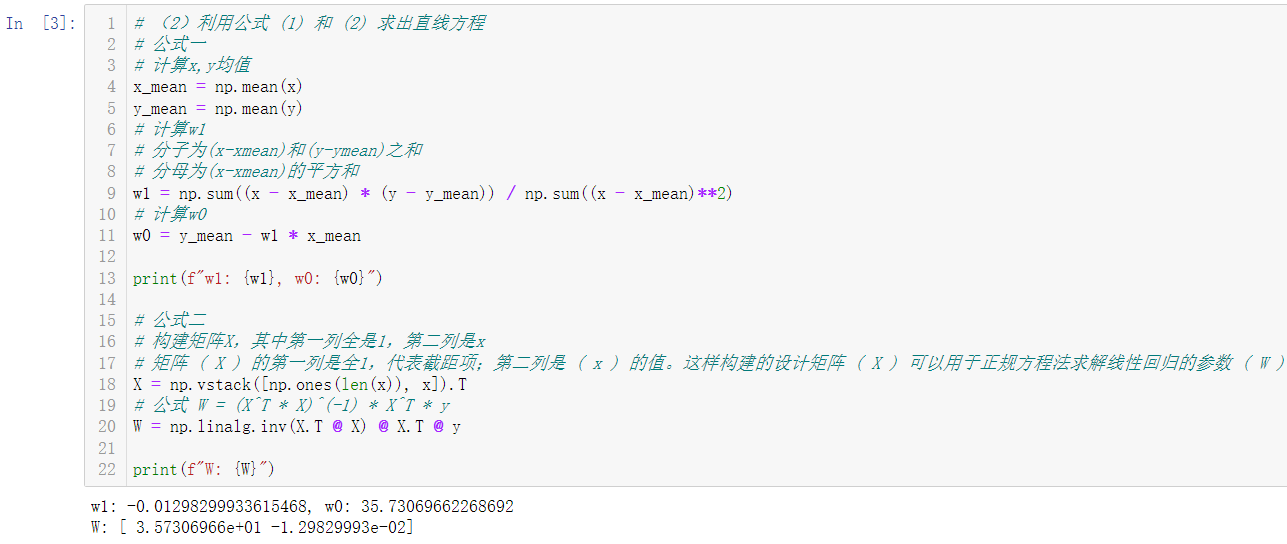
**这里的说明大多数都在打代码的时候写在注释那里了，只有一些有趣的部分我在这里用文字说明。**

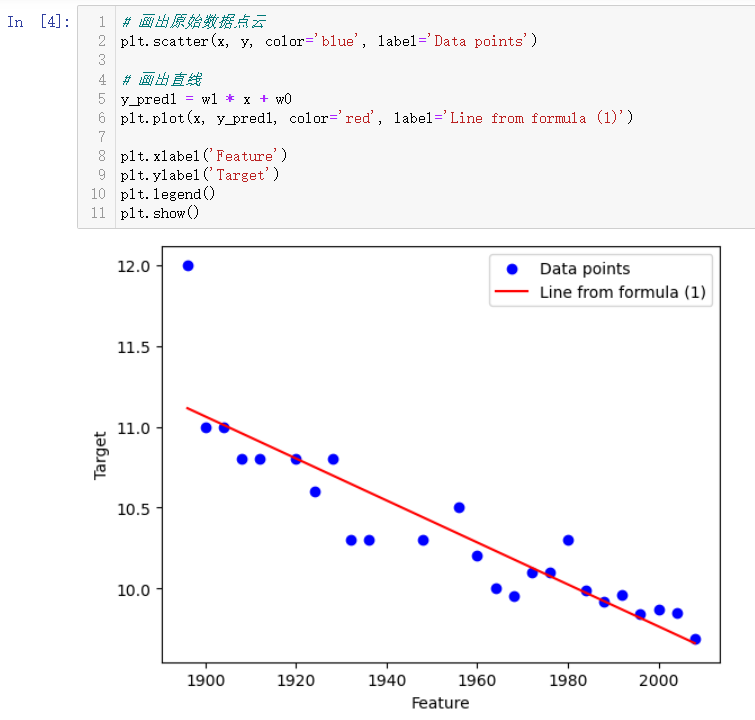
我用的是jupyter notebook，GPU 3050TI

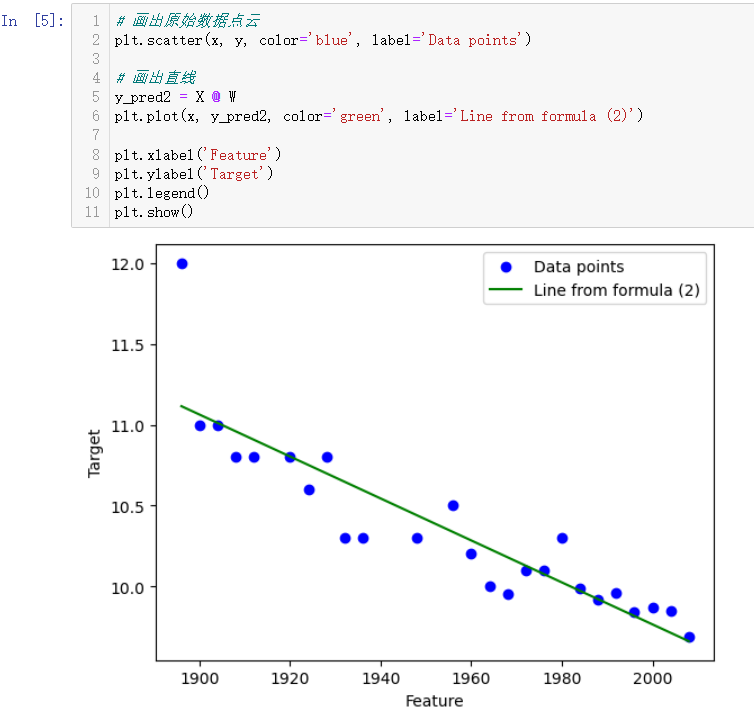
第一个实验的公式一。我用的是这个公式，不是最简的公式的形式，这样子是因为比较好打出代码



