Array数据分析

[控制control： 2](#_Toc22135)

[规划cvxpy： 2](#_Toc18945)

[矩阵numpy： 3](#_Toc23441)

[随机random： 5](#_Toc16776)

[傅里叶fft： 5](#_Toc25119)

[线代linalg： 5](#_Toc24975)

[分配ot： 6](#_Toc30097)

[表单pandas： 6](#_Toc8390)

[科学scipy： 7](#_Toc20200)

[优化optimize： 7](#_Toc17649)

[插值interpolate： 7](#_Toc29303)

[机器sklearn： 8](#_Toc18483)

[聚类cluster： 8](#_Toc17548)

[数据dataset： 9](#_Toc19847)

[集成ensemble： 9](#_Toc5234)

[线性linear\_model： 10](#_Toc17993)

[采样model\_selection： 10](#_Toc22637)

[多分类multiclass： 10](#_Toc14522)

[贝叶斯naive\_bayes： 11](#_Toc27288)

[近邻neighbors： 11](#_Toc22494)

[感知机neural\_network： 11](#_Toc21977)

[管道pipeline： 12](#_Toc24369)

[预处理preprocessing： 12](#_Toc31552)

[向量机svm： 12](#_Toc30970)

[决策树tree： 12](#_Toc23436)

[符号sympy： 13](#_Toc27720)

[可视visdom： 14](#_Toc28076)

[森林xgboost： 15](#_Toc30694)

控制control：

import control as ctrl

|  |  |  |
| --- | --- | --- |
| 传递函数： | | |
| tf('s') | | 微分因子 |
| tf(num, den, sys=None) | | 传递函数 |
| 实例方法 | pole() | 极点 |
| zero() | 零点 |
| damp() | 特征参数、特征根 |

|  |  |  |
| --- | --- | --- |
| step\_info(sys) | 返回 | 阶跃动态指标 |
| step\_response(sys, T) | 阶跃响应 |
| impulse\_response(sys, T) | 脉冲响应 |
| initial\_response(sys, T) | 斜坡响应 |
| forced\_response(sys, T) | 抛物线响应 |
| nyquist\_plot(syslist) | 绘制 | Nyquist图 |
| bode\_plot(syslist) | Bode图 |
| root\_locus(sys) | 根轨迹图 |

|  |  |  |
| --- | --- | --- |
| 状态空间： | | |
| ss(A, B, C, D, dt=0) | | 线性系统 |
| ss2tf(A, B, C, D) | | 线性系统 -> 传递函数 |
| 实例属性 | nstate | 状态数 |
| 实例方法 | poles() | 极点 |
| zeros() | 零点 |
| damp() | 自然频率，阻尼比，极点 |

|  |  |
| --- | --- |
| ctrb(a, b) | 能控性判别矩阵 |
| obsv(a, c) | 能观性判别矩阵 |
| lyap(a, q) | 李雅普诺夫矩阵P |

规划cvxpy：

from cvxopt import matrix

|  |  |
| --- | --- |
| matrix(seq) | 返回序列对应的矩阵 |

from cvxopt.solvers import qp

|  |  |
| --- | --- |
| 规划solvers (列向量x)： | |
| qp(P, q, G, h, A, b) | 二次规划：min (0.5 \* x.T @ P @ x + q.T @ x  s.t. G @ x <= h , A @ x == b  P: 正半定矩阵 |

import cvxpy as cp

|  |  |  |
| --- | --- | --- |
| 基本对象： | | |
| Variable(shape, name, boolean, integer, pos, neg) | 实例化变量 | .value |
| Minimize(expr) | 实例化优化目标 | 最小 |
| Maximize(expr) | 最大 |
| Problem(target, constraints) | 实例化规划问题 | |
| PRO.solve(solver) | 更新.status、.value和变量 | |



|  |  |  |  |
| --- | --- | --- | --- |
| 运算： | | | |
| sum() | 求和 | multiply() | 逐项乘 |

