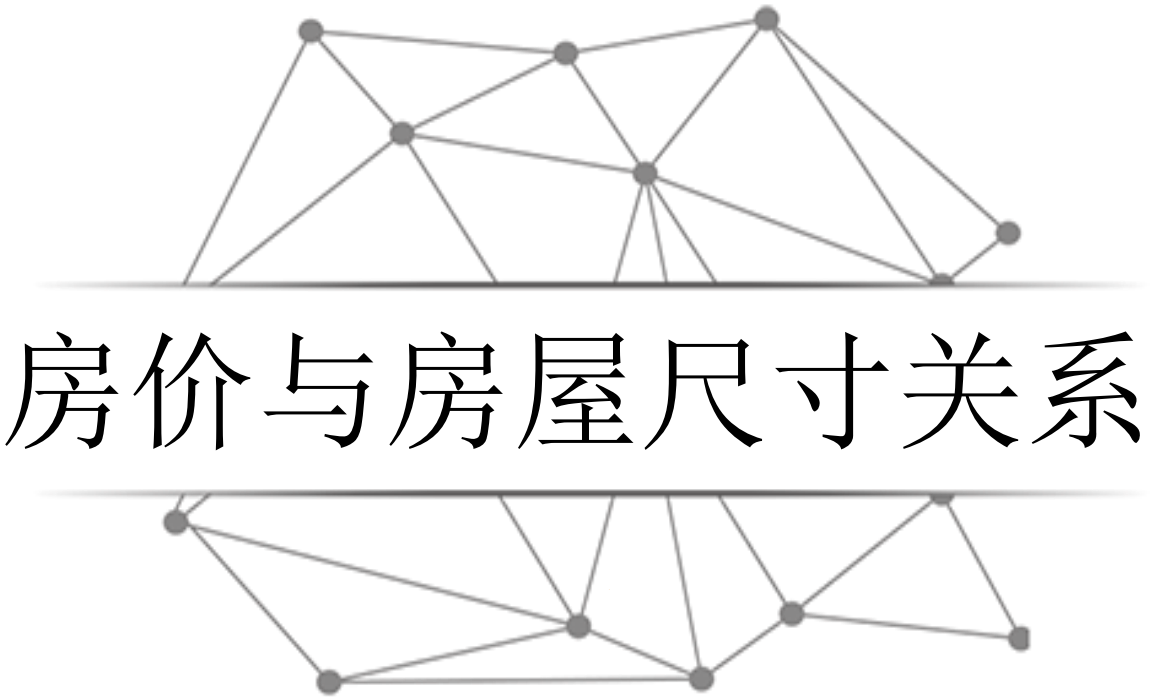
监督学习

ML16

礼欣

www.python123.org



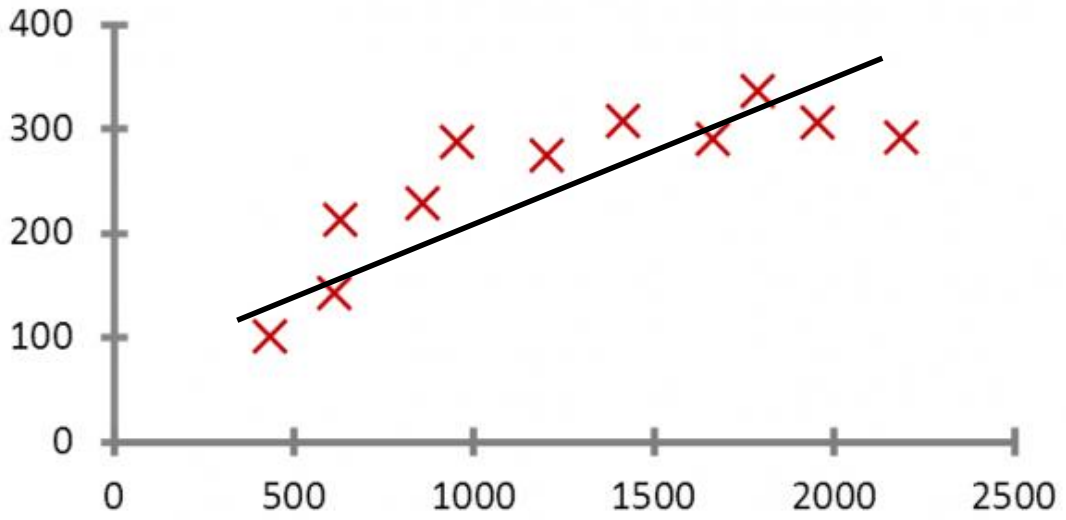
线性回归+房价与房屋尺寸关系的线性拟合

线性回归

- 线性回归(Linear Regression)是利用数理统计中回归分析，来确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法。
- 线性回归利用称为线性回归方程的最小平方函数对一个或多个自变量和因变量之间关系进行建模。这种函数是一个或多个称为回归系数的模型参数的线性组合。只有一个自变量的情况称为简单回归,大于一个自变量情况的叫做多元回归。

线性回归

线性回归：使用形如$y=w^Tx+b$的线性模型拟合数据输入和输出之间的映射关系的。



线性回归的实际用途

线性回归有很多实际的用途，分为以下两类：

1.如果目标是预测或者映射，线性回归可以用来对观测数据集的y和x的值拟合出一个预测模型。当完成这样一个模型以后，对于一个新增的x值，在没有给定与它相配对的y的情况下，可以用这个拟合过的模型预测出一个y值。

2.给定一个变量y和一些变量$X_1,\cdots,X_p$，这些变量有可能与y相关，线性回归分析可以用来量化y与X_j之间相关性的强度，评估出与y不相关的X_j，并识别出哪些X_j的子集包含了关于y的冗余信息。

线性回归的应用

背景：与房价密切相关的除了单位的房价，还有房屋的尺寸。我们可以根据已知的房屋成交价和房屋的尺寸进行线性回归，继而可以对已知房屋尺寸，而未知房屋成交价格的实例进行成交价格的预测。

