《python数据分析与挖掘实战》笔记第2章

发布于2021-03-02 20:53:54阅读 1550

**文章目录**

* + [第2章：python数据分析简介](https://cloud.tencent.com/developer?from=10680)
    - [2.2、python使用入门](https://cloud.tencent.com/developer?from=10680)
      * [2.2.3、数据结构](https://cloud.tencent.com/developer?from=10680)
        + [(1)列表/元组](https://cloud.tencent.com/developer?from=10680)
        + [(2)字典](https://cloud.tencent.com/developer?from=10680)
        + [(3)集合](https://cloud.tencent.com/developer?from=10680)
        + [(4)函数式编程](https://cloud.tencent.com/developer?from=10680)
      * [2.2.4、库的导入与添加](https://cloud.tencent.com/developer?from=10680)
    - [2.3、python数据分析工具](https://cloud.tencent.com/developer?from=10680)
      * [2.3.1、numpy](https://cloud.tencent.com/developer?from=10680)
      * [2.3.2、scipy](https://cloud.tencent.com/developer?from=10680)
      * [2.3.3、matplotlib](https://cloud.tencent.com/developer?from=10680)
      * [2.3.4、pandas](https://cloud.tencent.com/developer?from=10680)
      * [2.3.5、statsmodels](https://cloud.tencent.com/developer?from=10680)
      * [2.3.6、scikit-learn](https://cloud.tencent.com/developer?from=10680)
      * [2.3.7、keras](https://cloud.tencent.com/developer?from=10680)
      * [2.3.8、gensim](https://cloud.tencent.com/developer?from=10680)

**第2章：python数据分析简介**

**2.2、python使用入门**

**2.2.3、数据结构**

python有4个内建的数据结构–list（列表）、tuple（元组）、dictionary（字典）以及set（集合），它们可以统称为[容器](https://cloud.tencent.com/product/tke?from=10680)。

**(1)列表/元组**

从功能上看，列表与元组的区别是，列表可以被修改，而元组不可以。 如果已经有了一个列表a，同时想复制a，命名为变量b，那么b==a是无效的，这个时候b仅仅是a的一个别名（或者说引用），修改b也会修改a的。正确的复制方法应该是b==a[:]

* 表2-1 列表/元组相关的函数

| 函数 | 功能 | 函数 | 功能 |
| --- | --- | --- | --- |
| cmp(a,b) | 比较两个列表/元组的元素 | min(a) | 返回列表/元组元素最小值 |
| len(a) | 列表/元组元素个数 | sum(a) | 将列表/元组中的元素求和 |
| max(a) | 返回列表/元组元素最大值 | sorted(a) | 对列表的元素进行升序排序 |

* 表2-2列表相关的方法

| 函 数 | 功 能 |
| --- | --- |
| a.append(1) | 将1添加到列表a末尾 |
| a.count(1) | 统计列表a中元素1出现的次数 |
| a.extend([1, 2]) | 将列表[1, 2]的内容追加到列表a的末尾中 |
| a.index(1) | 从列表a中找出第一个1的索引位置 |
| a.insert(2, 1) | 将1插入列表a的索引为2的位置 |
| a.pop(1) | 移除列表a中索引为1的元素 |

**(2)字典**

dict([['today',20],['tomorrow',30]]) #也相当于{'today':20, 'tomorrow':30 }

dict.fromkeys(['today', 'tomorrow'],20) #相当于{'today':20, 'tomorrow':20}

复制

**(3)集合**

Python内置了集合这一数据结构，同数学上的集合概念基本上是一致的，它与列表的区 别在于：1.它的元素是不重复的，而且是无序的；2.它不支持索引。一般我们通过花括号{} 或者set()函数来创建一个集合。

s = (1, 2, 2, 3} #注意2会自动去重，得到{1, 2, 3}

s = set([1, 2, 2, 3]) #同样，它将列表转换为集合，得到{1, 2, 3}

复制

由于集合的特殊性(特别是无序性)，因此集合有一些特别的运算。

a = t | s # t和s的并集

b = t & s #t和s的交集

c = t - s #求差集(项在t中，但不在s中)

d = t^s #对称差集(项在t或s中，但不会同时出现在二者中)

