ZIXING AI
自兴人工智能

机器学习之——线性回归

主讲：刘丽珏

正则方程(Normal Equation)

$$f(W) = Y^TY - Y^TXW - W^TX^TY + W^TX^TXW$$
$$= Y^TY - 2W^TX^TY + W^TX^TXW$$

$$\frac{\partial W^TX^TXW}{\partial W} = (X^TX + (X^TX)^T)W = 2X^TXW$$

- 对W求导

$f(W)' = 2X^TXW - 2X^TY$

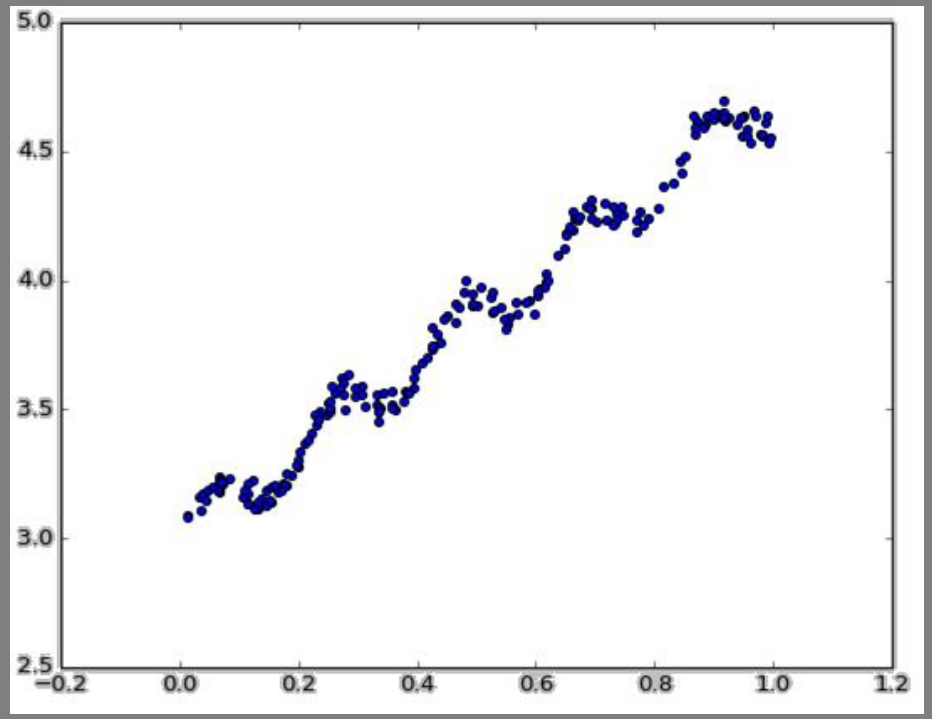
令$X^TXW - X^TY = 0$，求出

$\hat{W} = (X^TX)^{-1}X^TY$　公式（1）**正则方程**

逆矩阵存在的条件$|X^TX|$<>0，
行列式不等于0

线性回归实战

- 对右边的散点图给出最佳拟合直线
  - 对应数据文件——ex0.txt
  - 编程思想
    - 读入数据文件中的数据
    - 建立输入、输出矩阵
    - 根据公式（1）计算回归系数



实验结果

- 输入下列命令输出结果图
  - » import regression
  - » from numpy import *
  - » xArr,yArr=regression.loadDataSet('ex0.txt')
  - » ws=regression.standRegres(xArr,yArr)
  - » xMat=mat(xArr)
  - » yMat=mat(yArr)
  - » yHat=xMat*ws
  - » import matplotlib.pyplot as plt
  - » fig=plt.figure()
  - » ax=fig.add_subplot(111)
  - » ax.scatter(xMat[:,1].flatten().A[0], yMat.T[:,0].flatten().A[0])
  - » xCopy=xMat.copy()
  - » xCopy.sort(0)
  - » yHat=xCopy*ws
  - » ax.plot(xCopy[:,1],yHat)
  - » plt.show()

