Python机器学习重要的库

数据分析与数据挖掘

重要的Python库

- **NumPy：**Numpy（Numerical Python的简称）是Python科学计算的基础包。
  - 快速高效的多维数组对象ndarray
  - 用于对数组执行元素级计算以及直接对数组执行数学运算的函数
  - 用于读写硬盘上基于数组的数据集的工具
  - 线性代数运算、傅里叶变换、以及随机数生成
  - 用于将C、C++、Fortran代码集成到Python的工具

  除了为Python提供快速的数组处理能力，NumPy在数据分析方面还作为在算法之间传递数据的容器

- **Pandas：**pandas提供了使我们能够快速便捷地处理结构化数据的大量数据结构和函数
  - Pandas兼具NumPy高性能的数据计算功能以及电子表格和关系型数据库（如SQL）灵活的数据处理功能。提供了复杂精细的索引功能、一边更为便捷地完成重塑、切片和切块、聚合以及选取数据子集等操作。
  - Pandas提供了大量适用于金融数据的高性能时间序列功能和工具

其他的Python库

- **Matplotlib：**matplotlib是最流行的用于绘制数据图标的Python库，实现数据可视化。
- **IPython：**Ipython是Python科学计算标准工具集的组成部分，它将其他所有的东西联系到了一起，为交互是和探索式计算提供了一个强健而高效的环境。
- **SciPy：**SciPy是一组专门解决科学计算中各种标准问题域的包的集合，包括
  - Scipy.integrate：数值积分例程和韦恩方程式求解器
  - Scipy.linalg：扩展了由numpy.linalg提供的线性代数例程和矩阵分解功能
  - Scipy.optimize：函数优化器（最小化器）以及根查找算法
  - Scipy.signal：信号处理工具
  - Scipy.sparse：稀疏矩阵和系数线性系统求解器
  - Scipy.special：SPECFUN（这是一个实现了许多常用数学函数（如伽马函数）的Fortran库）的包装器
  - Scipy.stats：标准连续的离散概率分布（如密度函数、采样器、连续分布函数等）、各种统计检验方法，以及更好的描述统计法。
  - Scipy.weave：利用内敛C++代码加速数组计算的工具

常用第三方Python库安装和导入

- 安装
  - 下载并安装**Anaconda**，它附带了预安装的库。

    Anaconda Python 是完全免费、跨平台、企业级的Python发行大规模数据处理、预测分析和科学计算工具。
  - PyCharm已经集成NumPy、Pandas、Matplotlib等常用库。
- 导入
  - import numpy as np
  - from socket import gethostname, socket

3.1 NumPy库介绍

- NumPy是高性能科学计算和数据分析的基础包。部分功能如下：
  - ndarray, 具有矢量算术运算和复杂广播能力的快速且节省空间的多维数组。
  - 用于对整组数据进行快速运算的标准数学函数（无需编写循环）。
  - 用于读写磁盘数据的工具以及用于操作内存映射文件的工具。
  - 线性代数、随机数生成以及傅里叶变换功能。
  - 用于集成C、C++、Fortran等语言编写的代码的工具。
- 首先要导入numpy库：import numpy as np

生成函数	作用
np.array(x)	将输入数据转化为一个ndarray
np.array(x, dtype)	将输入数据转化为一个类型为type的ndarray
np.asarray(array)	将输入数据转化为一个新的（copy）ndarray
np.ones(N)	生成一个N长度的一维全一ndarray
np.ones(N, dtype)	生成一个N长度类型是dtype的一维全一ndarray
np.ones_like(ndarray)	生成一个形状与参数相同的全一ndarray
np.zeros(N)	生成一个N长度的一维全零ndarray
np.zeros(N, dtype)	生成一个N长度类型位dtype的一维全零ndarray
np.zeros_like(ndarray)	类似np.ones_like(ndarray)
np.empty(N)	生成一个N长度的未初始化一维ndarray
np.empty(N, dtype)	生成一个N长度类型是dtype的未初始化一维ndarray
np.empty(ndarray)	类似np.ones_like(ndarray)
np.eye(N) np.identity(N)	创建一个N * N的单位矩阵（对角线为1，其余为0）
np.arange(num)	生成一个从0到num-1步数为1的一维ndarray
np.arange(begin, end)	生成一个从begin到end-1步数为1的一维ndarray
np.arange(begin, end, step)	生成一个从begin到end-step的步数为step的一维ndarray
np.in1d(ndarray, [x,y,...])	检查ndarray中的元素是否等于[x,y,...]中的一个，返回bool数组