复制

**(4)函数式编程**

**在python中，函数式编程主要由几个函数的使用构成：lambda()、map()、reduces()、filter()，**

f=lambda x : x+2 #定义函数f(x)=x+2

g=lambda x,y : x+y # 定义函数g(x,y)=x+y

a=[1,2,3]

b=map(lambda x: x+2,a)

b=list(b) # 结果是[3,4,5]

复制

在3.x需要b=list(b)这一步，在2.x不需要这步，原因是在3.x中，map函数仅仅是创建一个待运行的命令容器，只有其它函数调用它的时候才返回结果。

**reduce()函数用于递归计算**

reduce(lambda x,y: x\*y,range(1,n+1))

等价于

s=1

for i in range(1,n+1):

s=s\*i

复制

在2.x中上述命令可以直接运行，在3.x中reduce函数已经被移除了全局命名空间，它被置于fuctools库中，如需使用，则需要通过 from fuctools import reduce

**filter()函数是一个过滤器**

b=filter(lambda x :x>5 and x<8,range(10))

b=list(b) # 结果是[6,7]

复制

**2.2.4、库的导入与添加**

（1）库的导入

import math

math.sin (1) #计算正弦

math.exp (1) #计算指数

math.pi #内置的圆周率常数

import math as m

m.sin (1) #计算正弦

from math import exp as e #只导入math库中的exp函数，并起别名e

e(1) #计算指数

sin (1) #此时sin (1)和math.sin(l)都会出错，因为没被导入

复制

（2）导入 future 特征(For 2.x) Python 2.x与3.x之间的差别不仅仅在内核上，也表现在代码的实现中。比如，在2.x 中，print是作为一个语句出现的，用法为print a :但是在3.x中，它是作为函数出现的，用 法为print(a)。为了保证兼容性，本书的基本代数是使用3.x的语法编写的，而使用2.x的读 者，可以通过引入fbture特征的方式兼容代码，如，

#将print变成函数形式，即用print (a)格式输出

from \_\_future\_\_ import print\_function

#3.x的3/2=1.5, 3//2才等于 1; 2.x中3/2=1

from \_\_future\_\_ import division

复制

**2.3、python数据分析工具**

**表2-4 Python数据挖掘相关扩展库**

| 扩展库 | 简 介 |
| --- | --- |
| Numpy | 提供数组支持，以及相应的高效的处理函数 |
| Scipy | 提供矩阵支持，以及矩阵相关的数值计算模块 |
| Matplotlib | 强大的数据可视化工具、作图库 |
| Pandas | 强大、灵活的数据分析和探索工具 |
| StatsModels | 统计建模和计量经济学，包括描述统计、统计模型估计和推断 |
| Scikit-Leam | 支持回归、分类、聚类等的强大的机器学习库 |
| Keras | 深度学习库，用于建立神经网络以及深度学习模型 |
| Gensim | 用来做文本主题模型的库，文本挖掘可能用到 |

**2.3.1、numpy**

pip install numpy

复制

代码清单2-1，numpy基本操作

# -\*- coding: utf-8 -\*-

import numpy as np #一般以np作为numpy的别名

a = np.array([2, 0, 1, 5]) #创建数组

print(a) #输出数组

print(a[:3]) #引用前三个数字（切片）

print(a.min()) #输出a的最小值

a.sort() #将a的元素从小到大排序，此操作直接修改a，因此这时候a为[0, 1, 2, 5]

print(a)

b= np.array([[1, 2, 3], [4, 5, 6]]) #创建二维数组

print(b\*b) #输出数组的平方阵，即[[1, 4, 9], [16, 25, 36]]

复制

参考链接：http://www.numpy.org/ http://reverland.org/python/2012/08/22/numpy/

**2.3.2、scipy**

scipy包含的功能有最优化、线性代数、积分、插值、拟合、特殊函数、快速傅里叶变化、信号处理和[图像处理](https://cloud.tencent.com/product/tiia?from=10680)、常微分方程求解和其它科学与工程中常用的计算，这些功能都是数据挖掘和建模必备的。

scipy依赖于numpy

pip install scipy

复制

代码清单2-2，scipy求解非线性方程组和数值积分

# -\*- coding: utf-8 -\*-

#求解非线性方程组2x1-x2^2=1,x1^2-x2=2

from scipy.optimize import fsolve #导入求解方程组的函数

def f(x): #定义要求解的方程组

x1 = x[0]

x2 = x[1]

return [2\*x1 - x2\*\*2 - 1, x1\*\*2 - x2 -2]

result = fsolve(f, [1,1]) #输入初值[1, 1]并求解

print(result) #输出结果，为array([ 1.91963957, 1.68501606])