矩阵numpy：

import numpy as np

基类：ndarray

|  |  |  |  |
| --- | --- | --- | --- |
| 常量： | | | |
| pi | п | nan | 浮点数空值 |

|  |  |
| --- | --- |
| 基本操作： | |
| array(seq, dtype=np.bool\_) | 返回序列对应的矩阵 |
| x.flatten() | 返回矩阵展开行向量 |
| x.astype(np.float32) | 转换x的数据类型 |
| x.T | 返回维度逆置的矩阵 |
| x @ y | 返回x左乘y的结果 |
| x.copy() | 返回x的拷贝 |
| round(x, num) | 精度控制 |
| ceil(x) / floor(x) | 向上 / 向下取整 |
| real(x) / imag(x) | 实部 / 虚部 |
| angle(x, deg) | 复数相角 |
| max(x) / min(x) | 最大 / 最小值及索引 |

|  |  |
| --- | --- |
| bool运算： | |
| x[bool] | 利用bool矩阵进行索引 |
| isnan(x) | 返回is nan的bool矩阵 |
| where(x, true\_out, false\_out) | 返回x==0的类bool矩阵 |
| where(x) | 返回非0元素的索引元组 |
| any(x) / x.any() | 判断是否有非0元素 |
| all(x) / x.all() | 判断是否全为非0元素 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 矩阵属性访问、变换： | | | | |
| x.ndim | 返回  矩阵  属性 | 行数 | | |
| x.shape | 维度 | | |
| x.size | 元素数 | | |
| x.dtype | 数据类型 | | |
| x.sum() | cells之和 | | 默认：偏移量  列 axis = 0  行 axis = 1  keepdims：  是否保持维度 |
| x.min() / x.max() | cells的最值 | |
| x.argmin() / x.argmax() | cells最值索引 | |
| x.item() | 转标量 | |
| x.mean() | cells平均值 | |
| percentile(x, percent) | percent分界数值 | |
| concatenate(xs) | 返回  变换  矩阵 | 在原维度 | 合并 |
| stack(xs) | 创建维度 |
| split(x, n) | n均等 | 分割 |
| array\_split(x, n) | n不均等 |
| cumsum(x) | cells累加 | |
| diff(x) | cells元素差 | |
| sort(x) | cells按行排序 | |
| clip(x, a\_min, a\_max) | cells∈[a\_min, a\_max] | |
| repeat(x) | cells重复 | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 矩阵公式： | | | | | | | |
| cov(x) | 协方差 | std(x, ddof) | 标准差 | var(x) | 方差 | sign(x) | 正负 |
| sqrt(x) |  | log(x, where) |  | log2(x) |  | log10(x) |  |
| abs(x) | |x| | sin(x) |  | cos(x) |  | tan(x) |  |
| exp(x) |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 矩阵创建： | | |
| x.reshape(shape) | 返回  指定  形状  矩阵 | cells继承x |
| full(shape, ele, dtype) | cells = ele |
| zeros(shape, dtype) | cells = 0 |
| empty(shape, dtype) | cells -> 0 |
| ones(shape, dtype) | cells = 1 |
| eye(rows, cols) | 单位矩阵 |
| arange(min, max, h) | 返回  行向量 | [min, max) 步长h |
| linspace(min, max, n) | 等差数列 |
| logspace(min, max, n) | 等比数列 |
| meshgrid(\*x) | 笛卡尔积 | |
| mgrid[\*slices] |
| append(x, value) | 添加元素 | |
| delete(x, idx, axis) | 移除元素 | |

|  |  |  |
| --- | --- | --- |
| 功能函数： | | |
| diag(x) | 转化为对角线矩阵 / 对角线行向量 | |
| sort(x, axis) | 返回排序结果 | |
| argsort(x, axis) | 返回排序结果的索引 | |
| prod(x, axis) | 返回累乘结果 | |
| interp(x, xp, fp, left, right) | 返回1D线性插值结果 | |
| polyfit(x, y, deg) | 拟合多项式并返回系数 | |
| poly1d(param, r, variable) | r=False | 系数向量 -> 多项式函数 |
| r=True | 解向量 -> 多项式函数 |
| unique(x) | 返回去重数据 | |
| histogram(data, bins, range) | 返回直方图 (区间计数, 区间边界) | |
| bincount(x, min\_length) | 返回数值统计结果 | |
| x.transpose(\*idx) | 按索引重排维度返回数组 | |
| pad(x, pad\_width, constant\_values) | 矩阵边界填充 | |

|  |  |
| --- | --- |
| 角度弧度： | |
| degrees / rad2deg | 弧度 → 角度 |
| radians / deg2rad | 角度 → 弧度 |

|  |  |
| --- | --- |
| IO： | |
| finfo(dtype) | 返回数据类型的极限信息 |
| loadtxt(fname, dtype) | 载入txt中的数组 |
| savetxt(fname, x) | 将数组写入txt |
| set\_printoptions(precision, threshold, suppress) | 设置输出格式 |