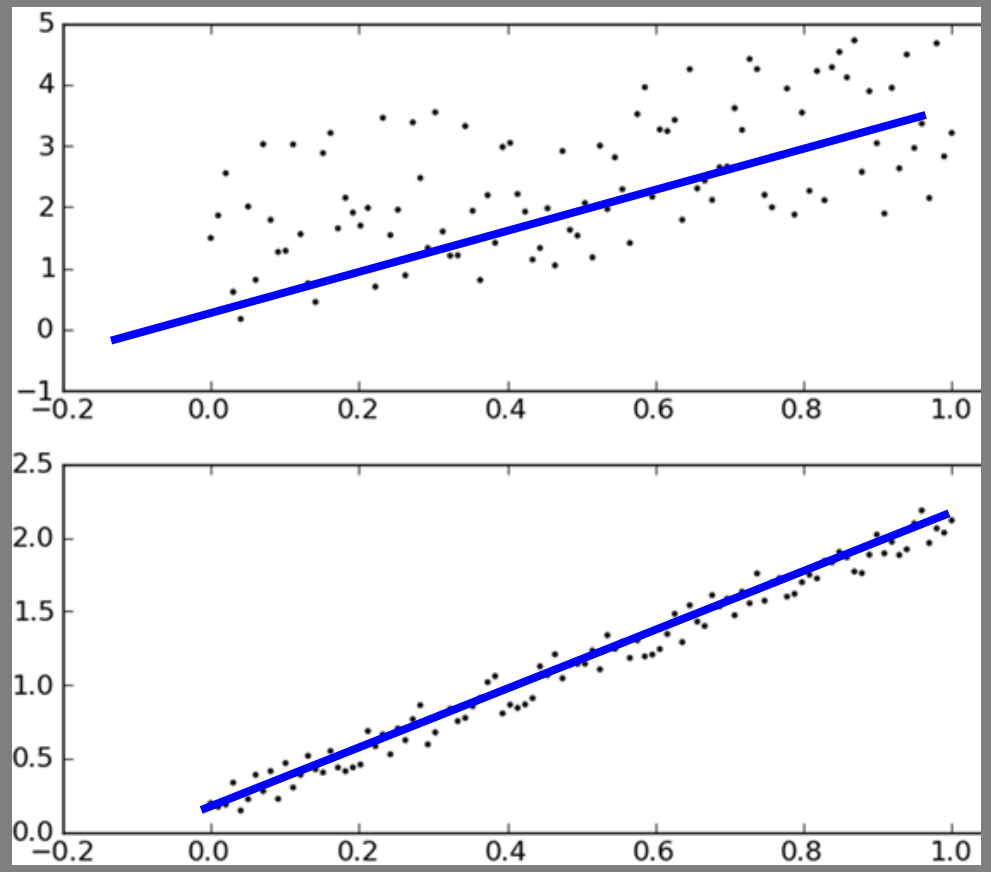
相关系数分析

- 如何判断模型的优劣？
  - 右边两组数据集得到完全相同的回归系数(0, 2.0)
  - 通过预测值和真实值的匹配程度判断优劣
    - 相关系数

$$r(X,Y) = \frac{cov(X,Y)}{\sqrt{D(X)}\sqrt{D(Y)}}$$

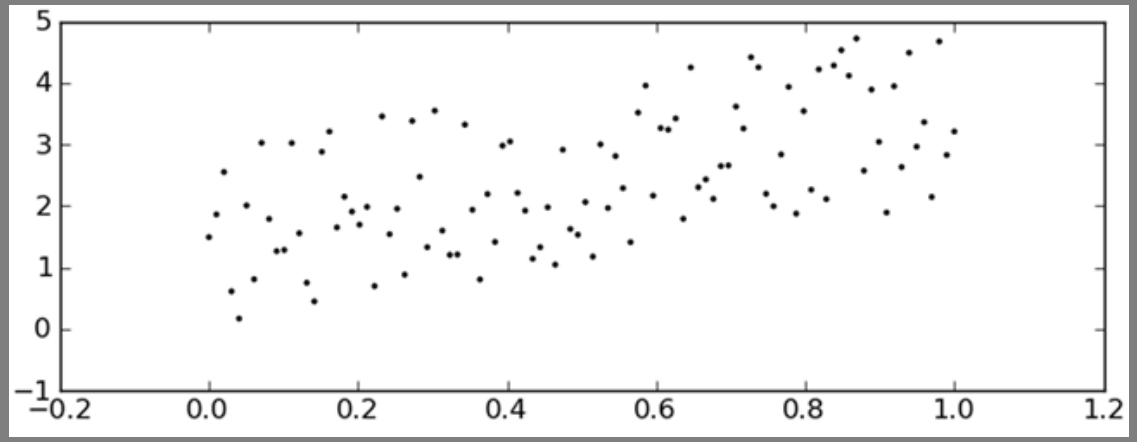