矩阵函数	说明
np.diag(ndarray)	以一维数组的形式返回方阵的对角线（或非对角线）元素
np.diag([x,y,...])	将一维数组转化为方阵（非对角线元素为0）
np.dot(ndarray, ndarray)	矩阵乘法
np.trace(ndarray)	计算对角线元素的和

排序函数	说明
np.sort(ndarray)	排序，返回副本
np.unique(ndarray)	返回ndarray中的元素，排除重复元素之后，并进行排序
np.intersect1d(ndarray1, ndarray2)	返回二者的交集并排序。
np.union1d(ndarray1, ndarray2)	返回二者的并集并排序。
np.setdiff1d(ndarray1, ndarray2)	返回二者的差。
np.setxor1d(ndarray1, ndarray2)	返回二者的对称差

一元计算函数	说明
np.abs(ndarray)	计算绝对值
np.fabs(ndarray)	计算绝对值（非复数）
np.mean(ndarray)	求平均值
np.sqrt(ndarray)	计算x^0.5
np.square(ndarray)	计算x^2
np.exp(ndarray)	计算e^x
log、log10、log2、log1p	计算自然对数、底为10的log、底为2的log、底为(1+x)的log
np.sign(ndarray)	计算正负号：1（正）、0（0）、-1（负）
np.ceil(ndarray)	计算大于等于改值的最小整数
np.floor(ndarray)	计算小于等于该值的最大整数
np.rint(ndarray)	四舍五入到最近的整数，保留dtype
np.modf(ndarray)	将数组的小数和整数部分以两个独立的数组方式返回
np.isnan(ndarray)	返回一个判断是否是NaN的bool型数组
np.isfinite(ndarray)	返回一个判断是否是有穷（非inf，非NaN）的bool型数组
np.isinf(ndarray)	返回一个判断是否是无穷的bool型数组
cos、cosh、sin、sinh、tan、tanh	普通型和双曲型三角函数
arccos、arccosh、arcsin、arcsinh、arctan、arctanh	反三角函数和双曲型反三角函数
np.logical_not(ndarray)	计算各元素not x的真值，相当于-ndarray

多元计算函数	说明
np.add(ndarray, ndarray)	相加
np.subtract(ndarray, ndarray)	相减
np.multiply(ndarray, ndarray)	乘法
np.divide(ndarray, ndarray)	除法
np.floor_divide(ndarray, ndarray)	圆整除法（丢弃余数）
np.power(ndarray, ndarray)	次方
np.mod(ndarray, ndarray)	求模
np.maximum(ndarray, ndarray)	求最大值
np.fmax(ndarray, ndarray)	求最大值（忽略NaN ）
np.minimun(ndarray, ndarray)	求最小值
np.fmin(ndarray, ndarray)	求最小值（忽略NaN ）
np.copysign(ndarray, ndarray)	将参数2中的符号赋予参数1
np.greater(ndarray, ndarray)	>
np.greater_equal(ndarray, ndarray)	>=
np.less(ndarray, ndarray)	<
np.less_equal(ndarray, ndarray)	<=
np.equal(ndarray, ndarray)	==
np.not_equal(ndarray, ndarray)	!=
logical_and(ndarray, ndarray)	&
logical_or(ndarray, ndarray)	|
logical_xor(ndarray, ndarray)	^
np.dot(ndarray, ndarray)	计算两个ndarray的矩阵内积
np.ix_([x,y,m,n],...)	生成一个索引器，用于Fancy indexing(花式索引)

文件读写	说明
np.save(string, ndarray)	将ndarray保存到文件名为 [string].npy 的文件中（无压缩）
np.savez(string, ndarray1, ndarray2, ...)	将所有的ndarray压缩保存到文件名为[string].npy的文件中
np.savetxt(sring, ndarray, fmt, newline='\n')	将ndarray写入文件，格式为fmt
np.load(string)	读取文件名string的文件内容并转化为ndarray对象（或字典对象）
np.loadtxt(string, delimiter)	读取文件名string的文件内容，以delimiter为分隔符转化为ndarray

3.2 Pandas库介绍

- pandas 是基于NumPy 的一种工具，该工具是为了解决数据分析任务而创建的。Pandas 纳入了大量库和一些标准的数据模型，提供了高效地操作大型数据集所需的工具。pandas提供了大量能使我们快速便捷地处理数据的函数和方法。
- 使用方法
  - from pandas import Series, DataFrame
  - import pandas as pd

