《用 Python 玩转数据》项目—线性回归分析入门之波士顿房价预测

Dazhuang@NJU

一、背景

分类和回归属于机器学习领域有监督学习算法（聚类则属于无监督学习算法）的两种方法，有监督学习是通过已有的训练样本去训练得到一个模型，再使用这个模型将所有的输入映射到相应的输出，若输出结果是离散型，例如今天会下雨（1）或今天不会下雨（0），则称为分类，若输出结果是连续型，例如输入一组房屋的属性预测房价是多少，则称为回归。本项目将利用著名的 boston 房价数据集来介绍简单的线性回归，若模型预测效果不够理想则可在此基础上进一步进行非线性回归的尝试。

二、数据和算法

1. 数据源

boston 房价数据与 iris 和 minst 等一样都是机器学习中著名的基础数据集，我们可以在此基础上进行实验，进一步可爬取网络数据进行类似研究。

获取 boston 房价数据与获取 iris 数据集的方法类似:

```
from sklearn import datasets      # 利用机器学习 Python 包 sklearn
boston = datasets.load_boston()
X = boston.data
y = boston.target
print(X.shape)
(506, 13)
```

boston 房价数据 X 中包含 506 条记录，每条记录包含房屋的 13 条属性，房价信息属性 MEDV 在 boston.target 中，具体（翻译成中文）可通过如下语句查看:

```
>>> print(boston.DESCR)
…
:Attribute Information (in order):
        - CRIM      城镇人均犯罪率
        - ZN        占地面积超过 25,000 平方尺的住宅用地比例
        - INDUS     城镇中非商业用地比例
        - CHAS      Charles River 虚拟变量（如果边界是河流则为 1;否则为 0）
        - NOX       一氧化氮浓度
        - RM        每栋住宅平均房间数
        - AGE       1940 年前建成的自住房屋比例
        - DIS       距五个波士顿就业中心的加权距离
        - RAD       距离高速公路的便利指数
```

- TAX 每 10,000 美元的全额房产税率
- PTRATIO 城镇中学生教师比例
- B 城镇中黑人比例
- LSTAT 人口中低收入阶层比例
- MEDV 自住房房价中位数

2. 线性回归分析

维基百科中对于线性回归的解释为：

在统计学中，线性回归（Linear regression）是利用称为线性回归方程的最小二乘函数对一个或多个自变量和因变量之间关系进行建模的一种回归分析。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。

回归分析的最初目的是估计模型的参数以便达到对数据的最佳拟合。在决定一个最佳拟合的不同标准之中，最小二乘法是非常优越的。

若回归分析中只有一个自变量和一个因变量且两者之间的关系可以用一条直线表示，则称这种回归分析为一元线性回归分析，若包含两个或两个以上自变量，且因变量和自变量之间是线性关系，则称这种回归分析为多元线性回归分析。线性回归分析的完整表达形式为：

$y = \alpha+\beta_1x_1+\beta_2x_2+\ldots+\beta_nx_n+\varepsilon$, 其中 x 为自变量，y 为因变量，ε 为误差

先以最简单的一元线性回归来了解线性回归的使用，再进一步看如何用多元线性回归进行分析。

一元线性回归的最简单形式为：

$y = \alpha+\beta x$, 其中 α 为截距，β 为回归系数即斜率

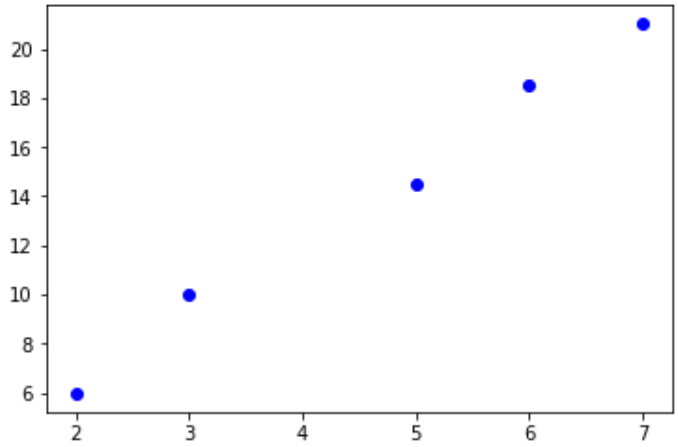
为达到最佳拟合（预测值与实际值尽量接近，损失小），要使残差（误差的观测值）平方和 $SS_{res} = \sum_{i=1}^{n}(y_i - f(x_i))^2$最小，常通过最小二乘法（另外还常用梯度下降法迭代计算）计算，要使 SS_{res} 最小，只要关于 α 和 β 的偏导数为 0，推导得 β 估量值的计算公式为（α 根据 β 值和公式得到）[1]：

$$\beta = \frac{\frac{1}{n}(\sum_{i=1}^{n} x_i\, y_i) - \bar{y}\bar{x}}{\frac{1}{n}(\sum_{i=1}^{n} {x_i}^2) - (\bar{x})^2}$$