#数值积分

from scipy import integrate #导入积分函数

def g(x): #定义被积函数

return (1-x\*\*2)\*\*0.5

pi\_2, err = integrate.quad(g, -1, 1) #积分结果和误差

print(pi\_2 \* 2) #由微积分知识知道积分结果为圆周率pi的一半

复制

参考链接：http://www.scipy.org/ http://reverland.org/python/2012/08/22/scipy/

**2.3.3、matplotlib**

pip install matplotlib

复制

代码清单2-3，matplotlip作图的基本代码

# -\*- coding: utf-8 -\*-

import numpy as np

import matplotlib.pyplot as plt #导入Matplotlib

x = np.linspace(0, 10, 1000) #作图的变量自变量

y = np.sin(x) + 1 #因变量y

z = np.cos(x\*\*2) + 1 #因变量z

plt.figure(figsize = (8, 4)) #设置图像大小

plt.plot(x,y,label = '$\sin x+1$', color = 'red', linewidth = 2) #作图，设置标签、线条颜色、线条大小

plt.plot(x, z, 'b--', label = '$\cos x^2+1$') #作图，设置标签、线条类型

plt.xlabel('Time(s) ') # x轴名称

plt.ylabel('Volt') # y轴名称

plt.title('A Simple Example') #标题

plt.ylim(0, 2.2) #显示的y轴范围

plt.legend() #显示图例

plt.show() #显示作图结果

复制

如果读者使用的是中文标签，就会发现中文标签无法正常显示。这是由于Matplotlib的默认字体是英文字体所致，解决它的办法是在作图之前手动指定默认字体为中文字体，如黑体（SimHei）。

plt.rcParams['font.sans-serif'] = ['SimHei'] # 这两句用来正式显示中文标签

复制

另外。保存作图图像时，负号有可能显示不正常，可以通过以下代码解决：

plt.rcParams['axes.unicode\_minus'] = False # 解决保存图像是负号'-'显示为方块的问题

复制

这里有一个小建议给读者：有时间多去Matplotlib提供的“画廊”欣赏它做出来漂亮效果，也许你就慢慢的爱上Matplotlib作图了。 画廊：http://matplotlib.org/gallery.html

参考链接：http://www.matplotlib.org/ http://reverland.org/python/2012/09/07/matplotlib-tutorial/

**2.3.4、pandas**

pip install pandas

复制

pandas是python下最强大的数据分析和探索工具，pandas的名称来自于面板数据（Panel Data）和python数据分析（Data Analysis）。

pip install xlrd # 为python添加读取excel的功能

pip install xlwt # 为python添加写入excel的功能

复制

pandas基本的数据结构是Series和DataFrame。Series就是序列，类似一维数组；DataFrame则是相当于一张二维的表格，类似二维数组，它的每一列都是一个Series。

代码清单2-4，pandas的简单例子

# -\*- coding: utf-8 -\*-

import pandas as pd #通常用pd作为pandas的别名。

s = pd.Series([1,2,3], index=['a', 'b', 'c']) #创建一个序列s

d = pd.DataFrame([[1, 2, 3], [4, 5, 6]], columns = ['a', 'b', 'c']) #创建一个表

d2 = pd.DataFrame(s) #也可以用已有的序列来创建表格

d.head() #预览前5行数据

d.describe() #数据基本统计量

#读取文件，注意文件的存储路径不能带有中文，否则读取可能出错。

pd.read\_excel('data.xls') #读取Excel文件，创建DataFrame。

pd.read\_csv('data.csv', encoding = 'utf-8') #读取文本格式的数据，一般用encoding指定编码。

复制

参考链接：http://pandas.pydata.org/pandas-docs/stable/ http://jingyan.baidu.com/season/43456