其中$cov(X,Y)$是协方差，$\sqrt{D(X)}$是方差

- 显然$r(X,X) = 1$

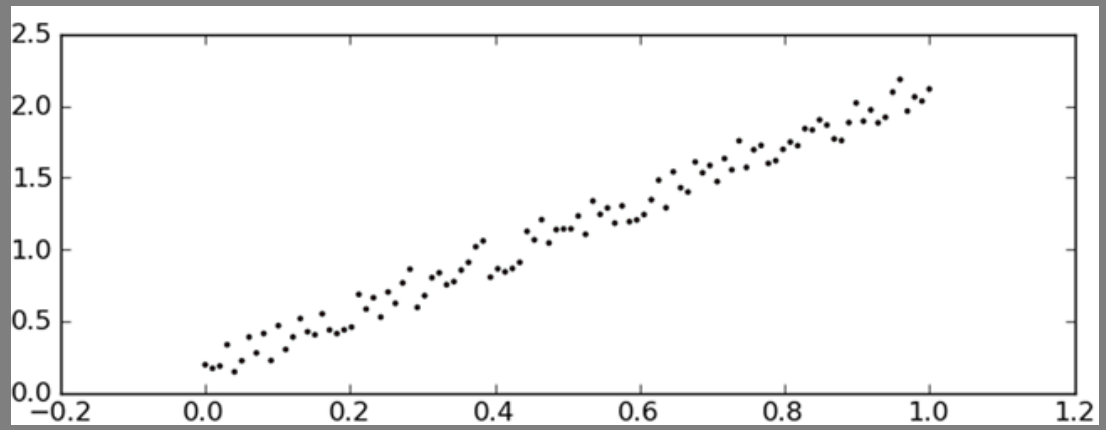


相关系数分析

- NumPy库中相关系数计算方法
  - corrcoef(yHat.T, yMat)
- 请分析根据ex0.txt数据集得到的结果的相关系数
  - 0.98647356