目标：对房屋成交信息建立回归方程，并依据回归方程对房屋价格进行预测

技术路线：`sklearn.linear_model.LinearRegression`

实例数据

为了方便展示，成交信息只使用了房屋的面积以及对应的成交价格。其中：

- 房屋面积单位为平方英尺（ft^2）房
- 屋成交价格单位为万

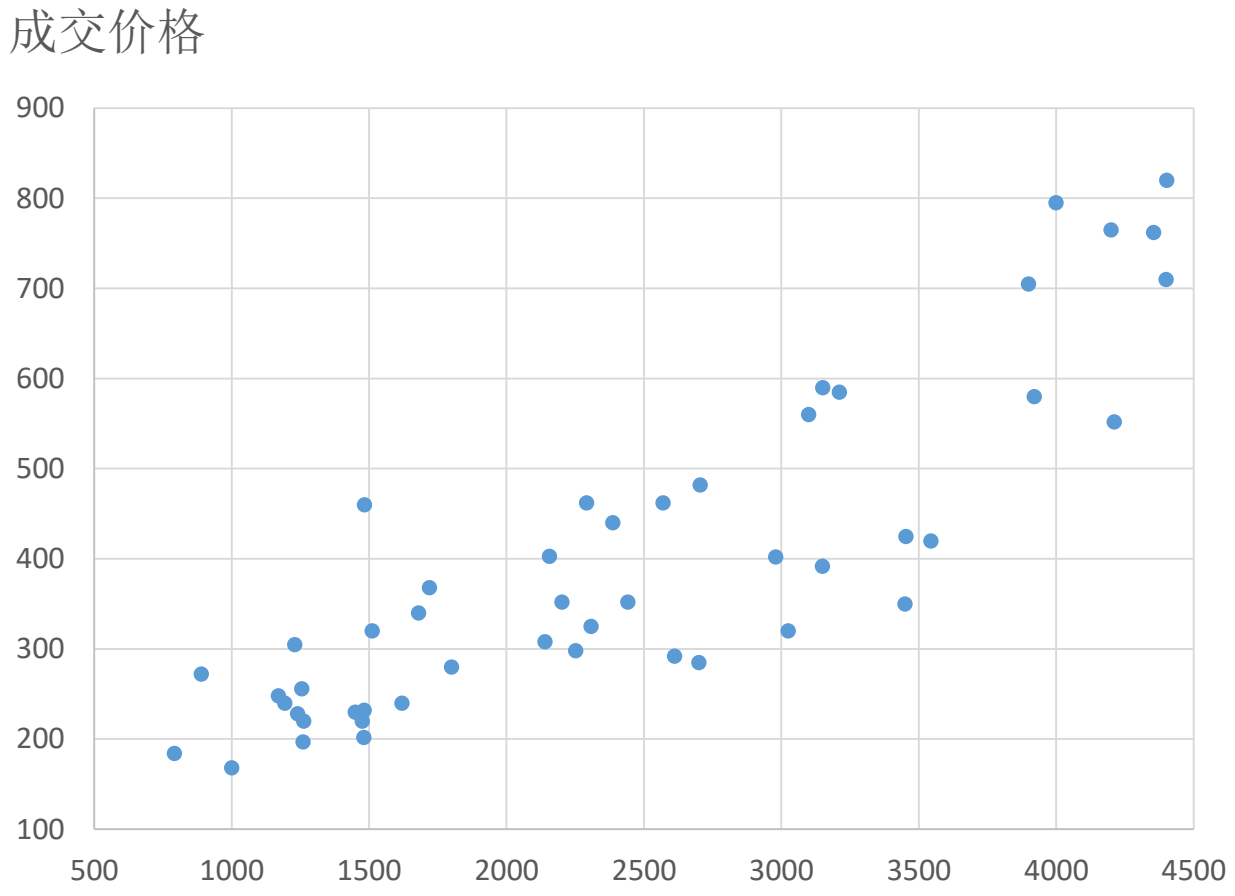
编号	房屋面积/ft^2	交易价格/万	编号	房屋面积/ft^2	交易价格/万
1	1000	168	26	2700	285
2	792	184	27	2612	292
3	1260	197	28	2705	482
4	1262	220	29	2570	462
5	1240	228	30	2442	352
6	1170	248	31	2387	440
7	1230	305	32	2292	462
8	1255	256	33	2308	325
9	1194	240	34	2252	298
10	1450	230	35	2202	352
11	1481	202	36	2157	403
12	1475	220	37	2140	308
13	1482	232	38	4000	795
14	1484	460	39	4200	765
15	1512	320	40	3900	705
16	1680	340	41	3544	420
17	1620	240	42	2980	402
18	1720	368	43	4355	762
19	1800	280	44	3150	392
20	4400	710	45	3025	320
21	4212	552	46	3450	350
22	3920	580	47	4402	820
23	3212	585	48	3454	425
24	3151	590	49	890	272
25	3100	560			

可行性分析

- 简单而直观的方式是通过数据的可视化直接观察房屋成交价格与房屋尺寸间是否存在线性关系。
- 对于本实验的数据来说，散点图就可以很好的将其在二维平面中进行可视化表示。

可行性分析