**2.3.5、statsmodels**

pip install statsmodels

复制

pandas着眼于数据的读取、处理和探索，而statsmodels则更注重数据的统计建模分析，它使得python有了R语言的味道。statsmodels支持与pandas进行数据交互，因此，它与pandas结合，成为了python下强大的数据挖掘组合。

代码清单2-5，用statsmodels来进行ADF平稳性检验的例子

# -\*- coding: utf-8 -\*-

from statsmodels.tsa.stattools import adfuller as ADF # 导入ADF检验

import numpy as np

result=ADF(np.random.rand(100)) #返回的结果有ADF值、p值等

print(result)

复制

**2.3.6、scikit-learn**

pip install scikit-learn

复制

scikit-learn是python下强大的学习工具包，它提供了完善的机器学习工具箱，包括数据预处理、分类、回归、聚类、预测和模型分析等。 scikit-learn依赖于numpy、scipy和matplotlib。

代码清单2-6，创建一个简单的机器学习模型

# -\*- coding: utf-8 -\*-

from sklearn.linear\_model import LinearRegression # 导入线性回归模型

model=LinearRegression()# 建立线性回归模型

print(model)

复制

1）所有模型提供的接口有： model.fit()：训练模型，对于监督模型来说是fit(X,y)，对于非监督模型是fit(X)。

2）监督模型提供的接口有： model.predict(X\_new)：预测新样本 model.predict\_proba(X\_new)：预测概率，仅对某些模型有用（比如LR） model.score()：得分越高，fit越好

3）非监督模型提供的接口有： model.transform()：从数据间学到新的“基空间”。 model.fit\_transform()：从数据中学到新的基并将这个数据按照这组“基”进行转换。

scikit-learn本身提供一些实例数据，比较常见的有安德森鸢尾花卉数据集、手写图像数据集等。

代码清单2-7，创建一个SVM模型

# -\*- coding: utf-8 -\*-

from sklearn import datasets # 导入数据集

iris=datasets.load\_iris() # 加载数据集

print(iris.data.shape) # 查看数据集大小

from sklearn import svm # 导入SVM模型

clf=svm.LinearSVC() # 建立线性SVM分类器

clf.fit(iris.data,iris.target) # 用数据训练模型

result=clf.predict([[5.0,3.6,1.3,0.25]]) # 训练好模型之后，输入新的数据进行预测

print(result)

params=clf.coef\_ # 查看训练好模型的参数

print(params)

复制

参考链接：http://scikit-learn.org/

**2.3.7、keras**

虽然scikit-learn足够强大，但是它并没有包含一种强大的模型–人工神经网络。

本书用keras库搭建神经网络。事实上，keras并非简单的神经网络库，而是一个基于Theano的强大的深度学习库，利用它不仅仅可以搭建普通的神经网络，还可以搭建各种深度学习模型，如自编码器、循环神经网络、递归神经网络、卷积神经网络等特点，最重要的是它还实现了[GPU](https://cloud.tencent.com/product/gpu?from=10680)加速，使得密集型数据的处理速度是CPU的数十倍。

有必要介绍一下Theano，它也是python的一个库，用来定义、优化和高效的解决多维数组数据对应数学表达式的模拟估计问题。它具有高效地实现符号分解、高度优化的速度和稳定性等特点，最重要的是它还实现了GPU加速，使得密集型数据的处理速度是CPU的数十倍。

用Theano就可以搭建起高效的神经网络模型，但是对于普通读者来说门槛还是相当高的，keras正是为此而生，它大大简化了搭建各种神经网络模型的步骤，允许普通用户轻松的搭建并求解具有几百个输入节点的深层神经网络，而且定制的自由度非常大，甚至可以惊呼：搭建神经网络可以如此简单！

安装keras之前需要先安装numpy，scipy，theano。安装theano先要准备一个C++编译器，这在linux下是自带的。因此linux下安装theano和keras非常简单，在windows下就没有那么简单了，因为它没有现成的编译环境。一般而言是先安装MinGW（windows下的GCC和G++）然后在安装Theano（提前安装好numpy等依赖库），最后安装keras。如果要实现GPU加速，还需要安装和配置CUDA。

pip install keras

pip install theano

复制

值得一提的是，在windows下keras的速度会大大折扣。

参考链接：http://deeplearing.net/software/theano/install.html#install http://github.com/fchollet/keras

