

火龙果软件学院

机器学习项目实践

Machine Learning Kaggle



Evolve by case

培训目标

- 数据竞赛的基本形式和数据载入方法
- 特征探索和特征工程
- 模型训练、调优和评估的一般方法
- 常用线性回归模型
- 梯度增强回归树和XGBoost



项目简介和数据载入

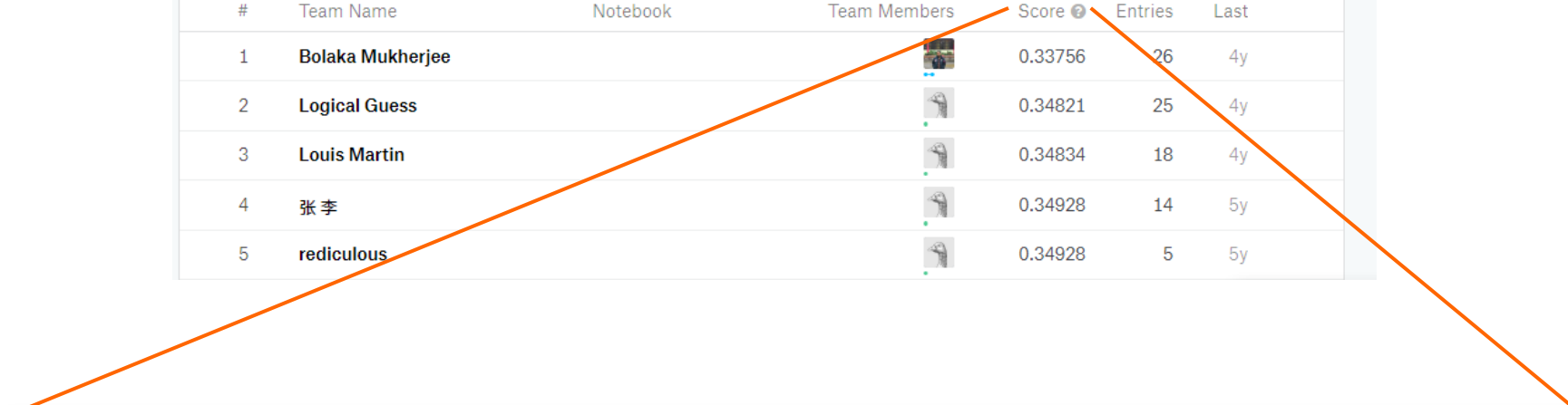


项目背景

- <https://www.kaggle.com/c/bike-sharing-demand/overview>
- In this competition, participants are asked to combine historical usage patterns with weather data in order to forecast bike rental demand in the Capital Bikeshare program in Washington, D.C.



Kaggle数据竞赛的形式



Overview Data Notebooks Discussion Leaderboard Rules Team My Submissions **Late Submission**

Public Leaderboard Private Leaderboard

This leaderboard is calculated with all of the test data. [Raw Data](#) [Refresh](#)

| # | Team Name | Notebook | Team Members | Score ? | Entries | Last |
|---|------------------|----------|--------------|---------|---------|------|
| 1 | Bolaka Mukherjee | | | 0.33756 | 26 | 4y |
| 2 | Logical Guess | | | 0.34821 | 25 | 4y |
| 3 | Louis Martin | | | 0.34834 | 18 | 4y |
| 4 | 张李 | | | 0.34928 | 14 | 5y |
| 5 | rediculous | | | 0.34928 | 5 | 5y |

| # | Team Name | Notebook | Team Members | Score ? | Entries | Last |
|---|------------------|----------|--------------|--|---------|------|
| 1 | Bolaka Mukherjee | | | <div>Score Root Mean Squared Logarithmic Error</div> | 26 | 4y |
| 2 | Logical Guess | | | 0.34821 | 25 | 4y |



数据格式

[Overview](#) [Data](#) [Notebooks](#) [Discussion](#) [Leaderboard](#) [Rules](#) [Team](#) [My Submissions](#) [Late Submission](#)

Data Description

[See, fork, and run a random forest benchmark model through Kaggle Scripts](#)

You are provided hourly rental data spanning two years. For this competition, the training set is comprised of the first 19 days of each month, while the test set is the 20th to the end of the month. You must predict the total count of bikes rented during each hour covered by the test set, using only information available prior to the rental period.

Data Fields

datetime - hourly date + timestamp
season - 1 = spring, 2 = summer, 3 = fall, 4 = winter
holiday - whether the day is considered a holiday
workingday - whether the day is neither a weekend nor holiday
weather - 1: Clear, Few clouds, Partly cloudy, Partly cloudy
2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
temp - temperature in Celsius
atemp - "feels like" temperature in Celsius
humidity - relative humidity
windspeed - wind speed
casual - number of non-registered user rentals initiated
registered - number of registered user rentals initiated
count - number of total rentals



数据下载

Data (189 KB)

[API](#) `kaggle competitions download -c bike-sharing-dem...` [?](#) [Download All](#)

| Data Sources | About this file | Columns |
|--|--------------------|--------------------------------|
| <div><div>sampleSubmission.c...</div><div>6494 x 2</div></div> | No description yet | <div><div>datetime</div></div> |
| <div><div>test.csv</div><div>6494 x 9</div></div> | | <div><div># count</div></div> |
| <div><div>train.csv</div><div>10.9k x 12</div></div> | | |

sampleSubmission.csv (139.51 KB)

2 of 2 columns ▾ Views

| datetime | # count |
|----------|---------|
| | |



数据载入与基本整理

- Pandas读入CSV数据
- 检查各列数据类型，对比网页说明修正错误的类型
- 检查异常数据列（NaN），确认并非文件编码、格式等错误导致
- 检查重复数据行，确认非文件分割、合并等错误导致

