可能是最全面的线性回归的波士顿房价预测实验-Python数据分析

原创

黑芝麻大汤圆 于 2020-05-13 10:41:04 发布 5255 收藏 65

分类专栏： 人工智能 文章标签： 数据分析 线性规划 线性代数 python

本文链接：https://blog.csdn.net/b_z_K____0012/article/details/106090242

Python数据分析-基于线性回归的波士顿房价预测实验

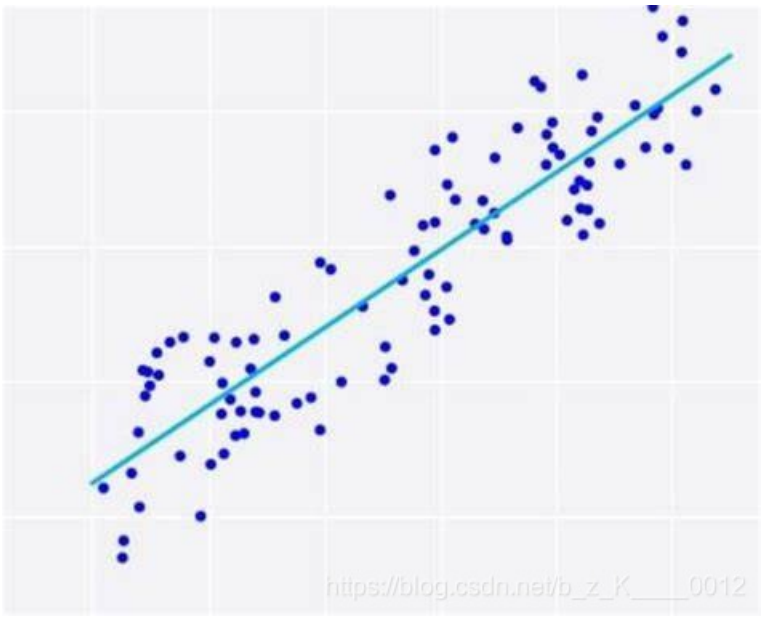
最近尝试着早起，因为六点起床的话就会发现早上的时间很充足，而且公园里慢跑的人真的很多，为优秀的人们献上我的膝盖。
进入正题，这是数据分析的第一个实验，老早就听说过这个实验，但是没有自己做过，一起来做一下吧。

一、 实验准备

1.1 实验概述

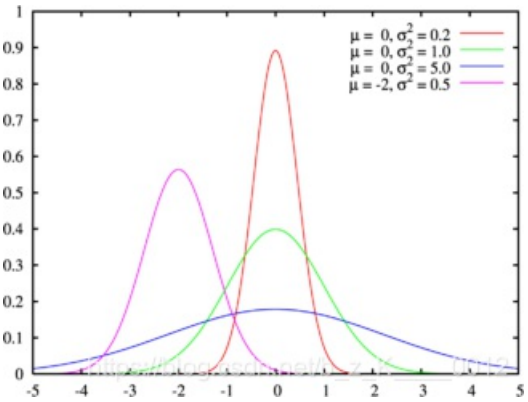
线性回归 (Linear regression): 是利用数理统计中回归分析来确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法。
如果回归分析中，只包括一个自变量和一个因变量，且二者的关系可用一条直线近似表示，这种回归分析称为一元线性回归分析。方程如下：$Y=\mathrm{a}_{x}+\mathrm{b}$

式中，x 代表t期自变量的值；
Y代表t期因变量的值； a、b代表一元线性回归方程的参数。
我从网上找了一张一元线性回归方程的图片，就是这个样子的：



误差：真实值与预测值之间肯定存在差异，常用ε来表示。

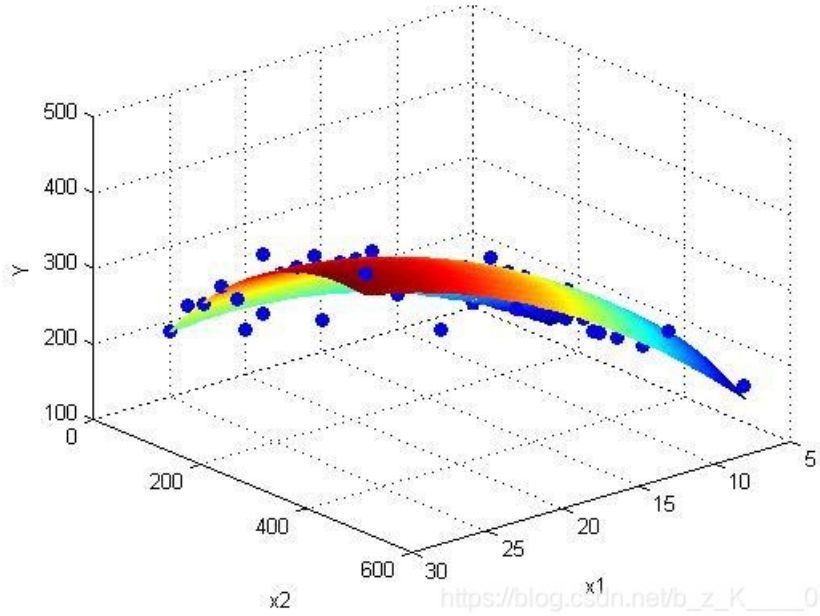
在线性回归推导的过程中，我们假设误差是独立同分布，并且服从均值为0，方差为σ的高斯分布（高斯分布就是正态分布）。
同样用一幅图片：