随机random：

|  |  |  |  |
| --- | --- | --- | --- |
|  | 定义 | 方差 | 概率密度 |
| 正态 / 高斯分布 |  |  |  |
| gumbel 分布 |  |  |  |

|  |  |  |
| --- | --- | --- |
| seed(idx) | 选定指定的随机数初态 | |
| random(size) / rand(\*size) | 返回均匀分布 |  |
| uniform(low, high, size) |  |
| randint(low, high, size) |
| randn(\*size) | 返回正态分布 |  |
| normal(mu, sigma, size) |  |
| gumbel(mu, beta, size) | 返回gumbel分布 |  |
| choice(a, size, replace, p) | 返回行向量指定次概率选择结果 | |
| shuffle(x) | 打乱第一维度元素 | |
| permutation(x | index) | 打乱所有元素 / 随机序列 | |

傅里叶fft：

|  |  |
| --- | --- |
| fft(x) | 在时域上进行傅里叶变换 |

线代linalg：

正交矩阵：

|  |  |  |
| --- | --- | --- |
| inv(x) | 返回逆矩阵 | 正交矩阵： |
| eig(x) | 返回特征值v, 特征向量U | 对称阵：为正交矩阵 |
| svd(x) | 奇异值分解 |  |
| det(x) | 返回行列式计算结果 |  |
| norm(x, ord, axis) | 返回L2范数 |  |
| solve(A, b) | 返回Ax = b的解 |  |
| multi\_dot(xS) | 返回多项乘法结果 |  |
| matrix\_power(x, power) | 返回矩阵幂 |  |

分配ot：

|  |  |
| --- | --- |
| 代价矩阵M，分布a → 分布b | |
| emd / emd2(a, b, M, return\_matrix) | 推土机距离 |
| sinkhorn / sinkhorn2(a, b, M, reg) | 熵正则化的推土机距离 |

表单pandas：

import pandas as pd

|  |  |
| --- | --- |
| concat(dfs, axis) | 表单拼接 |
| crosstab(idxSER, colSER, values) | 实例化交叉表 |

|  |  |  |
| --- | --- | --- |
| 向量： | | |
| Series(seq, index) | | 创建1D向量 |
| 实例方法 | apply(fun) | 应用映射函数 |

|  |  |  |  |
| --- | --- | --- | --- |
| 表单： | | | |
| DataFrame(seq, index, columns, dtype) | | 创建2D表单 | |
| 索引 | ['col'] | 列标签 | |
| loc['row'] / loc['row', 'col'] | 行标签 | |
| iloc[row\_idx] | 行序号 | |
| 轴属性 | index | 行索引 | 实例 |
| index.values | 列表 |
| index.name | 标签 |
| columns | 列索引 | 实例 |
| columns.values | 列表 |
| columns.name | 标签 |
| empty | 空数据表 | |
| dtypes | 数据类型 | |
| 轴操作 | iterrows() | 返回行迭代器 | |
| append(seq, ignore\_index=True) | 返回添加列的表单 | |
| insert(loc, col, value) | 在指定位置插入列 | |
| reset\_index(drop=True) | 重排行索引 | |
| sort\_index(axis, ascending) | 返回排序表单 | 按索引 |
| sort\_values(axis, by) | 按标签下值 |
| 过滤 | fillna(val, inplace) | 填充表单空值 | |
| isin(seq) | 返回表单元素在序列中的布尔表单 | |
| truncate(before, after) | 返回行过滤结果 | |
| drop\_duplicates(subset, keep, inplace) | 返回移除重复行的表单 | |
| 分析 | median(axis) | 返回表单在axis上的中值 | |
| info() | 输出有效值统计、数据类型 | |
| describe() | 返回统计数据 (cnt, mean, std, min ~ max) | |
| groupby(SER, axis) | 返回分组结果 | |
| sample(frac, random\_state) | 返回随机采样结果 | |
| plot. ...... | 绘图 (数据用列索引代替) | |

|  |  |  |
| --- | --- | --- |
| 时间序列： | | |
| Timestamp(str) | 返回时间戳实例 | |
| DatetimeIndex(seq) | 返回 | 日期 |
| date\_range(start, end, periods, freq) | 生成 |
| PeriodIndex(seq) | 返回 | 时期 |
| period\_range(start, end, periods, freq) | 生成 |
| Timedelta(days, minutes, seconds) | 返回时间间隔实例 | |