右图为数据的散点图，其中横坐标为房屋面积，纵坐标为房屋的成交价格。可以看出，靠近坐标左下角部分的点，表示房屋尺寸较小的房子，其对应的房屋成交价格也相对较低。同样的，靠近坐标右上部分的点对应于大尺寸高价格的房屋。从总体来看，房屋的面积和成交价格基本成正比。



实验过程

使用算法：线性回归

实现步骤：

1.建立工程并导入sklearn包

2.加载训练数据，建立回归方程

3.可视化处理

关于一些相关包的介绍：

- NumPy是Python语言的一个扩充程序库。支持高级大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。
- matplotlib的pyplot子库提供了和matlab类似的绘图API，方便用户快速绘制2D图表。

实现步骤——1.建立工程并导入sklearn包

- 创建house.py文件
- 导入sklearn相关包
  - `import matplotlib.pyplot as plt` ⟶ 表示，matplotlib的pyplot子库，它提供了和matlab类似的绘图API。
  - `from sklearn import linear_model`

表示，可以调用sklearn中的linear_model模块进行线性回归。

实现步骤——2.加载训练数据，建立回归方程

```
datasets_X = []
datasets_Y = []
fr = open('prices.txt','r')
lines = fr.readlines()
for line in lines:
    items = line.strip().split(',')
    datasets_X.append(int(items[0]))
    datasets_Y.append(int(items[1]))
length = len(datasets_X)
datasets_X = np.array(datasets_X).reshape([length,1])
datasets_Y = np.array(datasets_Y)
```