如果回归分析中包括两个或两个以上的自变量，且因变量和自变量之间是线性关系，则称为多元线性回归分析。就和小时候学的一元方程和多元方程一样，一元是只在一个维度的变化，多元则是多个维度的变化。表示如下：

$$Z = \beta Z*1 + \beta Z*2 + \ldots + \beta Z*k$$

β 被称为偏回归系数



说了这么多概念，我知道你们也不一定看，咱们来开始实验吧。本次实验将会使用sklearn工具包中的线性回归模型来完成波士顿房价预测。

1.2 实验目的

- [ ] 了解线性回归模型原理
- [ ] 学会数据集划分
- [ ] 学会数据标准化处理
- [ ] 学会使用sklearn中的线性回归模型进行训练、预测
- [ ] 查看、评估MSE、RMSE、MAE、r2_score

1.3 实验准备

服务器端：python3.6以上、numpy、pandas、sklearn、Jupyter Notebook
客户端：Google Chrome浏览器

二、 实验步骤

2.1 导入所需包

```
import pandas as pd
import numpy as np
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn import metrics
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error #平方绝对误差
```

2.2 加载数据集

波士顿地区房价数据获取，数据来自于sklearn自带数据集；

```
boston = datasets.load_boston()
print(boston.DESCR)      #获得关于房价的描述信息
x = boston.data      #获得数据集的特征属性列
y = boston.target      #获得数据集的label列
df = pd.DataFrame(data = np.c_[x,y],columns=np.append(boston.feature_names,['MEDV'])) #np.c_是按列连接两个矩阵，就是把两矩阵左右相加，要求列数相等
df = df[['RM','MEDV']]      #选择房间数属性列和房价属性列
print(df[:5])      #查看前5行的数据格式
```

先来把数据集的属性信息做一个显示：

翻译一下就是：按城镇划分的人均犯罪率，超过25,000平方英尺的住宅用地的比例，每个城镇的非零售商业用地比例，CHAS Charles River（这个我也不太清楚，但应该是0,1分布的一个值），氮氧化物浓度(千万分之一)，每个住宅的平均房间数，1940年以前建造的业主自住单位的业主的年龄比例，到五个波士顿就业中心的加权距离，放射状公路可达性指数，每10,000美元的全价值财产税税率，按城镇划分的学生-教师比例，b1000 (Bk - 0.63)^2，其中Bk是城镇黑人的比例
低的人口出生率，中值在1000美元的业主自住房屋。
（跟咱们国家看中的也差不多，是不是学区房，是不是商圈，几室几厅，附近有没有不良青年之类的~）

```
:Number of Attributes: 13 numeric/categorical predictive. Median Value (attribute 14) is usually the target.

:Attribute Information (in order):
    - CRIM     per capita crime rate by town
    - ZN       proportion of residential land zoned for lots over 25,000 sq.ft.
    - INDUS    proportion of non-retail business acres per town
    - CHAS     Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
    - NOX      nitric oxides concentration (parts per 10 million)
    - RM       average number of rooms per dwelling
    - AGE      proportion of owner-occupied units built prior to 1940
    - DIS      weighted distances to five Boston employment centres
    - RAD      index of accessibility to radial highways
    - TAX      full-value property-tax rate per $10,000
    - PTRATIO  pupil-teacher ratio by town
    - B        1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
    - LSTAT    % lower status of the population
    - MEDV     Median value of owner-occupied homes in $1000's
```


运行结果前五行的数据如图

	RM	MEDV
0	6.575	24.0
1	6.421	21.6
2	7.185	34.7
3	6.998	33.4
4	7.147	36.2

2.2 波士顿地区房价数据分割

train_test_split函数用于将矩阵随机划分为训练子集和测试子集，并返回划分好的训练集测试集样本和训练集测试集标签。

格式：

X_train,X_test, y_train, y_test =cross_validation.train_test_split(train_data,train_target,test_size=0.3, random_state=0)

参数解释：

train_data：被划分的样本特征集

train_target：被划分的样本标签

test_size：如果是浮点数，在0-1之间，表示样本占比；如果是整数的话就是样本的数量

random_state：是随机数的种子。

随机数种子：其实就是该组随机数的编号，在需要重复试验的时候，保证得到一组一样的随机数。比如你每次都填1，其他参数一样的情况下你得到的随机数组是一样的。但填0或不填，每次都会不一样。