|  |  |
| --- | --- |
| excel文件： | |
| read\_excel(file, sheet\_name, header, index\_col, true\_values) | 读取excel文件 |
| ExcelWriter(file\_name) | 创建excel文件实例 |
| <DataFrame>.to\_excel(<ExcelWriter>, index, sheet\_name) | 将表单导出到excel |
| <ExcelWriter>.save() | 保存excel |

|  |  |
| --- | --- |
| csv文件： | |
| read\_csv(file, index\_col, true\_values, encoding) | 读取csv文件 |
| <DataFrame>.to\_csv(file, index=False) | 将表单导出到csv |

科学scipy：

优化optimize：

|  |  |  |
| --- | --- | --- |
| linprog(c, A\_ub, b\_ub, A\_eq, b\_eq) | 线性规划：min x @ c.T  s.t. x @ A\_ub.T - b\_ub <= 0  x @ A\_eq.T - b\_eq == 0  x >= 0 | .fun：最小值 |
| .x：最优解 |
| .success：状态 |
| .message：信息 |
| minimize(fun, x0, method, constraints) | 非线性规划：{'type': 'eq'|'ineq'(>=), 'fun': FUN} | |
| leastsq(error\_fun, init) | 最小化指定初态下的平方和 | [ x, fun ] |
| curve\_fit(fun, x, y)[0] | 拟合函数，返回函数参数 | |
| linear\_sum\_assignment(cost, maximize) | 匈牙利算法返回最优分配策略 | |

插值interpolate：

|  |  |  |  |
| --- | --- | --- | --- |
| interp1d(x, y, kind) | kind | 'linear'：线性 | 'cubic'：立方 |
| 'zero'、'nearest'：阶梯 | |
| 实例化1D插值器 (callable) | | |
| interp2d(x, y, z, kind) | kind | 'linear'：线性 | 'cubic'：立方 |
| 实例化2D插值器 (callable) | | |



机器sklearn：

from efficient\_apriori import apriori

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 代码 | apriori(data, min\_support, min\_confidence) | | | 返回 (频繁项集, 规则) |
| 原理 | 项集 | 定义 | 单个项 / 一系列项目的集合 | |
| 性质 | 频繁项集的子集也是频繁项集 | |
| 支持度Sup | 项集在数据中出现的比例 | |
| 规则 | 定义A->B | 根据频繁项集挖掘出的结果 | |
| 置信度Conf | 出现先导也出现后继的比例 |  |
| 提升度Lift | 衡量先导与后继的独立性 |  |
| 确信度CF | 不发生先导而发生后继的概率  与这条规则错误的概率比值 |  |

聚类cluster：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 基于距离： | | | | |
| 代码 | KMeans(n\_cluster, max\_iter, n\_init, init) | | 实例化K均值聚类算法模型 | |
| n\_cluster：生成聚类数 | | max\_iter：最大迭代数，默认300 | |
| n\_init：子代数，默认10 | | init | 'k-means++'：选定初始质心 |
| algorithm：使用算法 | | 'random'：随机选定 |
| 实例属性 | labels\_ | 聚类结果 | |
| cluster\_centers\_ | 聚类中心信息 | |
| inertia\_ | 簇惯性SSE | |
| 实例方法 | fit(x) | 训练模型 | |
| 优点 | 收敛速度较快 | | | |
| 缺点 | 结果不稳定、无法解决样本不均衡的问题、容易收敛到局部最优解、受噪声影响大 | | | |
| 手肘法 | 876995db6780d5d2185fc43ff98325a | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 基于密度： | | | |
| 代码 | DBSCAN(eps, min\_samples, metric) | | 返回 (样本索引, 群集索引) |
| 形参 | eps | 邻域半径 |
| min\_samples | 最少样本 |
| 实例属性 | labels\_ | 聚类结果 |
| 实例方法 | fit(x) | 训练模型 |
| 优点 | 不需要划分个数 | | |
| 可以处理噪声点 | | |
| 可以处理任意形状的空间聚类问题 | | |
| 缺点 | 需要制定最小样本量和半径两个参数 | | |
| 数据量大时开销也大 | | |
| 样本集的密度不均匀、聚类间距差相差很大时，聚类质量较差 | | |