假设有一组特征 x 为 2、3、5、7、6，标签 y 为 6、10、14.5、21、18.5，将两者之间的关系用散点图进行展现：

```
X = np.array([2,3,5,7,6]).reshape(-1,1)
# -1 表示基于另一个确定的值如这里第二维为 1 进行维度计算，可将数组转换成 N*1 形状，根据需要可以有如 reshape(2, -1)和 reshpae(-1,4)这样的多种灵活的写法
y = np.array([6,10,14.5,21,18.5])
plt.scatter(X, y, color='blue')
```

[1] 一元和多元线性回归的详细数学推导请参见相关资料



由图可以看到两组数据之间呈现较明显的线性关系，我们可以编写表达式用程序计算出结果，更方便的是可以借助如 statsmodels 或 sklearn 来轻松地完成上述任务，后者代码如下所示：

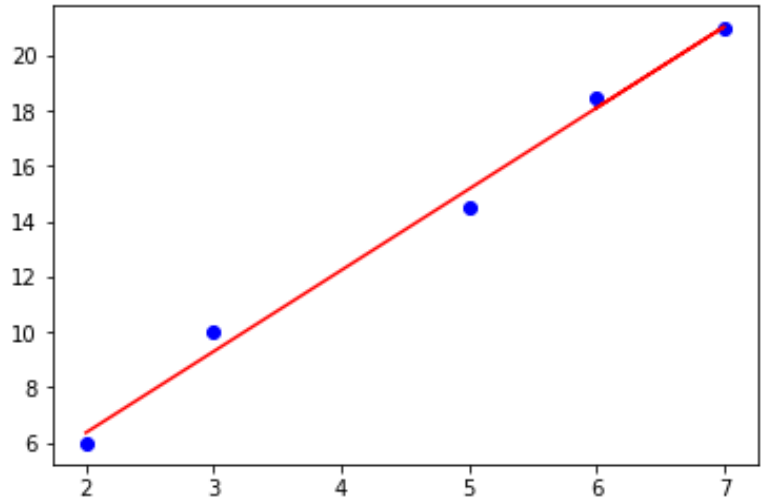
```
from sklearn import linear_model
import numpy as np

clf = linear_model.LinearRegression()
X = np.array([2,3,5,7,6]).reshape(-1,1)
y = np.array([6,10,14.5,21,18.5])
clf.fit(X,y)       # 训练模型
b, a = clf.coef_, clf.intercept_
print(b, a)
x = [[4]]
print(clf.predict(x))      # 预测
```

执行程序，b 和 a 的值分别为：

```
array([[2.93604651]])
array([0.49418605])
```

方程即为 y = 0.49418605+2.93604651x，预测值计算得 12.23837209。

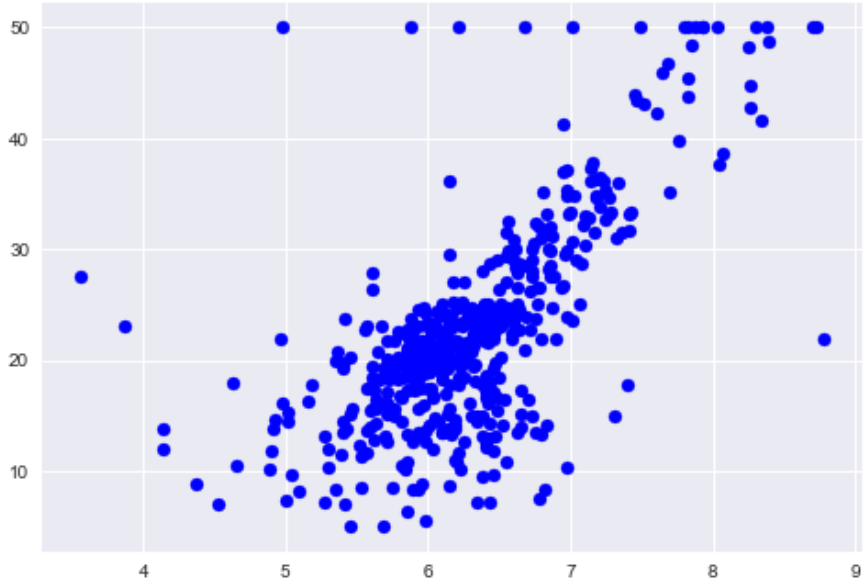
基于方程可以将拟合的直线绘制出来（plt.plot(X, a+b*X, color = 'red')，效果如下：



