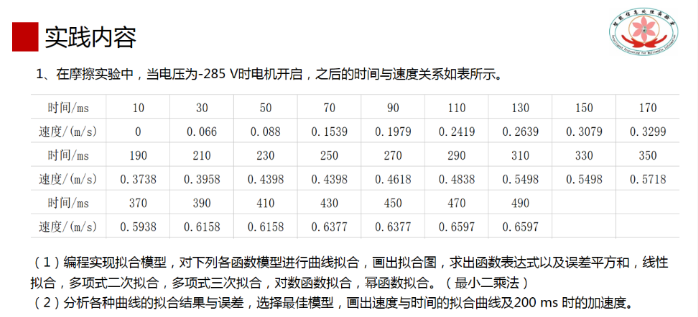
 

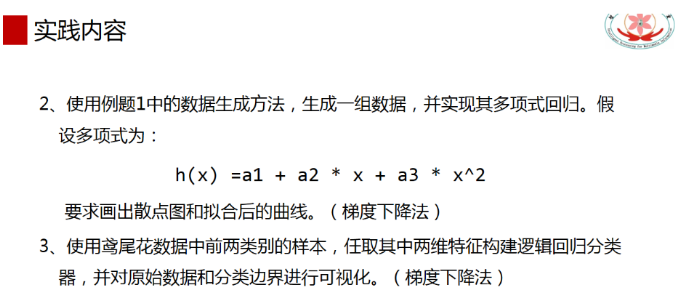
**智能信息系统综合实践**

**实验报告**

|  |  |
| --- | --- |
| **题 目：** | **回归分析** |
| **年 级：** | **2020级** |
| **专 业：** | **软件工程** |
| **学 号：** | **2020118035** |
| **姓 名：** | **王凯** |