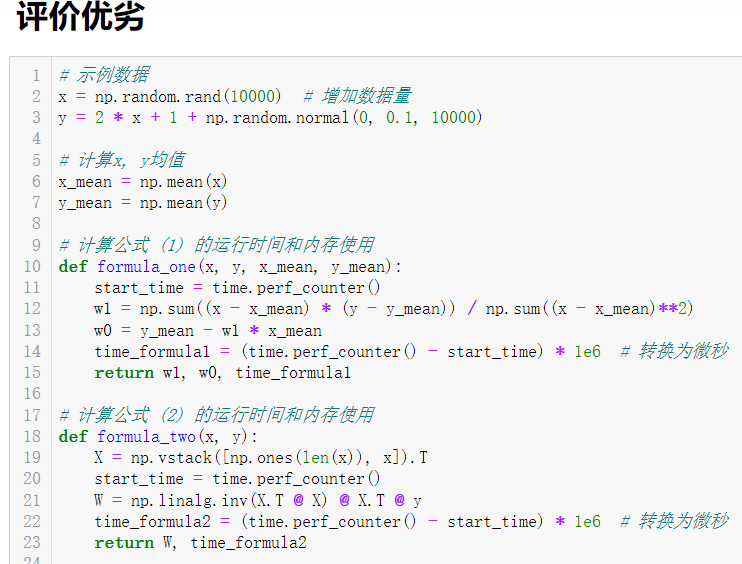


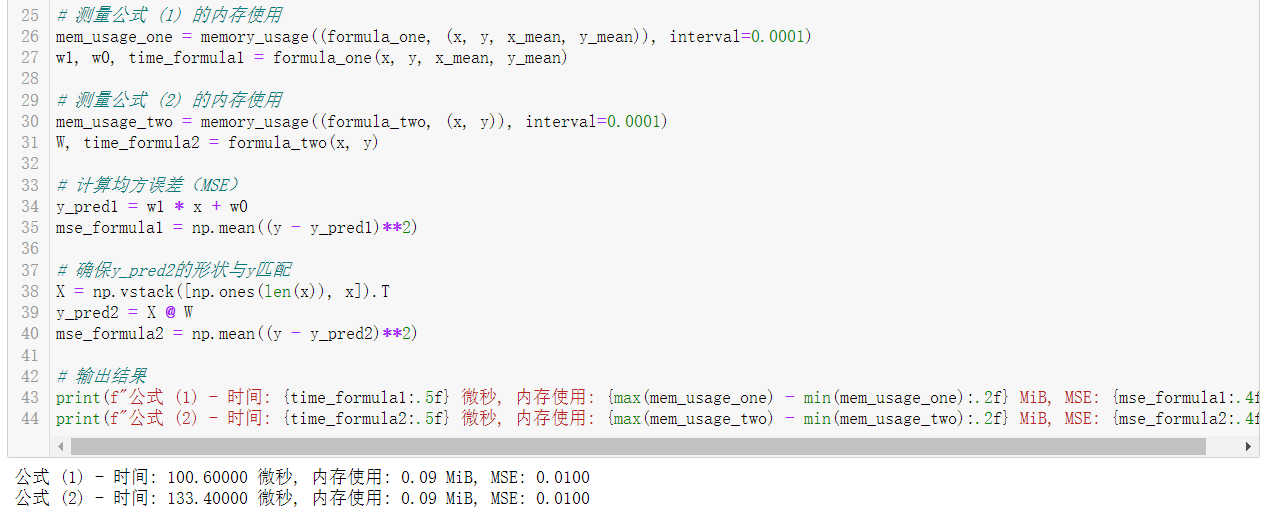






接下来是对两个不同方法的一个评估





这个我是随机生成的数据，因为给的那个数据比较小，我想扩展一下。从结果中我发现到，在小样本的时候，比如样本数量在1000到100000数量级的时候，矩阵形式会造成更大的一个运算复杂度，而公式一这种一般的形式的运算复杂度则较低。

然而我将样本的数量增加到1亿个的时候，会有一个截然不同的结果。



这时候矩阵运算的时间效率会更高，时间复杂度更低，而内存使用中矩阵形式的公式的内存使用随着样本数量的增加显著提升。

更具这个结果我做了一些分析：

公式一（基于均值和方差的计算）的时间复杂度是，所以少样本的情况下，在python的矩阵优化没有显著提供给矩阵公式的优化下，这个方式的时间复杂度最佳。

公式一只需要存储少量的中间结果，因此内存使用较低。

矩阵运算的计算复杂度为，但python这个东西对矩阵运算优化很厉害（比如我用的numpy），他们的现代线性代数库可以在多核CPU和GPU上高效地执行矩阵运算，从而显著提高计算速度。

而内存方面：阵运算需要更多的内存来存储设计矩阵和中间结果，固然需要用多一点内存。