代码清单2-8，搭建一个MLP（多层感知器）

# -\*- coding: utf-8 -\*-

from keras.models import Sequential

from keras.layers.core import Dense,Dropout,Activation

from keras.optimizers import SGD

model=Sequential() # 模型初始化

model.add(Dense(20,64)) # 添加输入层（20节点）、第一隐藏层（64节点）的连接

model.add(Activation('tanh')) # 第一隐藏层用tanh作为激活函数

model.add(Dropout(0.5)) # 使用Dropout防止过拟合

model.add(Dense(64,64)) # 添加第一隐藏层（64节点）、第二隐藏层（64节点）的连接

model.add(Activation('tanh')) # 第二隐藏层用tanh作为激活函数

model.add(Dropout(0.5)) # 使用Dropout防止过拟合

model.add(Dense(64,1)) # 添加第二隐藏层（64节点）、输出层（1节点）的连接

model.add(Activation('sigmoid')) # 输出层用sigmoid作为激活函数

sgd=SGD(lr=0.1,decay=1e-6,momentum=0.9,nesterov=True) #定义求解算法

model.compile(loss='mean\_squared\_error',optimizer=sgd) # 编译生成模型，损失函数为平均误差平方和

X\_train=None

y\_train=None

X\_test=None

y\_test=None

model.fit(X\_train,y\_train,nb\_epoch=20,batch\_size=16) # 训练模型

score=model.evaluate(X\_test,y\_test,batch\_size=16) # 测试模型

复制

要注意的是，keras的预测函数与scikit-learn有所差别，keras用model.predict()方法给出概率，model.predict\_classes()方法给出分类结果。

参考链接：http://radimrehurek.com/gensim/ http://www.52nlp.cn/ (如何计算两个文档的相似度二)

**2.3.8、gensim**

gensim是用来处理语言方面的任务，如文本相识度计算、LDA、word2vec等

代码清单2-9，gensim使用word2vec的简单例子

# -\*- coding: utf-8 -\*-

import gensim,logging

# logging是用来输出训练日志的

logging.basicConfig(format='%(asctime)s:%(levelname)s:%(message)s',level=logging.INFO)

# 分好词的句子，每个句子以词列表的形式输入

sentences=[['first','sentence'],['second','sentence']]

# 用以上句子训练词向量模型

model=gensim.models.Word2Vec(sentences,min\_count=1)

print(model['sentence']) # 输出单词sentence的词向量

复制

参考链接：http://radimrehurek.com/gensim/ http://www.52nlp.cn/ (如何计算两个文档的相似度二)

本文参与[腾讯云自媒体分享计划](https://cloud.tencent.com/developer/support-plan) ，欢迎热爱写作的你一起参与！

本文分享自作者个人站点/博客：https://blog.csdn.net/xc\_zhou/[复制](javascript:void(0);)

如有侵权，请联系 cloudcommunity@tencent.com 删除。

[展开阅读全文](javascript:;)

[HTTP](https://cloud.tencent.com/developer/tag/10620?entry=article)[Keras](https://cloud.tencent.com/developer/tag/10730?entry=article)[神经网络](https://cloud.tencent.com/developer/tag/10332?entry=article)

[举报](javascript:;)

点赞 1分享

[**登录**](javascript:;) 后参与评论

0 条评论

相关文章

* 《python数据分析与挖掘实战》笔记第5章

分类和预测是预测问题的两种主要类型，分类主要是预测分类标号（离散属性），而预测 主要是建立连续值函数模型，预测给定自变量对应的因变量的值。

[周小董](https://cloud.tencent.com/developer/user/4908043)

* 《python数据分析与挖掘实战》笔记第4章

数据预处理一方面是要提高数据的质量，另一方面是要让 数据更好地适应特定的挖掘技术或工具。统计发现，在数据挖掘的过程中，数据预处理工作量占到了整个过程的60%。

[周小董](https://cloud.tencent.com/developer/user/4908043)

* 《python数据分析与挖掘实战》笔记第3章

数据质量分析是数据挖掘中数据准备过程的重要一环，是数据预处理的前提，也是数据挖掘分析结论有效性和准确性的基础，没有可信的数据，数据挖掘构建的模型将是空中楼阁。

[周小董](https://cloud.tencent.com/developer/user/4908043)

* 《python数据分析与挖掘实战》笔记第1章

数据挖掘是从大量数据（包括文本）中挖掘出隐含的、先前未知的、对决策有潜在价值的关系、模式和趋势，并用这些知识和规则建立用于决策支持的模型，提供预测性决策支持的方...