数据dataset：

|  |  |
| --- | --- |
| load\_iris(as\_frame) | 返回鸢尾植物数据集 |
| make\_moons(n\_samples, noise, shuffle, random\_state) | 生成两曲线散点数据集 |

集成ensemble：

|  |  |  |  |
| --- | --- | --- | --- |
| 代码 | RandomForestClassifier(n\_estimators, random\_state) | | 实例化随机森林算法模型 |
| 实例属性 | features\_importances\_ | 特征权值 |
| 实例方法 | fit(x, y) | 训练模型 |
| score(x, y) | 返回模型准确度 |
| predict(x) | 返回预测结果 |
| predict\_proba(x) | 返回预测概率 |
| 优点 | 可处理很高维度的数据，无需降维、做特征选择 | | |
| 可判断特征的重要程度 | | |
| 可判断出不同特征之间的相互影响 | | |
| 不容易过拟合 | | |
| 对不平衡的数据集来说，可平衡误差 | | |
| 如有很大一部分的特征遗失，仍可以维持准确度 | | |
| 缺点 | 在某些噪音较大的分类或回归问题上会过拟合 | | |
| 取值划分较多的属性影响大，产出的属性权值不可信 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 代码 | BaggingClassifier(base\_estimator, n\_estimators) | | 实例化Bagging算法模型 |
| 实例方法 | fit(x, y) | 训练模型 |
| predict(x) | 返回预测结果 |
| predict\_proba(x) | 返回预测概率 |
| 原理 | 训练 | 有放回地采样得到k个训练集，得到k个弱学习器 | |
| 预测 | 多个弱学习器共同投票 | |

|  |  |  |  |
| --- | --- | --- | --- |
| 代码 | AdaBoostClassifier(base\_estimator, n\_estimators) | | 实例化AdaBoost算法模型 |
| 实例方法 | fit(x, y) | 训练模型 |
| predict(x) | 返回预测结果 |
| predict\_proba(x) | 返回预测概率 |
| 原理 | a71124afecaa3762e2ecf897ad6dd91 | | |

线性linear\_model：

|  |  |  |
| --- | --- | --- |
| LinearRegression(fit\_intercept, positive) | | 实例化线性回归模型 |
| 实例属性 | coef\_ | 返回特征系数 |
| intercept\_ | 返回偏置 |
| 实例方法 | fit(x, y) | 返回回归模型 |
| score(x, y) | 返回模型准确度 |
| predict(x) | 返回预测结果 |

|  |  |  |
| --- | --- | --- |
| LogisticRegression(penalty, fit\_intercept) | | 实例化逻辑回归模型 |
| 实例属性 | coef\_ | 返回特征系数 |
| intercept\_ | 返回偏置 |
| 实例方法 | fit(x, y) | 返回回归模型 |
| score(x, y) | 返回模型准确度 |
| predict(x) | 返回预测结果 |
| predict\_proba(x) | 返回预测概率 |

采样model\_selection：

|  |  |
| --- | --- |
| train\_test\_split(\*arrays, test\_size) | 数据拆分：train\_x, eval\_x, train\_y, eval\_y... |
| cross\_val\_predict(estimator, x, y, cv) | K折交叉验证 |

多分类multiclass：

|  |  |  |
| --- | --- | --- |
| OneVsOneClassifier(estimator) | | 实例化投票模型 |
| OneVsRestClassifier(estimator) | | 实例化阳性检测模型 |
| 实例方法 | fit(x, y) | 返回回归模型 |
| score(x, y) | 返回模型准确度 |
| predict(x) | 返回预测结果 |

贝叶斯naive\_bayes：

|  |  |  |  |
| --- | --- | --- | --- |
| 代码 | GaussianNB() | | 实例化高斯朴素贝叶斯模型 |
| 实例属性 | class\_prior\_ | 类别的先验概率 |
| theta\_ | 各类各特征均值 |
| sigma\_ | 各类各特征方差 |
| 实例方法 | fit(x, y) | 训练模型 |
| score(x, y) | 返回模型准确度 |
| predict(x) | 返回预测结果 |
| predict\_proba(x) | 返回预测概率 |
| 原理 |  | | |
| 密度函数 |  | | |
| 连续值 | 通过样本集计算每个分类下的均值和方差 | | |
| 计算出新数据与均值的距离 (如：密度函数) | | |
| 优点 | 适合大规模数据，对于噪声点和无关属性比较健壮 | | |
| 缺点 | 需要考虑特征之间的相互独立性 | | |
| 拓展 | 半朴素贝叶斯、AODE | | |