Series常用函数

函数	说明
Series([x,y,...])Series({'a':x,'b':y,...}, index=param1)	生成一个Series
Series.copy()	复制一个Series
Series.reindex([x,y,...], fill_value=NaN) Series.reindex([x,y,...], method=NaN) Series.reindex(columns=[x,y,...])	重返回一个适应新索引的新对象，将缺失值填充为fill_value 返回适应新索引的新对象，填充方式为method 对列进行重新索引
Series.drop(index)	丢弃指定项
Series.map(f)	应用元素级函数

排序函数	说明
Series.sort_index(ascending=True)	根据索引返回已排序的新对象
Series.order(ascending=True)	根据值返回已排序的对象，NaN值在末尾
Series.rank(method='average', ascending=True, axis=0)	为各组分配一个平均排名
df.argmax() df.argmin()	返回含有最大值的索引位置 返回含有最小值的索引位置

DataFrame常用函数

函数	说明
	构建DataFrame
DataFrame(dict, columns=dict.index, index=[dict.columnnum])	数据矩阵，还可以传入行标和列标
DataFrame(二维ndarray)	每个序列会变成DataFrame的一列。所有序列的长度必须相同
DataFrame(由数组、列表或元组组成的字典)	类似于“由数组组成的字典”
DataFrame(NumPy的结构化/记录数组)	每个Series会成为一列。如果没有显式制定索引，则各Series的索引会被合并成结果的行索引
DataFrame(由Series组成的字典)	各内层字典会成为一列。键会被合并成结果的行索引。
DataFrame(由字典组成的字典)	各项将会成为DataFrame的一行。索引的并集会成为DataFrame的列标。
DataFrame(字典或Series的列表)	类似于二维ndarray
DataFrame(由列表或元组组成的列表)	沿用DataFrame
DataFrame(DataFrame)	类似于二维ndarray，但掩码结果会变成NA/缺失值
DataFrame(NumPy的MaskedArray)	
	返回一个适应新索引的新对象，将缺失值填充为fill_value，最大填充量为limit
df.reindex([x,y,...], fill_value=NaN, limit)	返回适应新索引的新对象，填充方式为method
df.reindex([x,y,...], method=NaN)	同时对行和列进行重新索引，默认复制新对象。
df.reindex([x,y,...], columns=[x,y,...],copy=True)	
df.drop(index, axis=0)	丢弃指定轴上的指定项。

汇总统计函数	说明
df.count()	非NaN的数量
df.describe()	一次性产生多个汇总统计
df.min()	最小值
df.min()	最大值
df.idxmax(axis=0, skipna=True)	返回含有最大值的index的Series
df.idxmin(axis=0, skipna=True)	返回含有最小值的index的Series
df.quantile(axis=0)	计算样本的分位数
df.sum(axis=0, skipna=True, level=NaN)	返回一个含有求和小计的Series
df.mean(axis=0, skipna=True, level=NaN)	返回一个含有平均值的Series
df.median(axis=0, skipna=True, level=NaN)	返回一个含有算术中位数的Series
df.mad(axis=0, skipna=True, level=NaN)	返回一个根据平均值计算平均绝对离差的Series
df.var(axis=0, skipna=True, level=NaN)	返回一个方差的Series
df.std(axis=0, skipna=True, level=NaN)	返回一个标准差的Series
df.skew(axis=0, skipna=True, level=NaN)	返回样本值的偏度（三阶距）
df.kurt(axis=0, skipna=True, level=NaN)	返回样本值的峰度（四阶距）
df.cumsum(axis=0, skipna=True, level=NaN)	返回样本的累计和
df.cummin(axis=0, skipna=True, level=NaN)	返回样本的累计最大值
df.cummax(axis=0, skipna=True, level=NaN)	返回样本的累计最小值
df.cumprod(axis=0, skipna=True, level=NaN)	返回样本的累计积
df.diff(axis=0)	返回样本的一阶差分
df.pct_change(axis=0)	返回样本的百分比数变化

排序函数	说明
df.sort_index(axis=0, ascending=True) df.sort_index(by=[a,b,...])	根据索引排序

计算函数	说明
df.add(df2, fill_value=NaN, axist=1)	元素级相加，对齐时找不到元素默认用fill_value
df.sub(df2, fill_value=NaN, axist=1)	元素级相减，对齐时找不到元素默认用fill_value
df.div(df2, fill_value=NaN, axist=1)	元素级相除，对齐时找不到元素默认用fill_value
df.mul(df2, fill_value=NaN, axist=1)	元素级相乘，对齐时找不到元素默认用fill_value
df.apply(f, axis=0)	将f函数应用到由各行各列所形成的一维数组上
df.applymap(f)	将f函数应用到各个元素上
df.cumsum(axis=0, skipna=True)	累加，返回累加后的dataframe