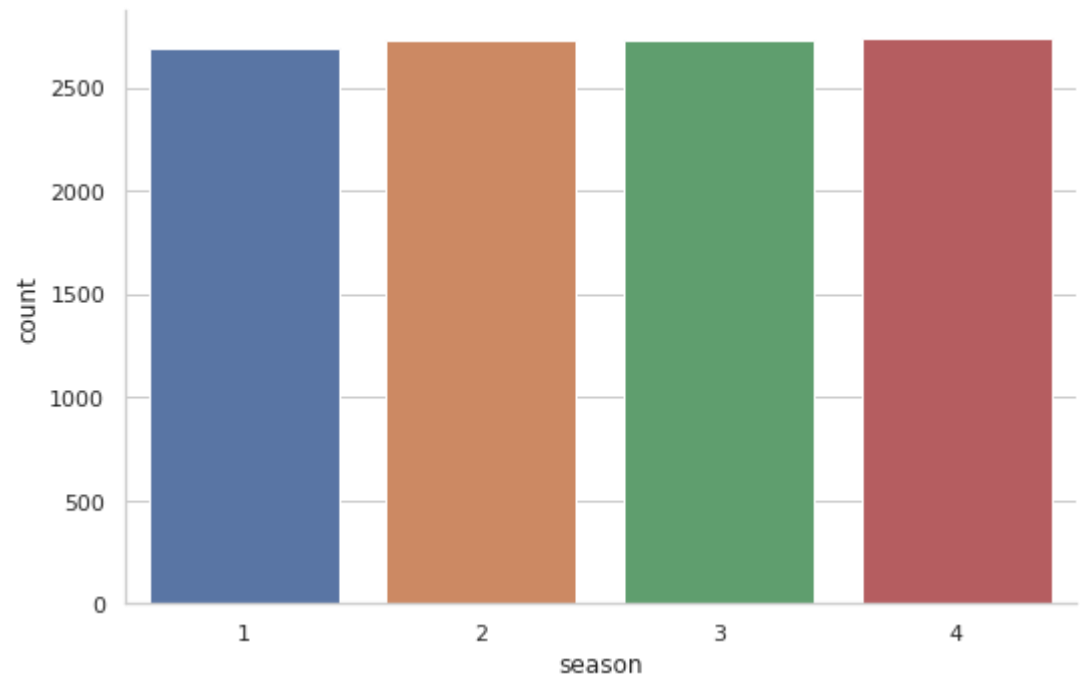
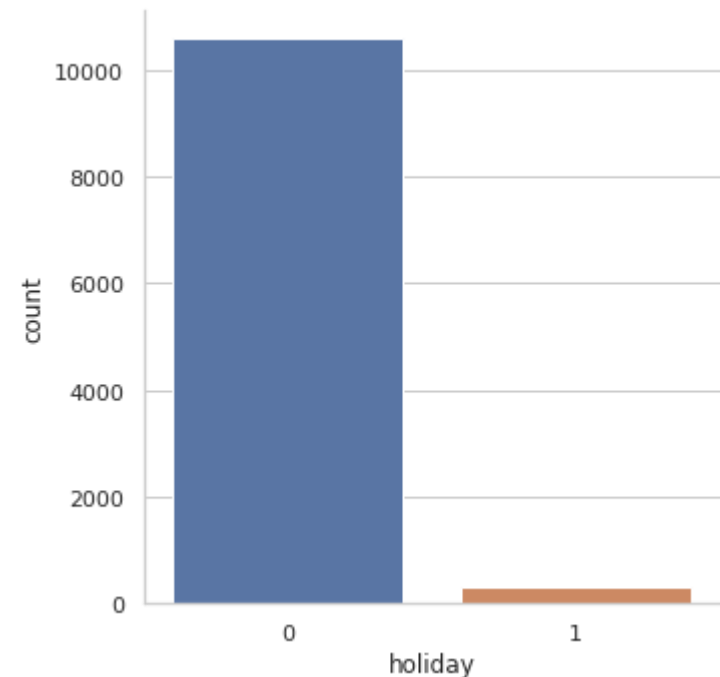


数据探索



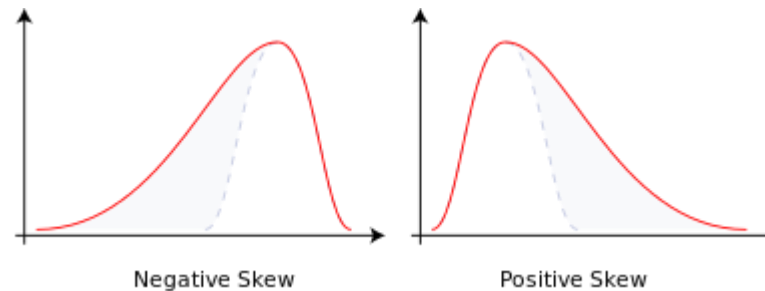
数据探索

- 观察各列数据取值和分布情况
- 不平衡的数据分布影响模型选择和训练方法



偏态的意义

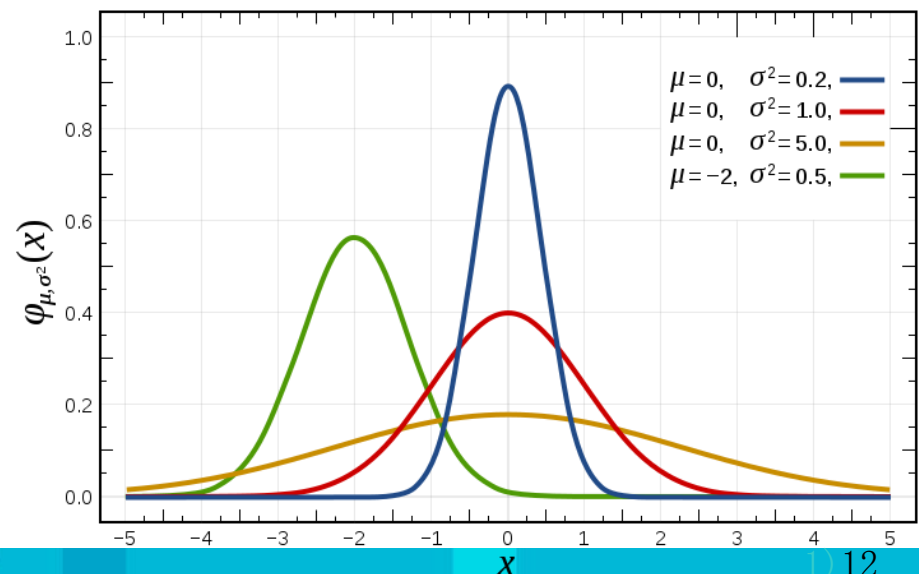
- 偏度为负（负偏态）就意味着在概率密度函数左侧的尾部比右侧的长
- 偏度为正（正偏态）就意味着在概率密度函数右侧的尾部比左侧的长
- 可以对偏态进行一些调整使其服从正态分布