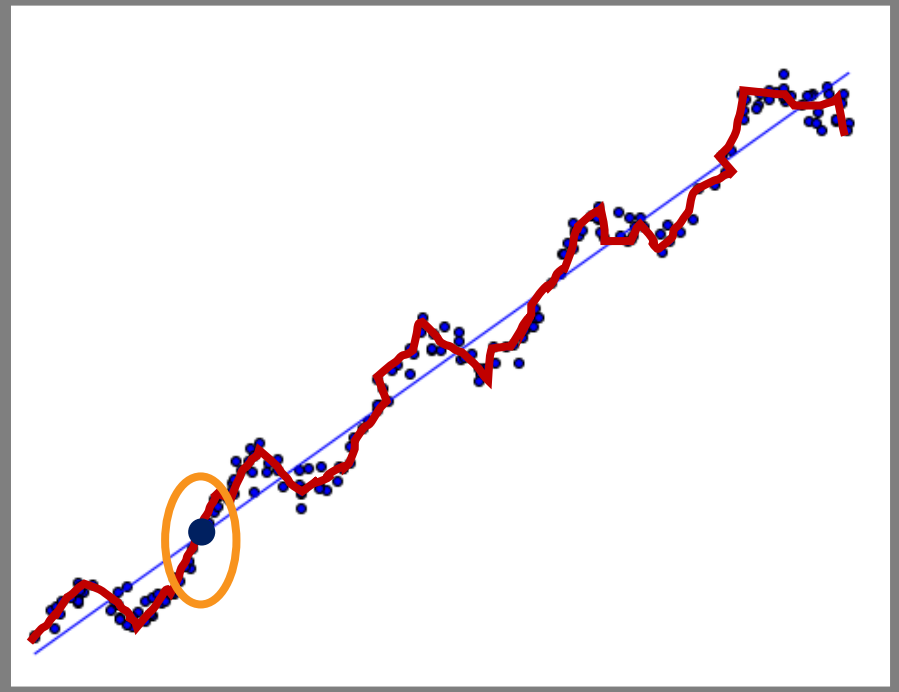
相关系数为0.58



相关系数为0.99

OLR存在的问题及改进

- 欠拟合现象常见
- 若出现欠拟合则不能取得最好的预测效果
- 显然红线的拟合效果更好
- 改进
  - 原算法所有输入输出对（x，y）采用同样的回归系数，画出一条直线
  - 若每个（x，y）有自己的回归系数则会出现一条折线
    - 预测点附近的点有更高的权重参与回归分析



梯度下降的一些注意事项

- 数据归一化
  - 由于样本不同特征的取值范围不一样，可能导致迭代很慢，为了减少特征取值的影响，可以对特征数据归一化
  - 常用归一化公式
    - $x = \frac{x-u}{\sigma}$
      - 其中u为均值，σ为均方差，x为特征
    - $x = \frac{x-x_{min}}{x_{max}-x_{min}}$