建立datasets_X和datasets_Y用来存储数据中的房屋尺寸和房屋成交价格。

打开数据集所在文件prices.txt，读取数据。

一次读取整个文件。

实现步骤——2.加载训练数据，建立回归方程

```
datasets_X = []
datasets_Y = []
fr = open('prices.txt','r')
lines = fr.readlines()
for line in lines:
    items = line.strip().split(',')
    datasets_X.append(int(items[0]))
    datasets_Y.append(int(items[1]))
length = len(datasets_X)
datasets_X = np.array(datasets_X).reshape([length,1])
datasets_Y = np.array(datasets_Y)
```

逐行进行操作，循环遍历所有数据

去除数据文件中的逗号

将读取的数据转换为int型，并分别写入datasets_X和datasets_Y。

实现步骤——2.加载训练数据，建立回归方程

```
datasets_X = []
datasets_Y = []
fr = open('prices.txt','r')
lines = fr.readlines()
for line in lines:
    items = line.strip().split(',')
    datasets_X.append(int(items[0]))
    datasets_Y.append(int(items[1]))
length = len(datasets_X)
datasets_X = np.array(datasets_X).reshape([length,1])
datasets_Y = np.array(datasets_Y)
```

length = len(datasets_X) → 求得datasets_X的长度，即为数据的总数。

datasets_X = np.array(datasets_X).reshape([length,1]) → 将datasets_X转化为数组，并变为二维，以符合线性回归拟合函数输入参数要求。

datasets_Y = np.array(datasets_Y) → 将datasets_Y转化为数组

实现步骤——2.加载训练数据，建立回归方程

- `minX = min(datasets_X)`
- `maxX = max(datasets_X)`
- `X = np.arange(minX,maxX).reshape([-1,1])` ⟶ 以数据datasets_X的最大值和最小值为范围，建立等差数列，方便后续画图。

- `linear = linear_model.LinearRegression()`
- `linear.fit(datasets_X, datasets_Y)` ⟶ 调用线性回归模块，建立回归方程，拟合数据

调用sklearn.linear_model.LinearRegression()所需参数：

- fit_intercept：布尔型参数，表示是否计算该模型截距。可选参数。
- normalize：布尔型参数，若为True，则X在回归前进行归一化。可选参数。默认值为False。
- copy_X：布尔型参数，若为True，则X将被复制；否则将被覆盖。可选参数。默认值为True。
- n_jobs：整型参数，表示用于计算的作业数量；若为-1，则用所有的CPU。可选参数。默认值为1。

实现步骤——2.加载训练数据，建立回归方程

- `minX = min(datasets_X)`
- `maxX = max(datasets_X)`
- `X = np.arange(minX,maxX).reshape([-1,1])` ⟶ 以数据datasets_X的最大值和最小值为范围，建立等差数列，方便后续画图。
- `linear = linear_model.LinearRegression()`
- `linear.fit(datasets_X, datasets_Y)` ⟶ 调用线性回归模块，建立回归方程，拟合数据

线性回归fit函数用于拟合输入输出数据，调用形式为`linear.fit(X,y, sample_weight=None)`：

- X：X为训练向量；
- y：y为相对于X的目标向量；
- sample_weight：分配给各个样本的权重数组，一般不需要使用，可省略。

实现步骤——2.加载训练数据，建立回归方程

- 如果有需要，可以通过两个属性查看回归方程的系数及截距。
- 具体的代码如下：

```
#查看回归方程系数
print('Coefficients:', linear.coef_)
#查看回归方程截距
print('intercept:', linear.intercept_)
```

实现步骤——3.可视化处理

- `plt.scatter(datasets_X, datasets_Y, color = 'red')`
- `plt.plot(X, linear.predict(X), color = 'blue')`
- `plt.xlabel('Area')`
- `plt.ylabel('Price')`
- `plt.show()`

scatter函数用于绘制数据点，这里表示用红色绘制数据点；

plot函数用来绘制直线，这里表示用蓝色绘制回归线；

xlabel和ylabel用来指定横纵坐标的名称。

结果展示

通过回归方程拟合的直线与原有数据点的关系如右图所示，依据该回归方程即可通过房屋的尺寸，来预测房屋的成交价格。