以下进一步讨论多元线性回归问题。

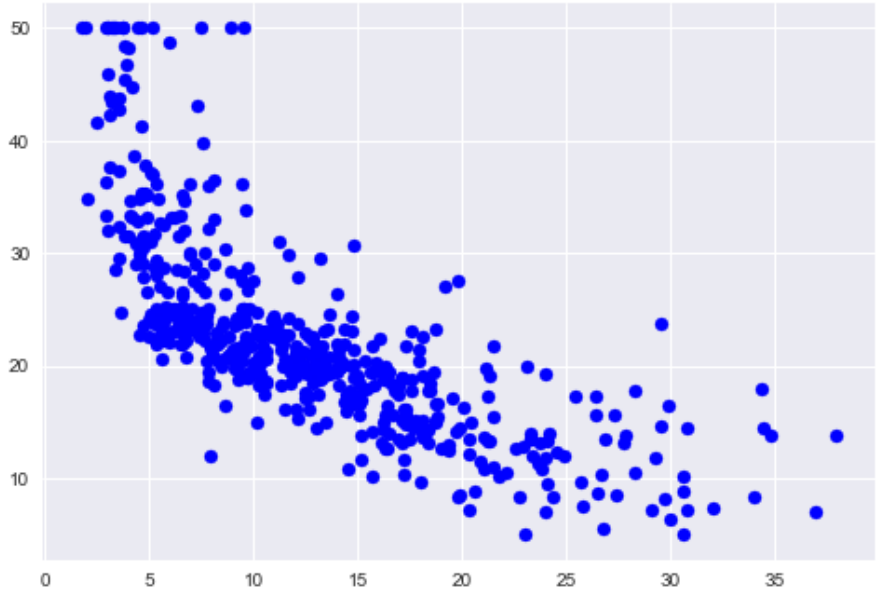
boston 房价数据中的房屋属性（多个）和房价 MEDV 之间是否存在线性关系呢？简单地进行绘制，例如与房价相关度高的每栋住宅平均房间数 RM 属性与房价 MEDV 之间的散点图如下：

```
>>> plt.scatter(X['RM'], y, color='blue')
```



人口中低收入阶层比例 LSTAT 与 MEDV 之间的散点图如下：

```
>>> plt.scatter(X['LSTAT'], y, color='blue')
```



从图看存在一定的线性关系，可以用线性回归模型进行尝试。boston 房价数据中有 13 个房屋属性，到底选择哪些属性作为回归方程的自变量呢？一种思路是借助相关分析查看与房屋价格 MEDV 相关性比较高的属性，也可以借助 sklearn.feature_selection 中的 SelectKBest() 函数指定最佳属性的个数选择出最佳的若干个属性，还可以借助用于确定自变量显著性的 T 检验来选择特征。以最后一种方法为例，设定一个衡量显著性的阈值，一般取 0.05，将所有特征放入模型进行训练，计算每个特征的 p 值（p 值若与选定显著性水平（0.05 或更低如 0.01）相比更小，则零假设即该系数在模型中本来不应该存在会被否定而不可接受，因此选

择该特征),将 p 值高于阈值的特征从训练模型中移除,再利用剩下的特征进行新一轮拟合,如果还存在 p 值高于阈值的特征则继续移除,直到均满足条件。以下利用 Python 统计分析模块 statsmodels 来查看特征 p 值和进行模型的训练和预测:

```
boston = datasets.load_boston()
# 房屋 13 个属性数据
X = pd.DataFrame(boston.data, columns = boston.feature_names)
y = pd.DataFrame(boston.target, columns = ['MEDV'])
import statsmodels.api as sm
# statsmodels 中的线性回归模型没有截距项,下行给训练集加上一列数值为 1 的特征
X_add1 = sm.add_constant(X)
model = sm.OLS(y, X_add1).fit() # sm.OLS()为普通最小二乘回归模型,fit()用于拟合
print (model.summary())
```

输出结果为:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   MEDV   R-squared:                       0.741
Model:                            OLS   Adj. R-squared:                  0.734
Method:                 Least Squares   F-statistic:                     108.1
Date:                Mon, 05 Nov 2018   Prob (F-statistic):          6.95e-135
Time:                        21:13:38   Log-Likelihood:                -1498.8
No. Observations:                 506   AIC:                             3026.
Df Residuals:                     492   BIC:                             3085.
Df Model:                          13
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const         36.4911      5.104      7.149      0.000        26.462    46.520
CRIM          -0.1072      0.033     -3.276      0.001        -0.171    -0.043
ZN             0.0464      0.014      3.380      0.001         0.019     0.073
INDUS          0.0209      0.061      0.339      0.735        -0.100     0.142
CHAS           2.6886      0.862      3.120      0.002         0.996     4.381
NOX          -17.7958      3.821     -4.658      0.000       -25.302   -10.289
RM             3.8048      0.418      9.102      0.000         2.983     4.626
AGE            0.0008      0.013      0.057      0.955        -0.025     0.027
DIS           -1.4758      0.199     -7.398      0.000        -1.868    -1.084
RAD            0.3057      0.066      4.608      0.000         0.175     0.436
TAX           -0.0123      0.004     -3.278      0.001        -0.020    -0.005
PTRATIO       -0.9535      0.131     -7.287      0.000        -1.211    -0.696
B              0.0094      0.003      3.500      0.001         0.004     0.015
LSTAT         -0.5255      0.051    -10.366      0.000        -0.625    -0.426
==============================================================================
Omnibus:                      178.029   Durbin-Watson:                   1.078
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              782.015
Skew:                           1.521   Prob(JB):                    1.54e-170
Kurtosis:                       8.276   Cond. No.                     1.51e+04
==============================================================================
```

可以看到属性 AGE 的 p 值远高于阈值 0.05,所以将其属性移除,继续训练,之后发现 INDUS 属性仍然原高于阈值,同样将 INDUS 属性也移除,最后留下 11 个属性进行训练。