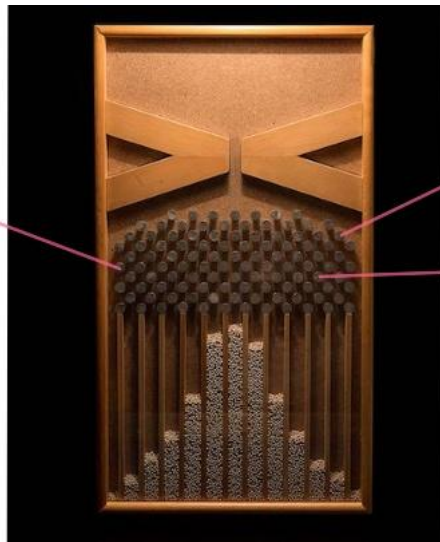
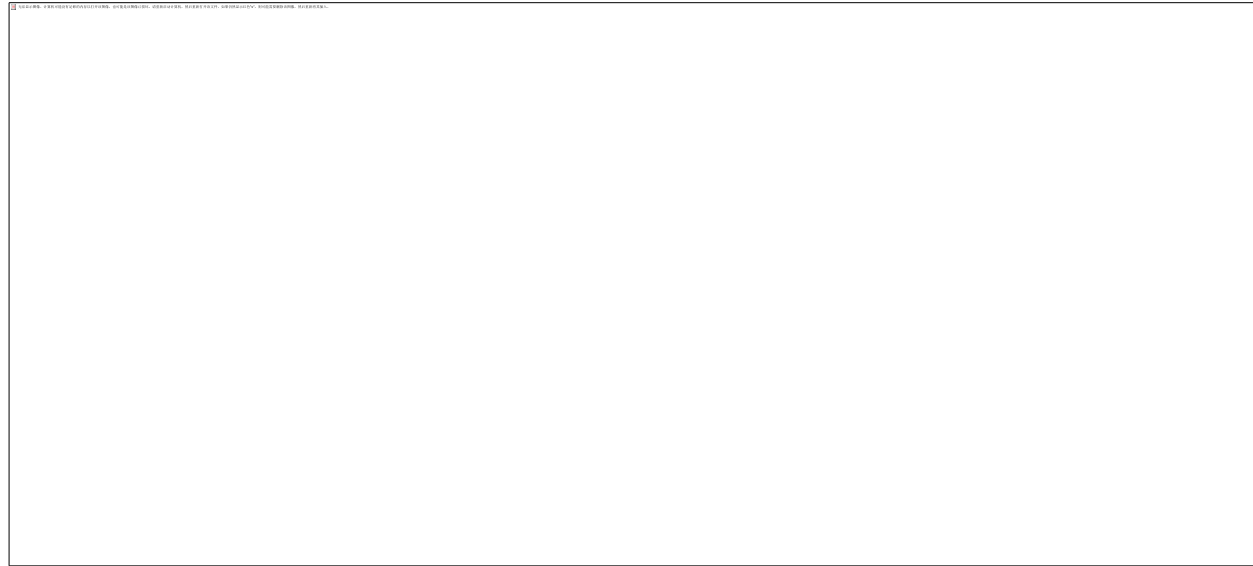
为什么喜欢正态分布

- 数据整体服从正态分布，那样本均值和方差则相互独立。
- 正态分布具有很多好的性质，很多模型假设数据服从正态分布（如线性回归假设误差服从正态分布）。
- **ML中很多model都假设数据或参数服从正态分布**

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



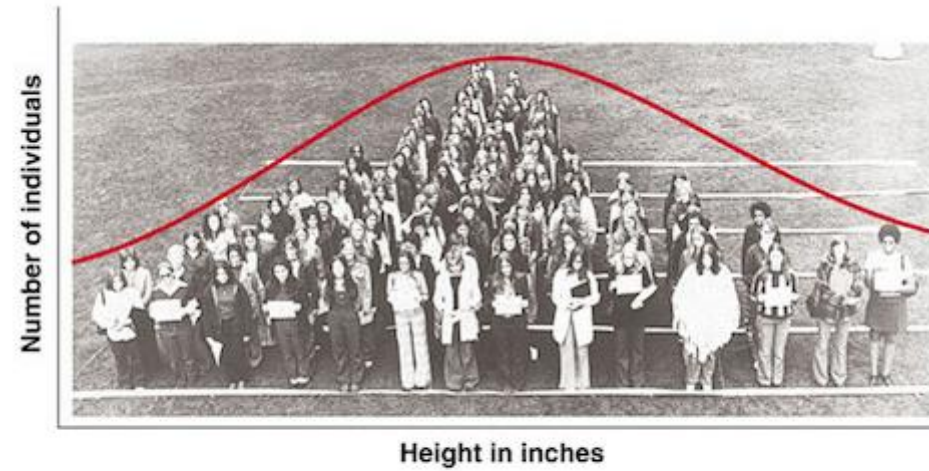
正态分布



饮食习惯

父母基因

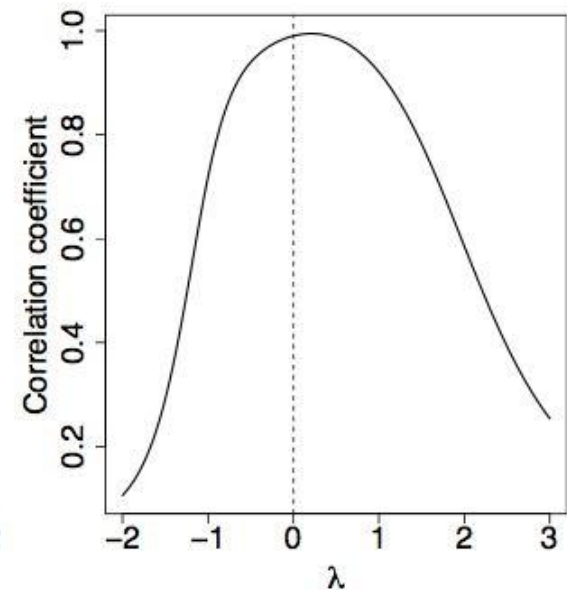
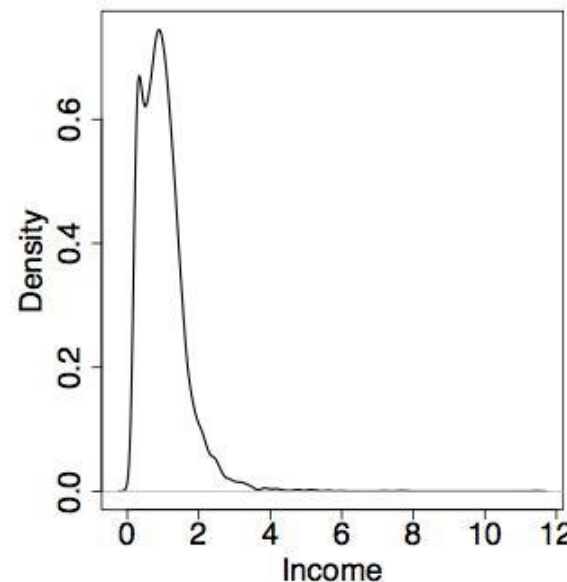
运动习惯