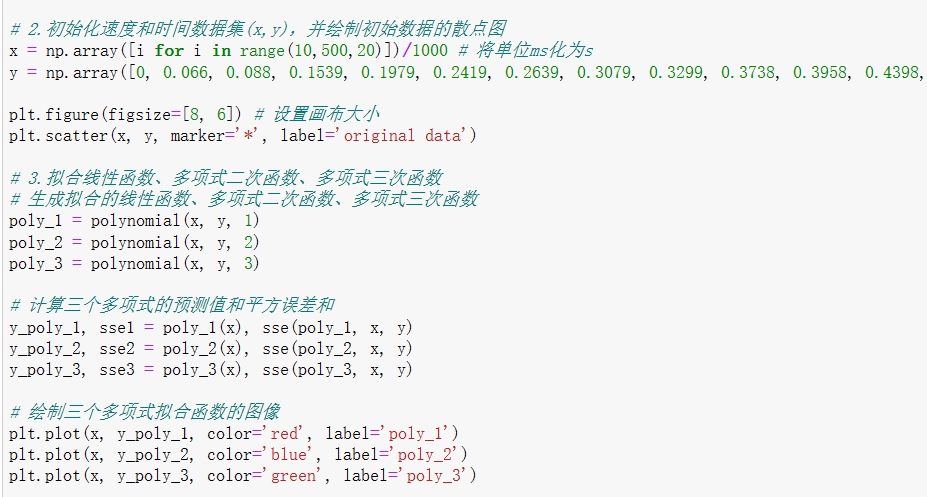
1. **题目**



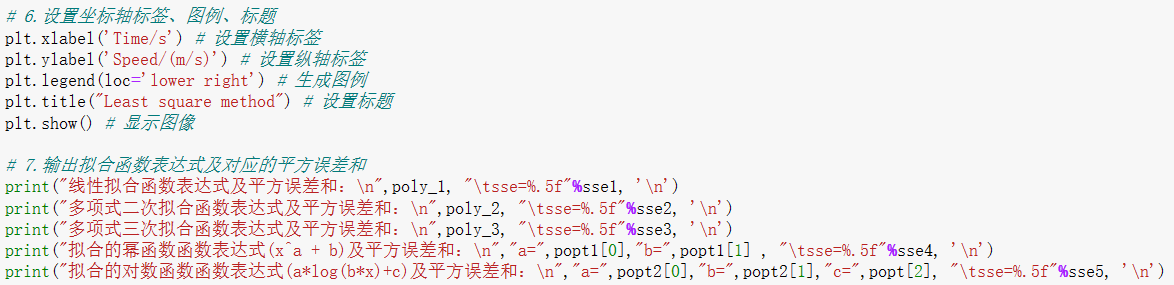


1. **解题步骤（代码中注释即为思路）**
2. 利用最小二乘法拟合函数
   1. 五个拟合函数的图像，
      1. 思路：
         1. 生成样本集(x,y)（注意：表中时间单位为ms，生成数据集时化成s，统一单位）
         2. 使用np.polyfit(x,y,deg)，设置最高项次数deg对样本集(x,y)进行多项式拟合
         3. 使用curve\_fit(fund, x, y)来对幂函数和对数函数进行拟合
         4. 使用拟合后的函数对样本集进行预测，生成(x,h)
         5. 利用sum((h-y)^2)计算各拟合函数对样本的预测值与真实值的误差平方和
         6. 绘制样本集(x,y)的散点图和五种拟合函数的(x,h)曲线图，观察分类效果
         7. 输出各分类函数对样本集的平方误差和，比较各个模型的性能。
      2. 代码实现