随机数的产生取决于种子，随机数和种子之间的关系遵从以下两个规则：

种子不同，产生不同的随机数；

种子相同，即使实例不同也产生相同的随机数。

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)    #划分数据集
```

2.3 训练与测试数据标准化处理

StandardScaler类是一个用来讲数据进行归一化和标准化的类，且是针对每一个特征维度来做的，而不是针对样本。
所谓归一化和标准化，即应用下列公式：
[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-9V6w9PvT-1589334964113)
(attachment:image.png)]
使得新的X数据集方差为1，均值为0，fit_transform方法是fit和transform的结合，fit_transform(X_train) 意思是找出X_train的μ和σ，并应用在X_train上。这时对于X_test，我们就可以直接使用transform方法。因为此时StandardScaler已经保存了X_train的μ和σ。

```
scaler = StandardScaler()      #作用：去均值和方差归一化。可保存训练集中的均值、方差参数，然后直接用于转换测试集数据。
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
```

2.4 使用线性回归模型进行训练并预测得分。

调用sklearn中LR模型并训练模型，用score()函数对模型进行打分。注：最好的得分为1.0，一般的得分都比1.0低，得分越低代表结果越差。

```
linreg = LinearRegression()
model = linreg.fit(x_train,y_train)

print(model.score(x_test,y_test))
```

好像这个得分确实不太高

```
print(model.score(x_test,y_test))
```

```
0.655877712671
```

2.5 查看MSE、RMSE、MAE、r2_score

为了获得对模型性能的无偏估计，在训练过程中使用未知数据对测试进行评估是至关重要的。所以，需要将数据集划分为训练数据集和测试数据集，前者用于模型的训练，后者用于模型在未知数据上泛化性能的评估。
以下有sklearn中各种衡量指标介绍：

对模型性能进行定量估计的方法称为均方误差（Mean Squared Error, MSE）, 它是线性回归模型拟合过程中，最小化误差平方和（SSE）代价函数的平均值。
公式如下：

$$\frac{1}{m}\sum_{i=1}^{m}(y_i-\hat{y}_i)^2$$

这里的y是测试集上的。

用 真实值-预测值 然后平方之后求和平均。

猛着看一下这个公式是不是觉得眼熟，这不就是线性回归的损失函数嘛！！！ 对，在线性回归的时候我们的目的就是让这个损失函数最小。那么模型做出来了，我们把损失函数丢到测试集上去看看损失值不就好了嘛。简单直观暴力！

```
print("MSE均方误差：",mean_squared_error(y_train,model.predict(x_train)))
print("RMSE均方根误差：",mean_squared_error(y_train,model.predict(x_train)) ** 0.5)
```

RMSE（Root Mean Squard Error）均方根误差。

$$\sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i-\hat{y}_i)^2}$$

这不就是MSE开个根号么。有意义么？其实实质是一样的。只不过用于数据更好的描述。
例如：要做房价预测，每平方是万元（真贵），我们预测结果也是万元。那么差值的平方单位应该是 千万级别的。那我们不太好描述自己做的模型效果。怎么说呢？我们的模型误差是 多少千万？。。。。。。于是干脆就开个根号就好了。我们误差的结果就跟我们数据是一个级别的可，在描述模型的时候就说，我们模型的误差是多少万元。

MAE(平均绝对误差)

$$\frac{1}{m}\sum_{i=1}^{m}\left|(y_i-\hat{y}_i)\right|$$

print(“MAE平均绝对误差：”,mean_absolute_error(y_train,model.predict(x_train)))

$$R2=1-\frac{\sum_i(\hat{y}_i-y_i)^2}{\sum_i(\bar{y}_i-y_i)^2}$$

其中，分子部分表示真实值与预测值的平方差之和，类似于均方差 MSE；分母部分表示真实值与均值的平方差之和，类似于方差 Var。

根据 R-Squared 的取值，来判断模型的好坏，其取值范围为[0,1]：

如果结果是 0，说明模型拟合效果很差；

如果结果是 1，说明模型无错误。

一般来说，R-Squared 越大，表示模型拟合效果越好。R-Squared 反映的是大概有多准，因为，随着样本数量的增加，R-Square必然增加，无法真正定量说明准确程度，只能大概定量。

小结

写了这么多有点累，但做学习分享还是很开心的。
找了半天适合结尾的笔记，延续一下每日一句。

第一项规则：在辩论中，获得最大利益的唯一方法，就是避免辩论。 第二项规则：尊重别人的意见，永远别指责对方是错的。 第三项规则： 如果你错了，迅速、郑重的承认下来。 第四项规则： 以友善的方法开始。 第五项规则： 使对方很快的回答“是！是”。 第六项规则：尽量让别人有多说话的机会。 第七项规则：使对方以为这是他的意念。 第八项规则： 要真诚的从他人的观点去看事情。 第九项规则： 同情对方的意念和欲望。 第十项规则： 激发更高尚的动机。
————————《人性的弱点》戴尔.卡耐基