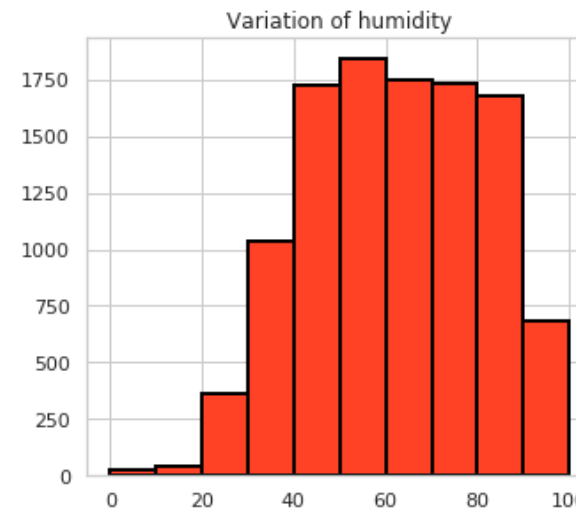
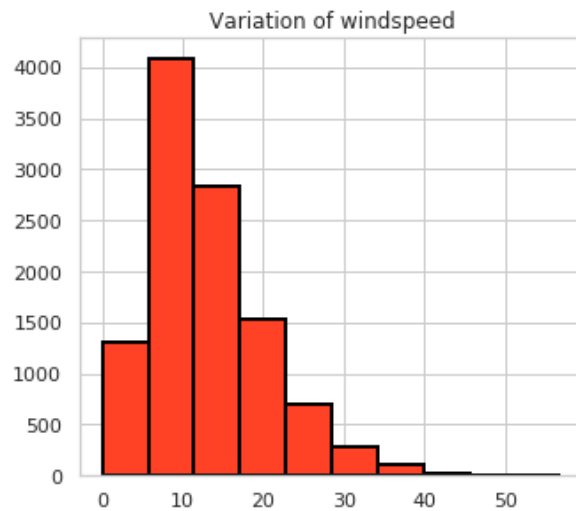
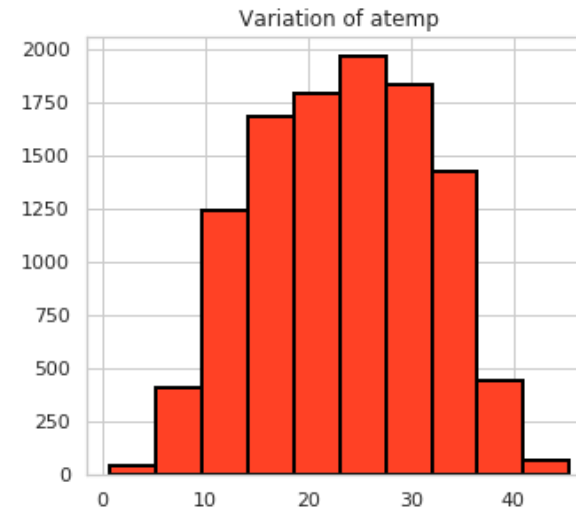
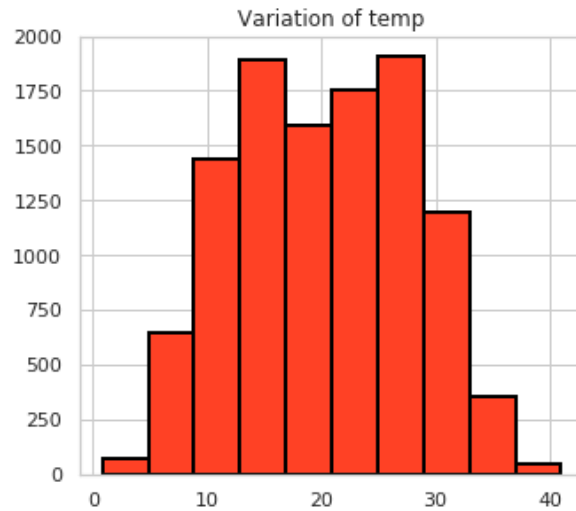
如果不是正态分布怎么办？

- 右偏（负偏态）的话可以对所有数据取对数、取平方根等，它的原理是因为这样的变换的导数是逐渐减小的，也就是说它的增速逐渐减缓，所以就可以把大的数据向左移，使数据接近正态分布。
- 左偏化为右偏后同上操作
- Box-Cox操作：<http://onlinestatbook.com/2/transformations/box-cox.html>

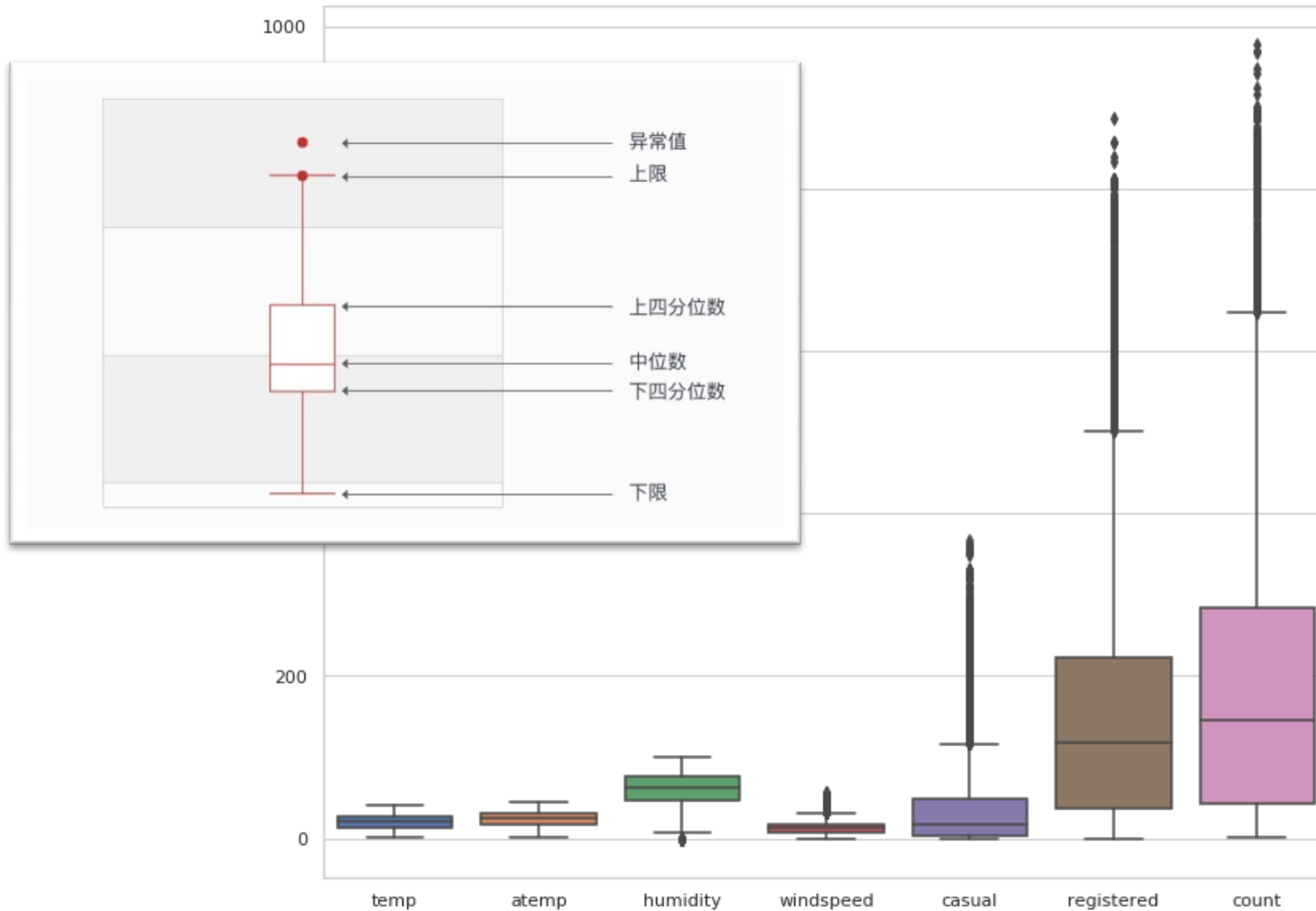
$$x'_\lambda = \frac{x^\lambda - 1}{\lambda}.$$