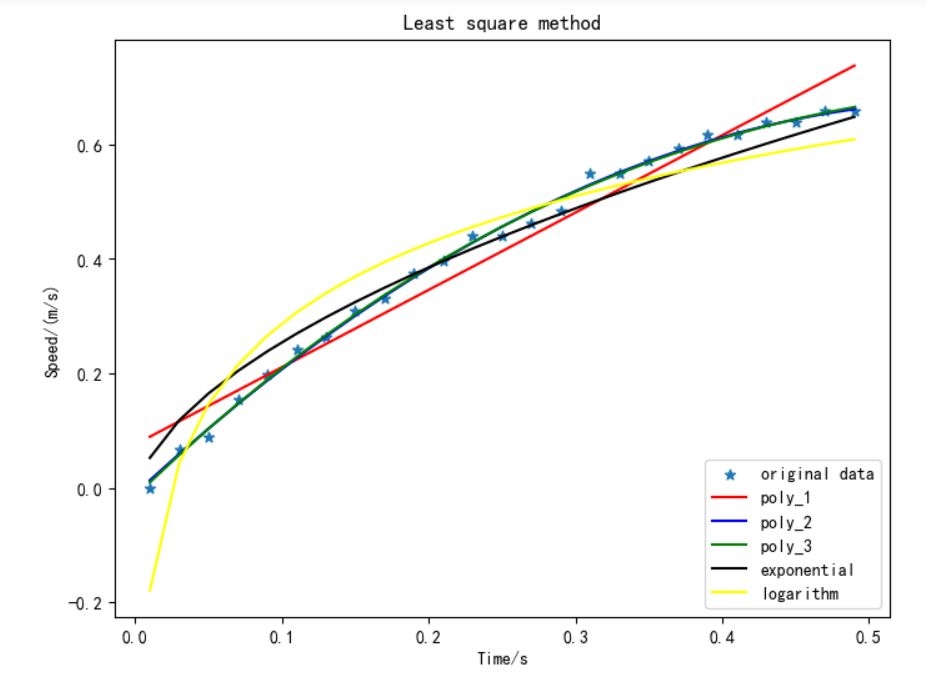


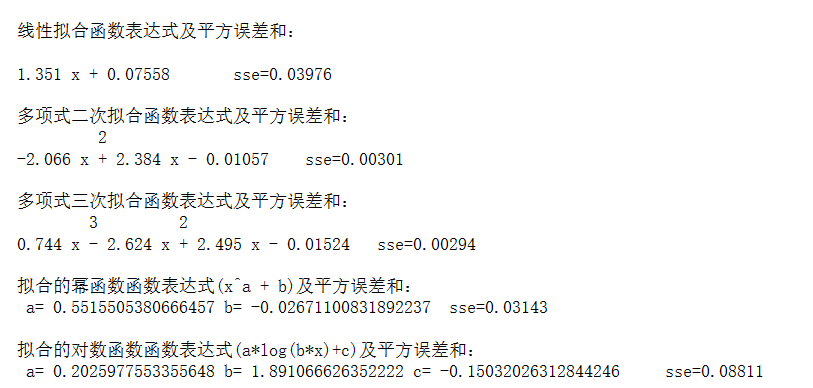




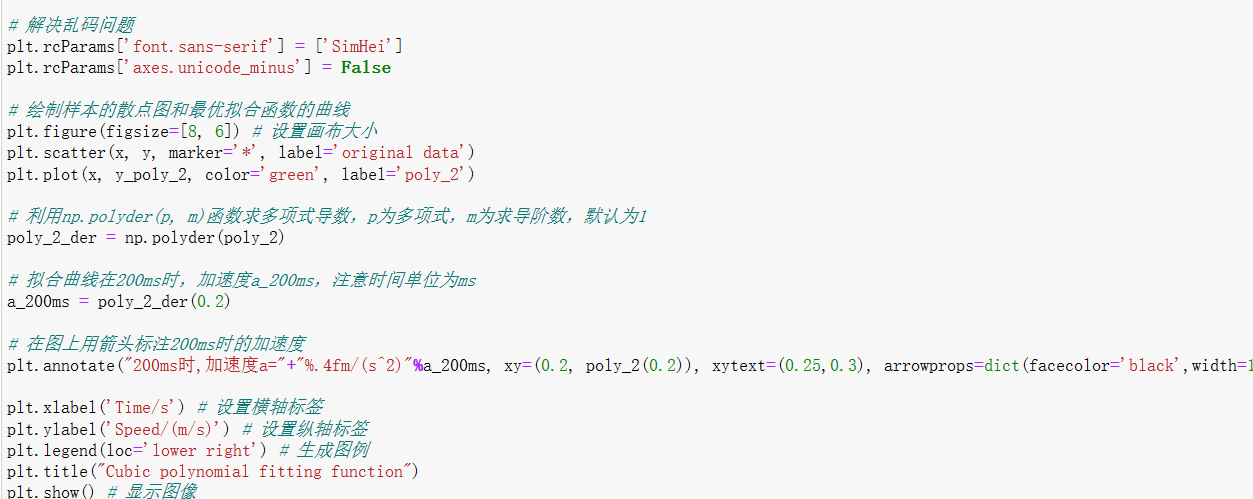


* + 1. 结果

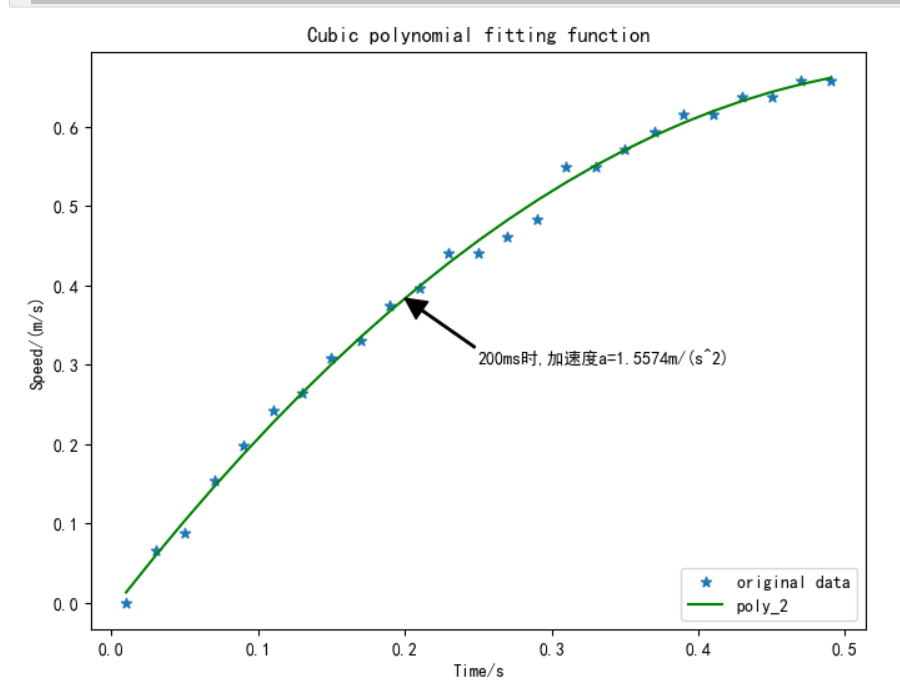




* 1. 性能最优的拟合函数图像：
     1. 思路：
        1. 上述五个模型中三次拟合模型误差平方和最小，但是二次拟合误差与其相差无几，根据奥卡姆剃刀原则，性能相近时，应该选择最简单的，故应该选择二次拟合
        2. 200ms时的加速度即为该二次多项式在0.2s时的一阶导数值。可利用np.polyder(fund)求多项式fund的一阶导数，然后求其在x=0.2处的函数值，即为加速度a
        3. 使用plt.annotate()利用箭头在曲线图上标注200ms时的加速度a
     2. 代码实现