[周小董](https://cloud.tencent.com/developer/user/4908043)

* Python数据分析与实战挖掘

基础篇 书推荐：《用python做科学计算》 ? 扩展库 简介 Numpy数组支持，以及相应的高效处理函数 Scipy矩阵支持，以及相应的矩阵数值计算模块 Ma...

[企鹅号小编](https://cloud.tencent.com/developer/user/1008345)

* Python数据分析与挖掘实战.pdf

[学一学大数据](https://cloud.tencent.com/developer/user/1685822)

* 关于《Python数据挖掘入门与实战》读书笔记二（亲和性分析）

首先确保最新的版本吧，python最让人头痛和无解的就是版本兼容性问题，不过既然选择了就不要怕麻烦，不断的纠错不断的尝试，在频繁的错误中提高自己。

[python与大数据分析](https://cloud.tencent.com/developer/user/7190337)

* 关于《Python数据挖掘入门与实战》读书笔记七（主成分分析二）

主成分分析算法（Principal Component Analysis，PCA）的目的是找到能用较少信息描述数据集的特征组合。它意在发现彼此之间没有相关性、能...

[python与大数据分析](https://cloud.tencent.com/developer/user/7190337)

* Python业务分析实战｜共享单车数据挖掘

自行车共享系统是传统自行车租赁的新一代，从注册会员、租赁到归还的整个过程都是自动化的。通过这些系统，用户可以很容易地从一个特定的位置租用自行车，并在另一个位置归...

[数据STUDIO](https://cloud.tencent.com/developer/user/8756457)

* 关于《Python数据挖掘入门与实战》读书笔记六（主成分分析一）

特征抽取是数据挖掘任务最为重要的一个环节，一般而言，它对最终结果的影响要高过数据挖掘算法本身。但怎样选取好的特征，还没有严格、快捷的规则可循，这也是数据挖掘科学...

[python与大数据分析](https://cloud.tencent.com/developer/user/7190337)

* Python数据分析与挖掘实战 (大数据技术丛书) - 张良均 等著

10余位数据挖掘领域资深专家和科研人员，10余年大数据挖掘咨询与实施经验结晶。从数据挖掘的应用出发，以电力、航空、医疗、互联网、生产制造以及公共服务等行业真实案...

[用户3157710](https://cloud.tencent.com/developer/user/3157710)

* Python 爬取淘宝商品数据挖掘分析实战

? 作者 孙方辉 本文为CDA志愿者投稿作品，转载需授权 项目内容 本案例选择>> 商品类目：沙发； 数量：共100页 4400个商品； 筛选条件：天猫...

[CDA数据分析师](https://cloud.tencent.com/developer/user/1283928)

* 关于《Python数据挖掘入门与实战》读书笔记一（数据挖掘、机器学习、深度学习、人工智能）

Python数据挖掘、Python机器学习、Python深度学习的书籍买了不少本了，但真正读下来的却很少，为何？涉及高等数学，图论、离散数学等等，对我来说和天文...

[python与大数据分析](https://cloud.tencent.com/developer/user/7190337)

* 慕课网数据分析与挖掘实战（一）-数据获取

其中，urllib，urllib2，requests，scrapy主要用于数据抓取 xpath主要用于分析提取有价值的内容 phantomjs，beauti...

[Meet相识](https://cloud.tencent.com/developer/user/3090887)

* 《数据挖掘与数据化运营实战 思路、方法、技巧与应用》—— 读书笔记

[用户1154259](https://cloud.tencent.com/developer/user/1154259)

* 关于《Python数据挖掘入门与实战》读书笔记三（估计器和近邻算法）

Scikit-learn库，实现了一系列数据挖掘算法，提供通用编程接口、标准化的测试和调参工具，便于用户尝试不同算法对其进行充分测试和查找优参数值。

[python与大数据分析](https://cloud.tencent.com/developer/user/7190337)