数据分布探索



异常值的判断



首次训练

- 训练集与测试集的分划
- 训练用列的选取
- 模型的导入与训练
- 在测试集进行预测
- 对预测结果进行评估



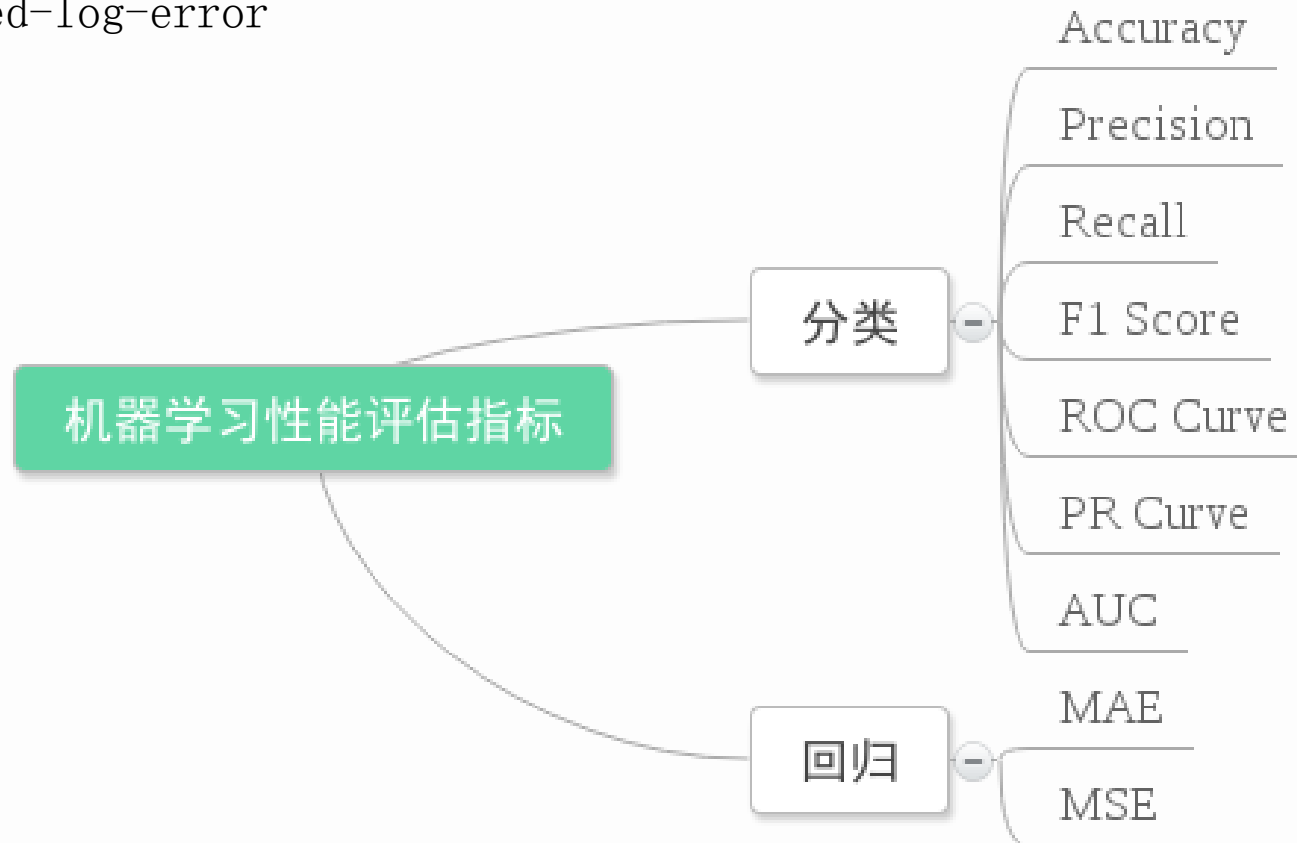
模型准确度评估



模型评估方法

完整文档:

https://scikit-learn.org/stable/modules/model_evaluation.html#mean-squared-log-error



https://blog.csdn.net/guohao_zhang

模型评估一般思路

- 样本中取出若干测试集
- 对每个测试集，使用待评估模型进行预测
- 将预测结果与实际采样结果做某种统计计算（MAE、MSE等）得到单个测试集上的评分
- 取所有评分的平均值



常用模型评估方法

- 平均绝对误差 (mean absolute error , MAE)

$$\text{MAE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} |y_i - \hat{y}_i|$$

- 均方误差 (mean squared error , MSE)

$$\text{MSE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2$$

- 均方对数误差 (mean squared logarithmic error , MSLE)

$$\text{MSLE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (\log_e(1 + y_i) - \log_e(1 + \hat{y}_i))^2$$

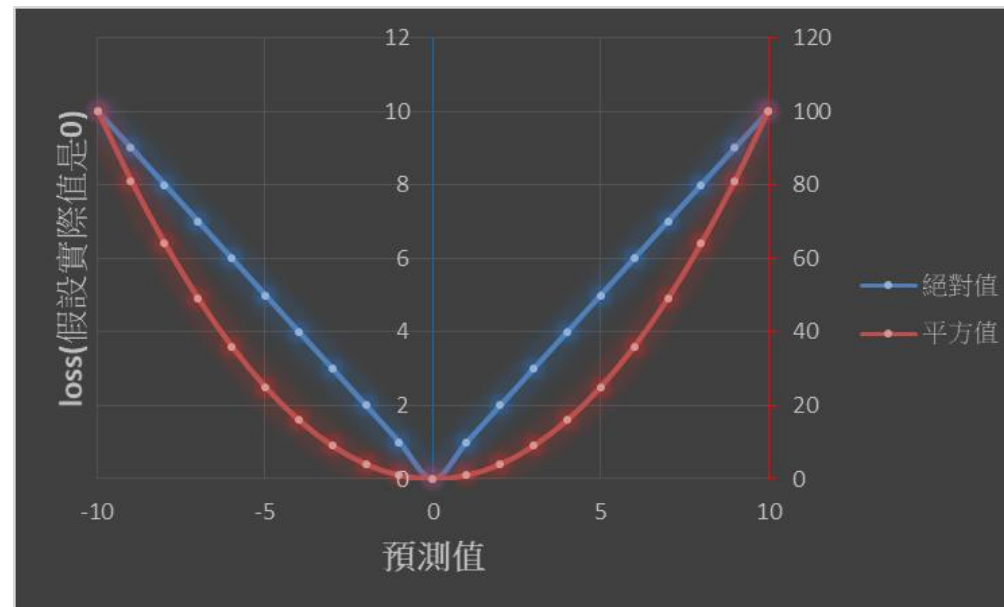
- MSE和MSLE引入了平方项导致量纲变化，一般取平方根，即RMSE和RMSLE



主要区别

| ID | residual | residual | residual ² |
|-------------------------|----------|----------|-----------------------|
| 1 | -10 | 10 | 100 |
| 2 | -5 | 5 | 25 |
| 3 | 0 | 0 | 0 |
| 4 | 5 | 5 | 25 |
| 5 | 10 | 10 | 100 |
| MAE=6, RMSE=7.07 | | | |

| ID | residual | residual | residual ² |
|---------------------------|----------|----------|-----------------------|
| 1 | -10 | 10 | 100 |
| 2 | -5 | 5 | 25 |
| 3 | 0 | 0 | 0 |
| 4 | 5 | 5 | 25 |
| 5 | 100 | 100 | 10000 |
| MAE=24, RMSE=45.06 | | | |