梯度下降算法

每个回归系数初始化为1

LOOP

计算整个数据集的梯度

$$\Delta = -\sum_{i=1}^{n} X_i(y_i - X_i W)$$

更新回归系数

$$\hat{W} = \hat{W} - \alpha\Delta$$

ENDLOOP

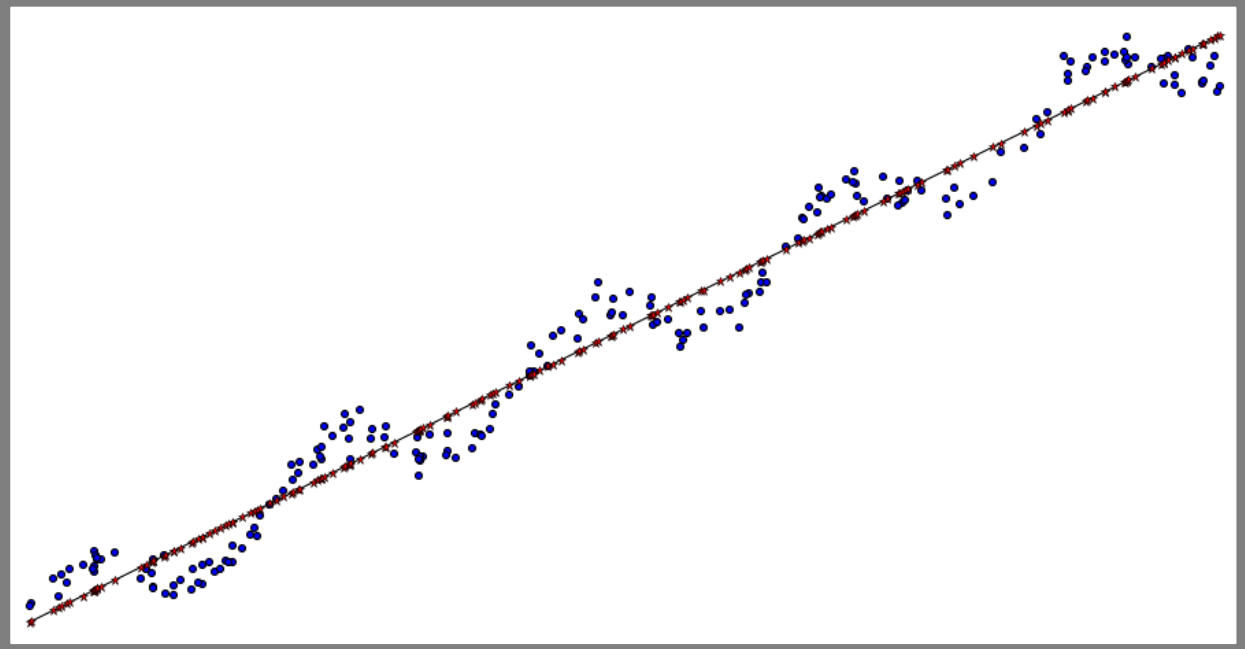
梯度下降实战

- 修改线性回归的代码，加入gradDscent(xArr, yArr)函数
- 结果相比较
  - 解析解

[[3.00774324]
[1.69532264]]

  - 梯度下降

[[3.00758726]
[1.69562035]]



随机梯度下降(Stochastic Gradient Descent)