```
# 移除两个属性/特征
X.drop('AGE', axis = 1, inplace = True)
```

```
X.drop('INDUS', axis = 1, inplace = True)
# 重新训练
X_add1 = sm.add_constant(X)
model = sm.OLS(y, X_add1).fit()
print(model.summary())
```

输出结果为：

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   MEDV   R-squared:                       0.741
Model:                            OLS   Adj. R-squared:                  0.735
Method:                 Least Squares   F-statistic:                     128.2
Date:                Mon, 05 Nov 2018   Prob (F-statistic):          5.74e-137
Time:                        22:17:56   Log-Likelihood:                -1498.9
No. Observations:                 506   AIC:                             3022.
Df Residuals:                     494   BIC:                             3073.
Df Model:                          11
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const         36.3694      5.069      7.176      0.000        26.411    46.328
CRIM          -0.1076      0.033     -3.296      0.001        -0.172    -0.043
ZN             0.0458      0.014      3.387      0.001         0.019     0.072
CHAS           2.7212      0.854      3.185      0.002         1.043     4.400
NOX          -17.3956      3.536     -4.920      0.000       -24.343   -10.448
RM             3.7966      0.406      9.343      0.000         2.998     4.595
DIS           -1.4934      0.186     -8.039      0.000        -1.858    -1.128
RAD            0.2991      0.063      4.719      0.000         0.175     0.424
TAX           -0.0118      0.003     -3.488      0.001        -0.018    -0.005
PTRATIO       -0.9471      0.129     -7.337      0.000        -1.201    -0.693
B              0.0094      0.003      3.508      0.000         0.004     0.015
LSTAT         -0.5232      0.047    -11.037      0.000        -0.616    -0.430
==============================================================================
Omnibus:                      178.444   Durbin-Watson:                   1.078
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              786.944
Skew:                           1.524   Prob(JB):                    1.31e-171
Kurtosis:                       8.295   Cond. No.                     1.47e+04
==============================================================================
```

coef 列即为计算出的回归系数，也可用“print(model.params)”输出，回归方程即为：

$y=36.3694-0.1076x_1+0.0458x_2+2.7212x_3-17.3956x_4+3.7966x_5-1.4934x_6+0.2991x_7-0.0118x_8-0.9471x_9+0.0094x_{10}-0.5232x_{11}$

假设测试数据如下：

```
X_test = np.array([[1, 0.006, 18.0, 0.0, 0.52, 6.6, 4.87, 1.0, 290.0, 15.2, 396.2, 5]])        # 第一个数 1 为同样添加的常数项
```

将测试数据代入方程即可进行计算预测，简单地还可以直接利用模型的 predict()获得预测结果：

```
print(model.predict(X_test))
[29.52160882]
```

预测结果为 29.52160882。

另外要说明的是，输出结果中的“Adj. R-squared”即为调整 R^2，源于 R^2 但比 R^2 更严谨更适合用来表示模型的拟合程度，R^2 也叫确定系数（coefficient of determination），表示模

型对现实数据拟合的程度，它是皮尔逊相关系数的平方，从数值上说，R^2介于 0~1 之间，越接近 1 表示拟合效果越好，0.735 这个数字说明测试集中有 7 成多的数据可通过模型解释，拟合效果较好，但还有一定的空间，一般认为超过 0.8 的模型拟合优度比较高，可以尝试使用非线性模型进行预测。初学者可以先关注调整 R^2 和 T 检验的 p 值，后续可以继续理解利用 F 检验检查方程的整体显著性以及进行残差分析等，有兴趣的学习者可进行深入研究。

三、参考资料

1. 回归分析概念

各统计学和机器学习相关资料，如经典的李航老师的《统计学习方法》

2. sklearn 之线性回归

http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

或其他相关资料

3. statsmodel 之线性回归

http://www.statsmodels.org/dev/regression.html

或其他相关资料