其他变化

- 均值→最大值，使分数对异常值更加敏感MaxError

$$\text{Max Error}(y, \hat{y}) = \max(|y_i - \hat{y}_i|)$$

- 均值→中位数，使分数对异常值更加稳定MedAE

$$\text{MedAE}(y, \hat{y}) = \text{median}(|y_1 - \hat{y}_1|, \dots, |y_n - \hat{y}_n|)$$

- 误差除以某个因子得到相对变化量

$$\text{explained_variance}(y, \hat{y}) = 1 - \frac{\text{Var}\{y - \hat{y}\}}{\text{Var}\{y\}}$$

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$



特征工程



什么是特征工程

- 请你设计一个身材分类器：
- 输入特征数据
 - ▶ X1：身高
 - ▶ X2：体重
- 输出标签
 - ▶ Y:身材等级（肥胖、超重、正常、过轻）



什么是特征工程——BMI指数就是典型的特征工程

- 目前判断肥胖的简单方法是使用BMI（指数）

$$\text{BMI} = \frac{\text{weight}_{(\text{kg})}}{\text{height}_{(\text{m})}^2}$$

身体质量指数(BMI)量表

| | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|
| 61.4 | 63.6 | 65.9 | 68.2 | 70.5 | 72.7 | 75.0 | 77.3 | 79.5 | 81.8 |
| 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 |
| 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 |
| 24 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| 23 | 24 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 22 | 23 | 24 | 25 | 25 | 26 | 27 | 28 | 29 | 30 |
| 21 | 22 | 23 | 24 | 25 | 25 | 26 | 27 | 28 | 29 |
| 21 | 22 | 22 | 23 | 24 | 25 | 25 | 26 | 27 | 28 |
| 20 | 21 | 22 | 22 | 23 | 24 | 25 | 25 | 26 | 27 |
| 20 | 20 | 21 | 22 | 22 | 23 | 24 | 25 | 25 | 26 |
| 19 | 20 | 20 | 21 | 22 | 23 | 23 | 24 | 25 | 26 |
| 18 | 19 | 20 | 21 | 21 | 22 | 23 | 23 | 24 | 25 |
| 18 | 19 | 19 | 20 | 21 | 21 | 22 | 23 | 23 | 24 |
| 17 | 18 | 19 | 19 | 20 | 21 | 21 | 22 | 23 | 24 |
| 17 | 18 | 18 | 19 | 19 | 20 | 21 | 21 | 22 | 23 |
| 16 | 17 | 18 | 18 | 19 | 20 | 20 | 21 | 21 | 22 |

身體質量指數(BMI)

(kg/m²)

BMI < 18.5

18.5 ≤ BMI < 24

過重：24 ≤ BMI < 27

輕度肥胖：27 ≤ BMI < 30

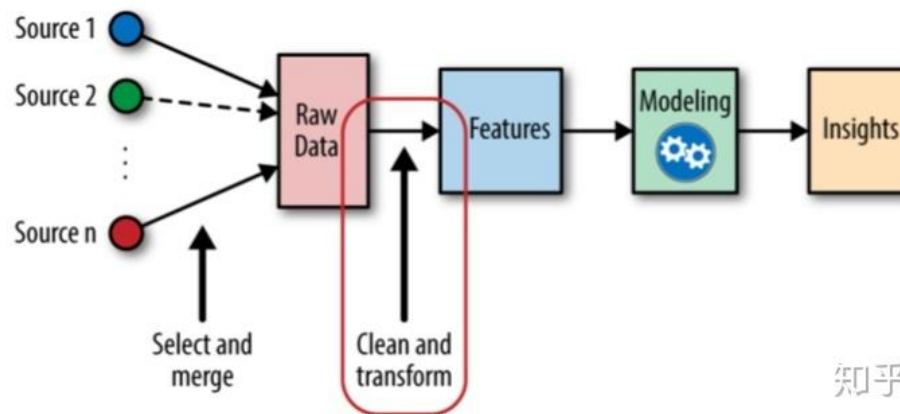
中度肥胖：30 ≤ BMI < 35

重度肥胖：BMI ≥ 35

创造新的X'

- 通过特征X，创造新的特征X'
- 衍生（升维）
 - ▶ 时间戳——月、日
- 筛选（降维）
 - ▶ 身高、体重——BMI
 - ▶ 出生年月日——年龄（浮点数）

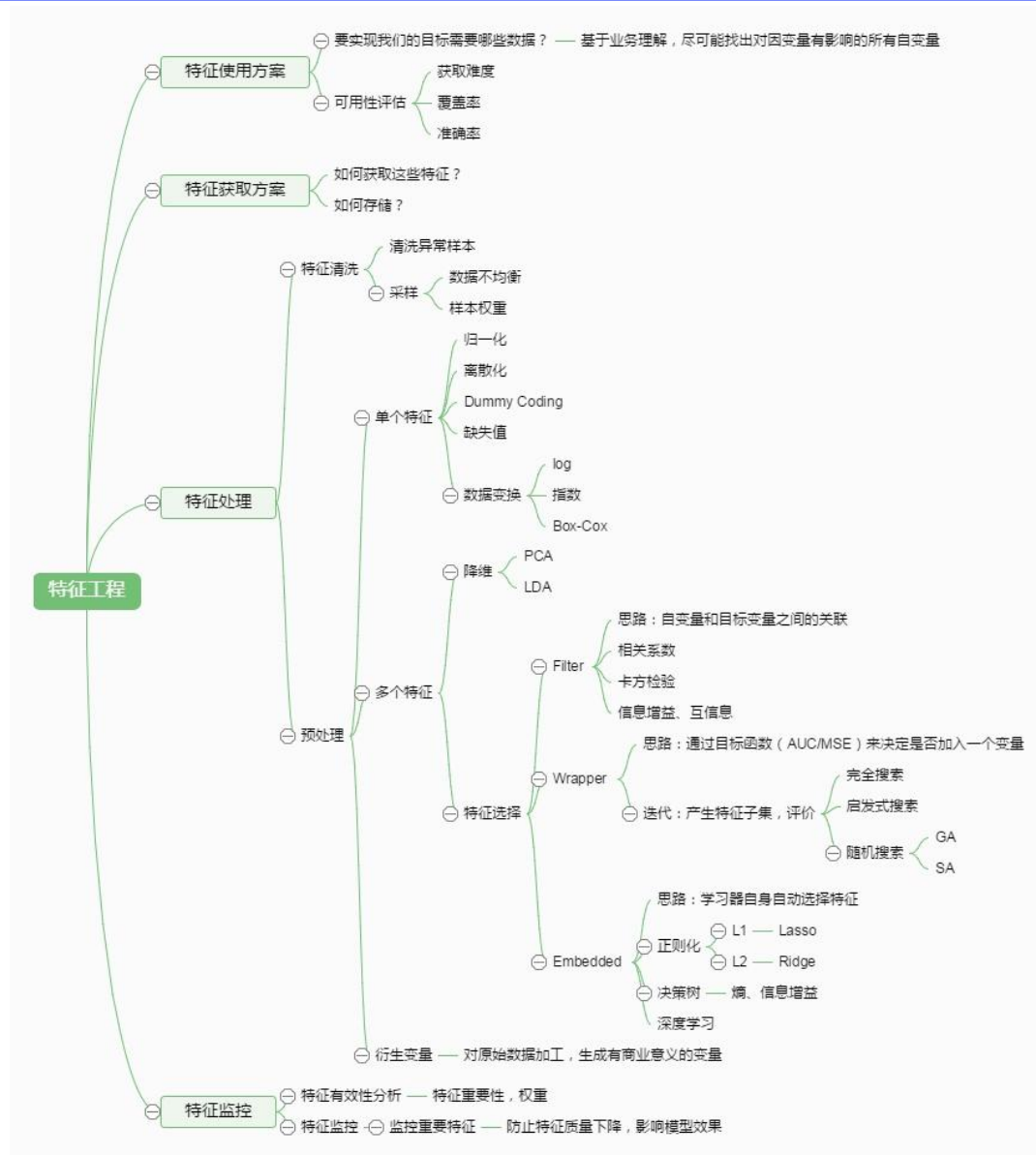
<https://www.slideshare.net/HJvanVeen/feature-engineering-72376750>