- 算法步骤

  BEGIN

  所有回归系数初始化为1

  对数据集中每个样本

  计算该样本的梯度

  更新回归系数值

  返回回归系数值

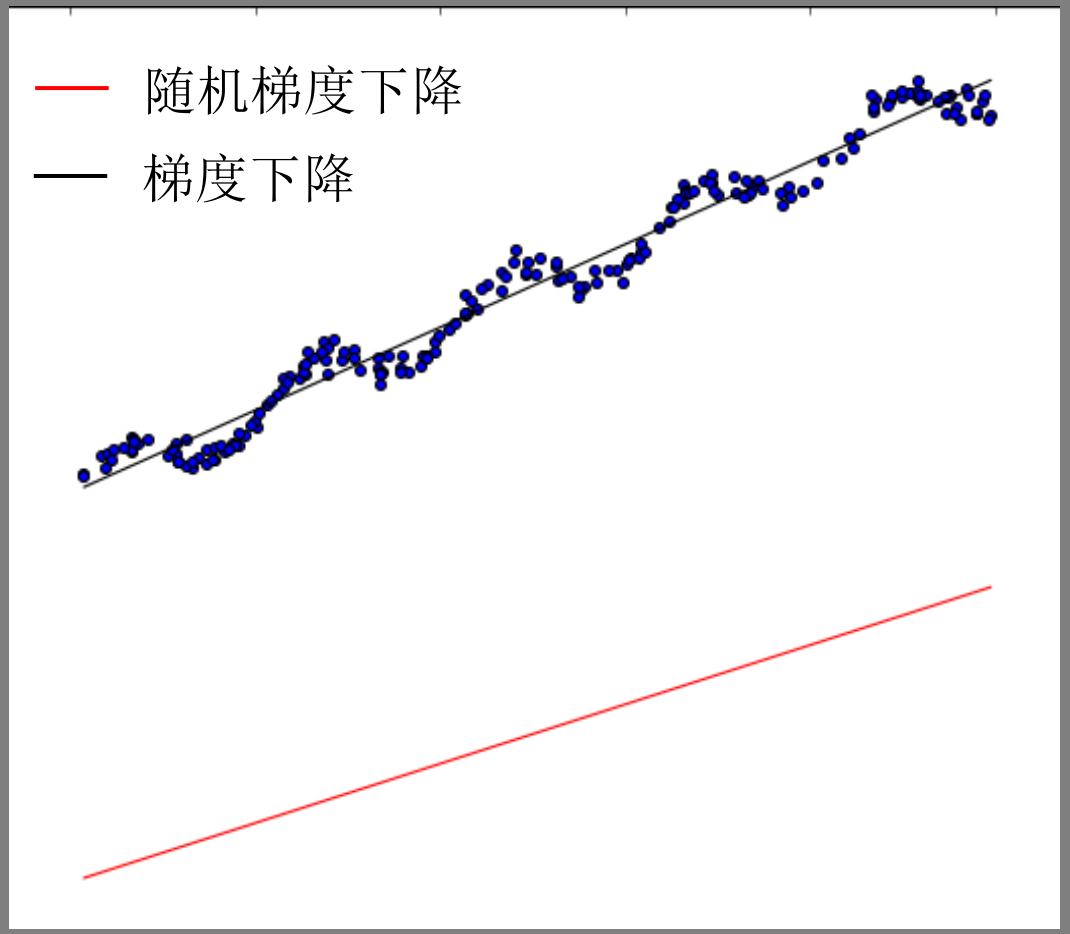
  END

- 修改代码
- 实验结果
  - 随机梯度下降的回归系数

    [1.4159566 1.21208609]
  - 梯度下降的回归系数

    [[3.00758726] [1.69562035]]

GD VS. SGD



- 随机梯度下降的结果明显不如梯度下降
- 但直接比较两者的结果并不公平
  - GD针对整个数据集迭代了500次
  - SGD的迭代次数只是数据集中样本的个数

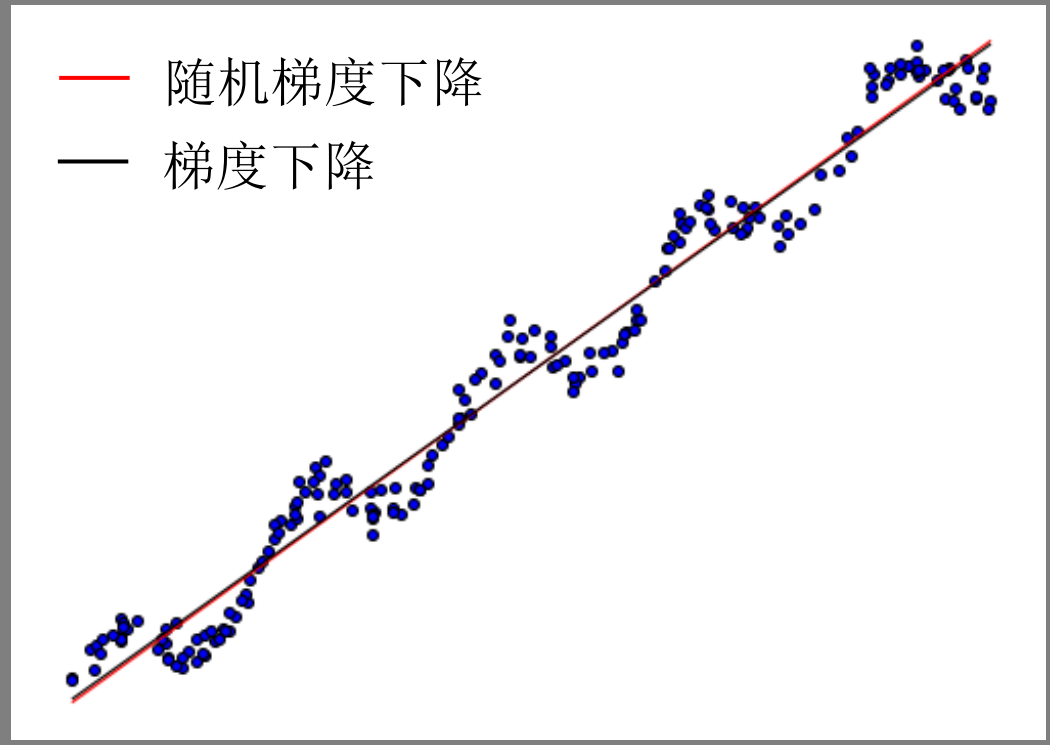
实验结果分析

- 修改代码
  - 增加步长，加快调整步伐
  - 让SGD针对所有样本迭代200次
- 第二种修改实验结果
  - SGD

[2.99841277 1.7135262]