近邻neighbors：

|  |  |  |  |
| --- | --- | --- | --- |
| 代码 | KNeighborsClassifier() | | 实例化KNN近邻算法模型 |
| 实例方法 | fit(x, y) | 训练模型 |
| score(x, y) | 返回模型准确度 |
| predict(x) | 返回预测结果 |
| predict\_proba(x) | 返回预测概率 |
| kneighbors(x, n\_neighbors) | 返回距离最近的n个样本信息 |
| 原理 | 全部训练样本作为代表向量，取K (奇数) 个最近邻进行投票 | | |
| K过小容易过拟合，K过大容易欠拟合，可对代表向量进行剪枝 | | |
| 优点 | 对于边界不规则的数据效果较好 | | |
| 缺点 | 只适合小数据集，数据不平衡效果不好，必须做数据标准化 | | |

感知机neural\_network：

|  |  |  |
| --- | --- | --- |
| MLPRegressor(hidden\_layer\_sizes, activation) | | 实例化MLP回归器 |
| 实例方法 | fit(x, y) | 训练模型 |
| score(x, y) | 返回模型分数 |
| predict(x) | 返回预测结果 |

管道pipeline：

|  |  |  |
| --- | --- | --- |
| make\_pipeline(\*models) | | 返回集成模型 |
| 实例方法 | fit(x, y) | 训练模型 |
| predict(x) | 返回预测结果 |

预处理preprocessing：

|  |  |  |
| --- | --- | --- |
| PolynomialFeatures(degree, include\_bias) | | 实例化二项式特征提取器 |
| 实例属性 | powers\_ | shape: [n\_out\_features, n\_in\_features] |
| 实例方法 | fit\_transform(x) | 返回二项式特征： |

|  |  |
| --- | --- |
| StandardScaler().fit\_transform(x) | 标准化 |
| minmax\_scale(x, feature\_range) | 归一化 |
| normalize(x, norm) | 规范化 |

向量机svm：

|  |  |  |  |
| --- | --- | --- | --- |
| 代码 | SVC(kernel='linear') | | 实例化支持向量机模型 |
| 实例属性 | coef\_ | 权值向量 |
| intercept\_ | 超平面偏置 |
| 实例方法 | fit(x, y) | 训练模型 |
| score(x, y) | 返回模型精度 |
| predict(x) | 返回预测结果 |
| predict\_proba(x) | 返回预测概率 |
| 原理  (二分类) | 分界超平面 |  | |
| 极端超平面 |  | |
| 超平面间隔  (maximize) |  | |
| 约束条件 |  | |
| 优化目标 |  | |
| 核函数 |  | |
| 优点 | 找到最优超平面，鲁棒性好 | | |
| 缺点 | 训练所需资源大 (类似深度学习) | | |

决策树tree：

|  |  |
| --- | --- |
| export\_graphviz(model, out\_file, feature\_names,  class\_names, filled=True, rounded) | 返回决策树的DOT格式数据 |

|  |  |  |  |
| --- | --- | --- | --- |
| 代码 | DecisionTreeClassifier(criterion, max\_depth, min\_samples\_leaf) | | 实例化分类器 |
| DecisionTreeRegressor(criterion, max\_depth, min\_samples\_leaf) | | 实例化回归器 |
| 实例属性 | feature\_importances\_ | 返回特征重要性 |
| 实例方法 | fit(x, y) | 训练模型 |
| score(x, y) | 返回模型准确度 |
| predict(x) | 返回预测结果 |
| predict\_proba(x) | 返回预测概率 |
| 原理 |  | | |
| 优点 | 可处理离散值、连续值、缺失值 | | |
| 缺点 | 容易过拟合 | | |
| 需要处理样本不均衡的问题 | | |
| 样本的变化会引发树结构巨变 | | |
| 剪枝 | 预剪枝：构建之初设定阈值，当分裂节点的熵阈值小于设定值停止 | | |
| 后剪枝：构建完成后，根据设定的条件判断是否合并中间节点 | | |
| 拓展 | 随机森林、GBDT、XGBoost | | |