特征工程的作用

- 数据和特征决定了机器学习的上限，而模型和算法只是逼近这个上限而已
- 特征工程的目的是最大限度地从原始数据中提取特征以供算法和模型使用
- Kaggle的好成绩一般来自被广泛证明的模型和精巧设计的特征工程





<https://www.zhihu.com/question/29316149/answer/110159647>

部分特征工程实例

| 数据源大类 | 原始数据字段 | 建议特征工程方向 |
|-------|--------|---------------------|
| 支付流水 | 支付编号 | 日/周/月支付频率 |
| | 支付账户 | 来往账户数量、账户间关联图谱 |
| | 支付时间 | 最早最近支付时间、支付时段分布 |
| | 支付金额 | 支付金额总和/平均值/最大值 |
| | 支付地点 | 地点类型分布、较频繁地点 |
| | 支付目的 | 较频繁目的 |
| | 支付状态 | 支付成功/失败次数 |
| 财富管理 | 申购编号 | 申购频率 |
| | 申购时间 | 最早最近申购时间 |
| | 申购金额 | 申购金额总和/平均值/最大值 |
| | 产品类型 | 产品类型分布、产品偏好 |
| | 产品收益 | 收益总和、日均收益 |
| | 持仓金额 | 当前持仓、历史最大持仓、日均持仓 |
| | 申请编号 | 申请频率 |
| 贷款信息 | 申请时间 | 最早最近申请时间 |
| | 授信金额 | 授信金额总和/平均值/最大值 |
| | 提现金额 | 授信金额总和/平均值/最大值、提现比例 |
| | 资方类型 | 资方个数 |
| | 申请状态 | 申请通过/拒绝次数 |
| | 还款时间 | 提前结清/正常/逾期总天数 |
| | 逾期金额 | 逾期金额总和/最大值 |
| | 还款状态 | 已结清/正常/逾期笔数 |

| | | |
|-------|-------|--|
| app登录 | 登录编号 | 日/周/月登录频率 |
| | 登陆时间 | 最早最近登录时间、时段分布 |
| | 操作类型 | 操作类型分布、业务线偏好 |
| 电商流水 | 订单编号 | 当月/近3个月/近6个月/近12个月订单总数 |
| | sku编号 | 当月/近3个月/近6个月/近12个月商品总数 |
| | 订单时间 | 最早最近订单时间、近12个月有消费月份数 |
| | 订单金额 | 当月/近3个月/近6个月/近12个月订单总金额/订单最大金额/平均单笔订单金额 |
| | 订单状态 | 当月/近3个月/近6个月/近12个月实付金额占比 |
| 收货地址 | 分期标识 | 当月/近3个月/近6个月/近12个月分期订单数占比 |
| | 订单编号 | 当月/近3个月/近6个月/近12个月使用收货地址个数 |
| | 订单时间 | 收货地址使用时长 |
| | 收货地址 | 城市等级、小区档次、地址稳定性、是否涉黑 |
| | 地址类型 | 最频繁收货地址类型、工作与住宅占比 |
| 运营商信息 | 通话数据 | 当月/近3个月/近6个月/近12个月通话量/通话次数、主叫/被叫/漫游通话量/通话次数占比、通话时段分布 |
| | 流量数据 | 当月/近3个月/近6个月/近12个月流量、流量时段分布 |
| | 账单信息 | 当月/近3个月/近6个月/近12个月账单金额平均值/最大值、当月储值金额、当前欠费金额 |
| | 客户信息 | 在网时长、在网状态、名下手机号码数量/终端设备数量/终端品牌 |
| | 互联网访问 | 各类别app访问总次数/总时长/活跃天数、app类别分布、是否非法网站 |



特征工程——增加哑变量

- 当自变量 x 为**多分类变量**时
 - ▶ 季节（春夏秋冬）、职业（学生、医生、工人、公务员）、告警级别（无、普通、严重）、天气（风雷雨雪阴晴）
- 该自变量 x 对应**一个**回归系数
 - ▶ x 每变化一个单位时，所引起因变量 y 的平均变化量
- 但对于多分类变量，只有一个回归系数并不恰当
 - ▶ x 变化并非量的变化，而是分类的切换
 - ▶ 模型被不恰当地简化了
- 将自变量 x 变为**多个**（互斥的、二分类的）哑变量
 - ▶ 是春季、是夏季、是秋季、是冬季——0, 1, 0, 0
- 对应地增加了若干的系数，系数的意义也产生了变化