  - GD

[[3.00758726] [1.69562035]]



随机梯度下降的改进

- 步长决定了每一次回归系数调整的幅度
  - 大的步长能加快收敛速度，但有可能错过最优值
  - 小的步长收敛速度慢，但有助于找到最优值
- 改进
  - 步长 $\alpha=f(iter)$
    - *iter*为迭代的代数
  - 即令步长等于当前迭代数的函数
    - 一般为线性函数
    - 随*iter*增加减少

练习

- 分别采用梯度下降和随机梯度下降算法对ex0.txt，ex1.txt数据集进行分析
- 改进SGD，设计一个步长的调整函数，尝试不同的参数，与未改进的SGD进行比较
- 将GD，SGD，ASGD的结果画在一张图上，用不同颜色表示，并给出最后的回归系数
- 分析OLR，LWLR，GD，SGD和改进的SGD的结果，计算相关系数，并自行设计表格，列表比较

练习——房价预测

- 数据文件：ex1data1.txt, ex1data2.txt

文件名	特征数	特征名	类型	样本数量
ex1data1.txt	1	房屋面积	连续型	97
		售价（标签）	连续型	
ex1data2.txt	2	房屋面积	连续型	47
		房间数量	整型	
		售价（标签）	连续型	

项目1——鲍鱼年龄预测

- 实验数据来自UCI数据集
  - 鲍鱼年龄可从鲍鱼壳上的年轮推算
  - 共4177条数据，8个特征，1个标签，无缺失数据

特征名	类型	单位	描述
性别（Sex）	标称型	--	M（雄），F（雌），I（婴儿）
长度（Length）	连续型	mm	外壳最长方向的长度
直径（Diameter）	连续型	mm	垂直于长度的尺寸
高度（Height）	连续型	mm	连壳带肉的厚度
总重（Whole weight ）	连续型	克	整个鲍鱼的重量
去壳重（Shucked weight ）	连续型	克	肉的重量
内脏重量（Viscera weight ）	连续型	克	放血后肠道重量
壳重（Shell weight ）	连续型	克	干燥后重量
轮数（Rings）	整型	--	标签，加1.5为鲍鱼的年龄

项目2——波士顿房价预测

- 波士顿房价数据集
  - 包含对房价的预测，以千美元计，给定的条件是房屋及其相邻房屋的详细信息
  - 共有 506 个样本，13 个特征和1个标签

特征名	描述	特征名	描述
CRIM	城镇人均犯罪率。	DIS	到波士顿五个中心区域的加权距离。
ZN	住宅用地超过 25000 sq.ft. 的比例。	RAD	辐射性公路的接近指数。
INDUS	城镇非零售商用土地的比例。	TAX	每 10000 美元的全值财产税率。
CHAS	查理斯河空变量（如果边界是河流，则为1；否则为0）。	PTRATIO	城镇师生比例。
NOX	一氧化氮浓度。	B	1000（Bk-0.63）^ 2，其中 Bk 指代城镇中黑人的比例。
RM	住宅平均房间数。	LSTAT	人口中地位低下者的比例。
AGE	1940 年之前建成的自用房屋比例。	MEDV	自住房的平均房价，以千美元计。

实验要求

- 在代码中加入分析预测误差的计算，用该误差来衡量预测的准确度
  - 误差计算公式

    $$rssError = \sum_{i=1}^{n}(y_i - y_i')^2$$

  - 从数据集中分出两部分，一部分作为训练集，一部分作为测试集
  - 分别采用OLR，k=0.1, 1, 10的LWLR，梯度下降和采用不同步长参数的随机梯度下降进行回归分析，得到预测模型
  - 计算每个模型在训练集上的误差，列表比较
  - 计算每个模型在测试集上的误差，列表比较
  - 分析上述实验结果，你得到什么启发？
  - 对所有数据进行实验，用所有数据的前面70%做训练集，剩下的做测试集进行交叉验证，可分工测试不同的参数，比比看谁的结果最优？