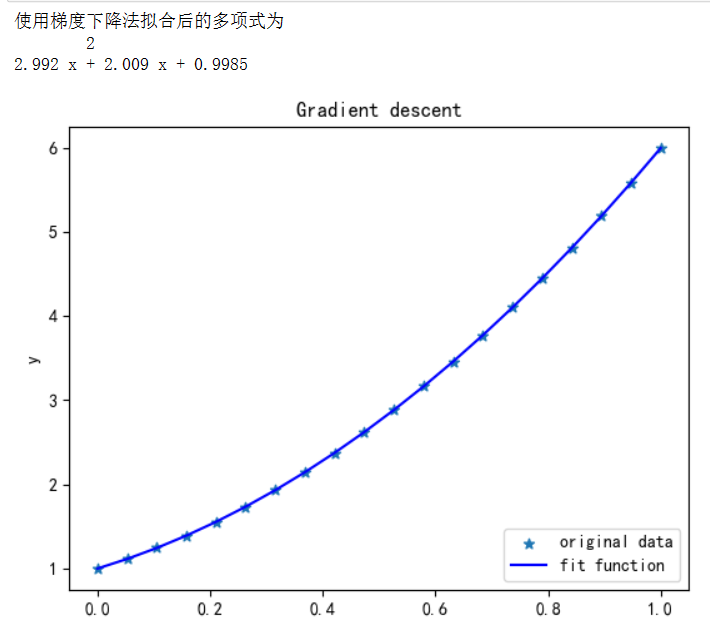
* + 1. 结果



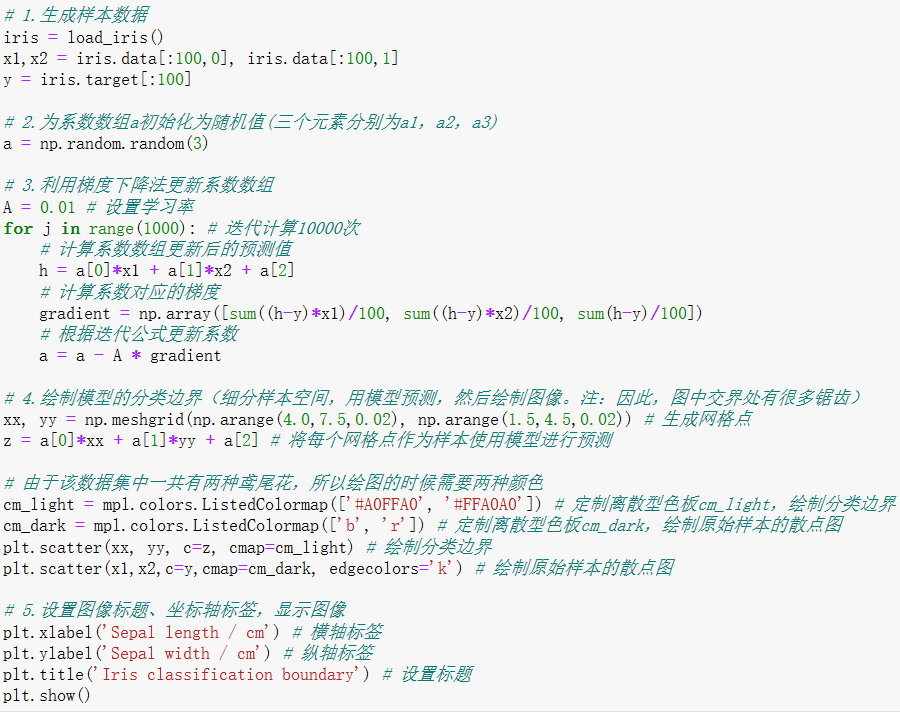
1. 利用梯度下降法拟合函数
   1. 思路：（注意：在模型性能表现不佳时，可以尝试调整学习率和迭代次数，观察模型预测效果）
      1. 生成样本数据(x,y)并绘制样本集对应的散点图
      2. 为系数数组a初始化为随机值(三个元素分别为a1，a2，a3)
      3. 利用梯度下降法更新系数数组
         1. 设置学习率A
         2. 计算系数数组更新后的预测值 h = a[0] + a[1]\*x + a[2]\*x\*\*2
         3. 计算系数数组对应的梯度数组 gradient = np.array([sum(h-y)/20, sum((h-y)\*x)/20, sum((h-y)\*x\*\*2)/20])
         4. 根据梯度数组和学习率更新系数数组 a = a - A \* gradient
         5. 重复2）-4）步，直至到达迭代次数
      4. 根据迭代1000次后的拟合函数，计算各样本的预测值
      5. 绘制拟合函数的曲线
      6. 生成拟合后的多项式，并输出
   2. 代码实现