符号sympy：

|  |  |  |  |
| --- | --- | --- | --- |
| oo | ∞ | pi | π |
| E | e | I | i |
| log(x, a) | loga(x)，a默认为e | | |
| sin(x) | sin( ) | asin(x) | arcsin( ) |
| cos(x) | cos( ) | acos(x) | arccos( ) |
| tan(x) | tan( ) | atan(x) | arctan( ) |
| sqrt(x) | x1/2 | root(x, n) | x1/n |
| factorial(x) | x! |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 希腊字母表： | | | | | | | |
| α | alpha | β | beta | γ/Γ | gamma | δ/Δ | delta |
| κ | kappa | λ | lambda | φ/Φ | phi | σ/Σ | sigma |
| ε | epsilon | ζ | zeta | η | eta | ω/Ω | omega |
| θ | theta | τ | tau | ρ | rho | ο | omicron |
| μ | mu | ψ | psi | ξ | xi | ν | nu |

|  |  |
| --- | --- |
| 输出： | |
| init\_printing() | 初始化print环境 |
| pprint(value) | 输出 |

|  |  |
| --- | --- |
| 声明： | |
| symbols('x, y', positive) | 返回自变量元组 |
| Function('f') | 返回未知函数 |
| Matrix(array) | 返回矩阵 |

|  |  |  |
| --- | --- | --- |
| 表达式简化： | | |
| expand(f) | 展开式 | |
| factor(f) | 折叠式 | |
| apart(f) | 分数 | 分离 |
| together(f) | 合并 |
| symplify(f) | 化简 | 普通 |
| trigsimp(f) | 三角 |
| powsimp(f) | 指数 |

|  |  |
| --- | --- |
| 求解： | |
| solve(f, \*symbols) | 返回表达式 = 0的解向量 |
| dsolve(f) | 返回常微分方程的通解 |
| summation(f, \*(x, L, R)) | 返回表达式在 [L, R] 的和 |
| limit(f, symbol, symbol\_0) | 返回表达式在 symbol -> symbol\_0 的极限 |
| diff(f, \*symbol, \*time) | 返回表达式对symbol的导数 |
| integrade(f, (symbol, L, R)) | 返回表达式关于symbol的积分 (可不指定上下限) |

|  |  |  |
| --- | --- | --- |
| 表达式/等式： | | |
| Eq(lhs, rhs) | | 返回实例化的等式 |
| 实例属性 | lhs | 等式左侧的表达式 |
| rhs | 等式右侧的表达式 |
| free\_symbols | 式子的自由变量集合 |
| 实例方法 | subs(dict) | 返回代入计算结果 |

|  |  |  |
| --- | --- | --- |
| 矩阵： | | |
| Matrix(seq) | | 返回序列对应的矩阵 |
| eye(\*shape) | | 返回单位阵 |
| zeros(\*shape) | | 返回0矩阵 |
| ones(\*shape) | | 返回1矩阵 |
| 实例属性 | shape | 矩阵形状 |
| 实例方法 | exp() | 返回e^x |
| inv() | 返回逆矩阵 |
| det() | 返回行列式 |
| eigenvals() | 返回特征值 |
| rref() | 返回行最简形矩阵 |
| rank() | 返回矩阵秩 |
| 类方法 | hstack(\*args) | 横向拼接 |
| vstack(\*args) | 纵向拼接 |

|  |  |  |
| --- | --- | --- |
| 变换： | | |
| laplace\_transform(f, t, s) | 拉普拉斯 | 变换 |
| inverse\_laplace\_transform(F, s, t) | 逆变换 |

可视visdom：

python -m visdom.server 获取监听服务器地址

from visdom import Visdom

|  |  |  |
| --- | --- | --- |
| Visdom() | | 实例化画布 |
| 实例方法 | line(y, x, win, update='append', opts) | 绘制折线 |
| images(img, win) | 绘制图像 |
| text(str, win, opts) | 展示文本 |
| win：窗口标题 | opts=dict(title, legend)：选项 |  |
| update：更新方式 | legend：图例列表 |  |

森林xgboost：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 增强森林 | 代码 | XGBClassifier(n\_estimators, learning\_rate, max\_depth) | | 实例化增强森林算法模型 |
| 实例方法 | fit(data, target) | 训练模型 |
| predict(data) | 返回预测结果 |
| 原理 | loss：复杂度 (叶结点数, 结点值) + 样本结点值和 | | |
| 优点 | 正则化、缺失值处理、剪枝、内置交叉验证 | | |
| 缺点 | 算法参数过多、只适合处理结构化数据、不适合处理超高维特征数据 | | |