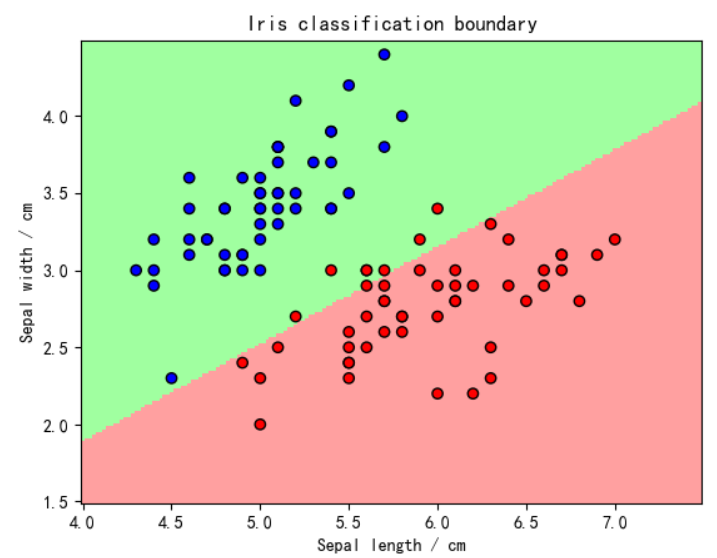


* 1. 结果



1. 利用梯度下降法处理鸢尾花数据集（共两种方法，一种直接使用LogisticRegression模型，另一种使用梯度下降法处理，由分类结果可知梯度下降法精度较差）
   1. 思路：
      1. 利用梯度下降法或直接使用LogisticRegression类训练模型
      2. 利用np.arange()将两个特征的值域细分，之后使用np.meshgrid()生成网格点，利用训练所得模型对每个网格点进行预测
      3. 定制离散型色板cm\_dark和cm\_light，利用原始数据集绘制原始样本的散点图，利用网格点及其预测值绘制模型的分类界面
   2. 方法一：梯度下降

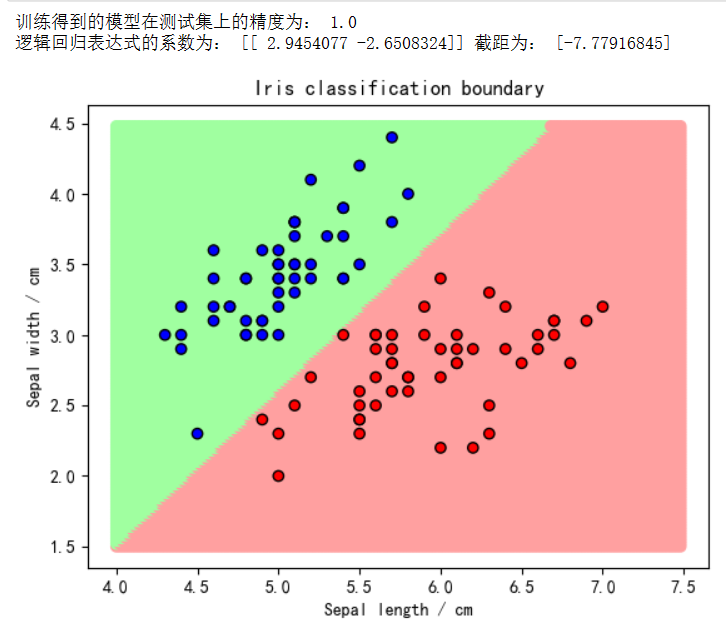




* 1. 方法二：LogisticsRegression模型







1. **总结**
2. 使用最小二乘法对多项式和曲线函数进行拟合时，可以直接使用python提供的方法，np.polyfit()和scipy.optimize.curve\_fit()，不必自己想办法使用使用数学公式还原拟合的过程。
3. 在使用梯度下降进行多项式回归时，迭代次数不要太少，避免模型性能太差。若模型性能较差，可以试着调整一下学习率，观察模型分类结果。
4. 绘制分类边界时，可以生成一系列间隔小的网格点，用模型预测每个网格点，利用这些网格点及其预测值绘制散点图就可以形成分类边界。使用自定义的离散型色板为每个网格点着色，可以让边界更加清晰。
5. 在处理鸢尾花数据集时，因为梯度下降法受到初始的随机系数值、迭代次数、学习率的影响，需要不断调整参数，性能并不是很好。我们可以直接使用train\_test\_split()方法得到训练集和测试集,然后用python提供的LogisticsRegression模型直接对训练集进行学习得到模型，结果显示该模型性能相较于梯度下降法更优。