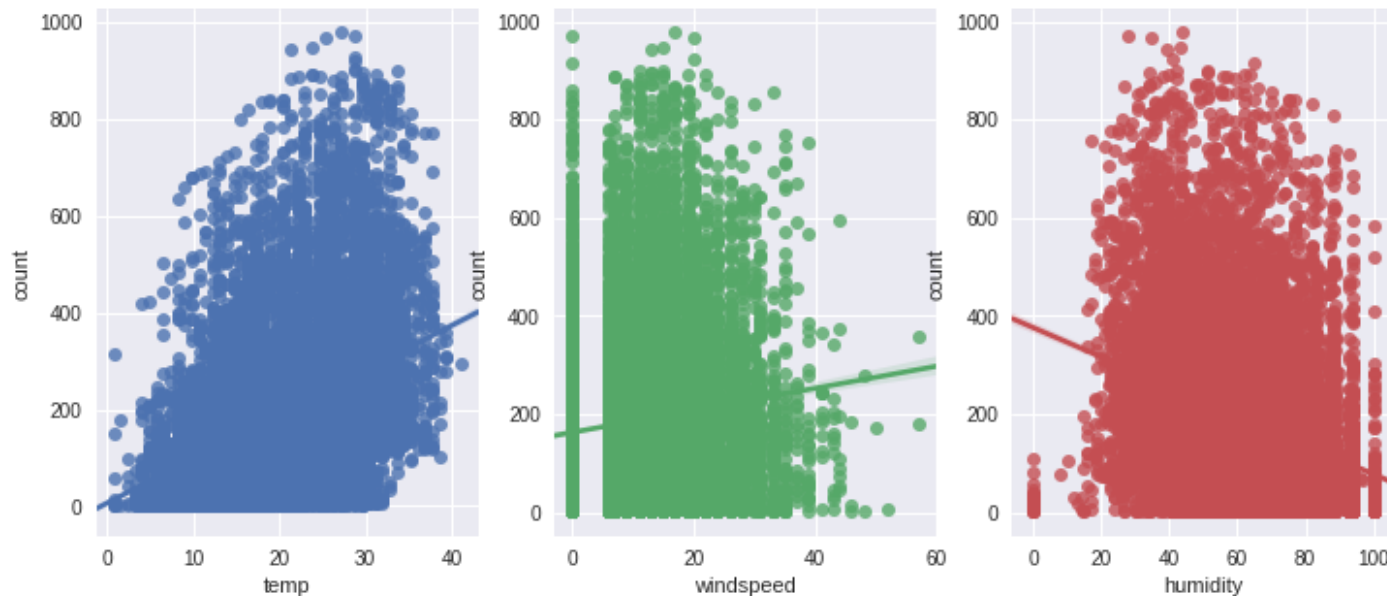
特征工程——拆分日期

- 日期类型本身难以拟合，时间戳一般不具备重复性
- 一般将日期升维
 - ▶ 拆分为年、月、日、时、分、秒等多个列
 - ▶ 可以二值化或抛弃部分列（经验或尝试）
- 也可以将多个日期、时间戳降维
 - ▶ 数据采集日期－出生日期 = 年龄
 - ▶ 类似有设备工作时长、会话时长、连接维持时间等



0值填充

- 观察风速数据的分布，存在大量零值和一个异常gap（中图最左）



- 推测是数据缺失导致，此种情况可以尝试填充0值
- 一般情况下，NaN值填充也可以参照处理

风速填充

- 使用风速不为0的数据训练一个随机森林回归模型
 - ▶ 特征：
"season","weather","humidity","month","temp","year","atemp"
 - ▶ 因变量：风速
- 预测风速为0的各个行



再次训练

- 季节、天气为哑变量
- 增加年、月、日、小时，丢弃日期列
- 对风速列进行0填充
- 训练模型并进行评估

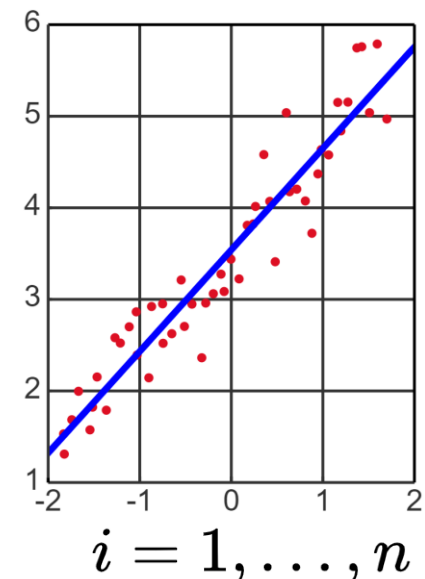


线性回归模型训练



线性回归

- 有 n 个数据样本，每个数据行具备 p 个特征，一般有 $p+1$ 个参数需要估计
- 因变量 Y 的条件均值在参数 β 里是线性的
- 古典假设：
 - ▶ 样本是在总体中随机抽取出来的
 - ▶ 因变量 Y 在实直线上是连续的
 - ▶ 残差项是独立且相同分布的
(残差和自变量相互独立)



$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \dots + \beta_p X_{ip} + \varepsilon_i,$$

$$i = 1, \dots, n$$



回归分析与最小二乘法

- 估计模型参数（各个beta），以便对（采样）数据达到最佳拟合
- 最小二乘法是比较优越的估计方法
- 如右图中四个数据采样：

▶ (1, 6), (2, 5), (3, 7), (4, 10)

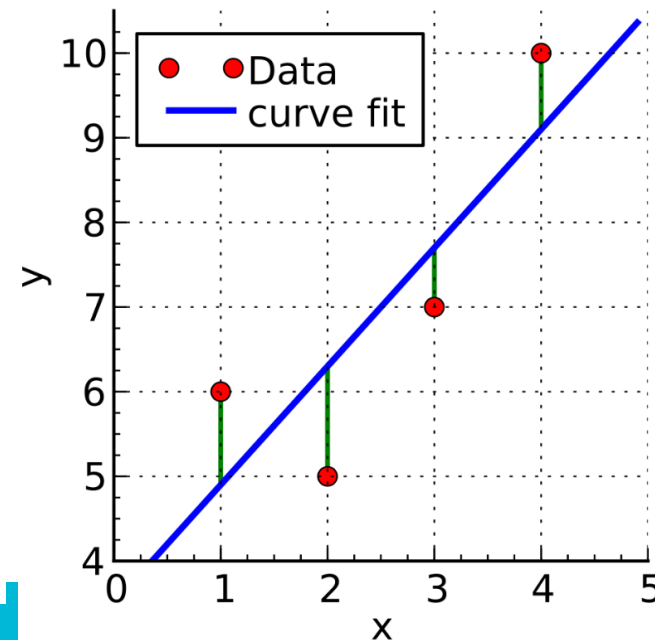
- 假设其符合线性模型 $y = b1 + b2 * x$ ，则

▶ $b1 + b2 * 1 = 6$ $b1 + b2 * 2 = 5$

▶ $b1 + b2 * 3 = 7$ $b1 + b2 * 4 = 10$

- 最小二乘法的优化目标是使方差和最小

$$S(\beta_1, \beta_2) = [6 - (\beta_1 + 1\beta_2)]^2 + [5 - (\beta_1 + 2\beta_2)]^2 \\ + [7 - (\beta_1 + 3\beta_2)]^2 + [10 - (\beta_1 + 4\beta_2)]^2.$$



最小二乘法的求解

- 对于某个优化目标

$$S(\beta_1, \beta_2) = [6 - (\beta_1 + 1\beta_2)]^2 + [5 - (\beta_1 + 2\beta_2)]^2 \\ + [7 - (\beta_1 + 3\beta_2)]^2 + [10 - (\beta_1 + 4\beta_2)]^2.$$

- 其最小值可以通过求解偏导数为0得到

$$\frac{\partial S}{\partial \beta_1} = 0 = 8\beta_1 + 20\beta_2 - 56 \\ \frac{\partial S}{\partial \beta_2} = 0 = 20\beta_1 + 60\beta_2 - 154.$$

- 解得 $b_1 = 3.5$, $b_2 = 1.4$, 也就是说 $y = 3.5 + 1.4x$ 是最优的



高维情况与解析解

- 最小二乘法的损失函数，很容易扩展到高维

$$\sum_{i=1}^n (y^{(i)} - \hat{y}^{(i)})^2$$

- 通过求偏导为零，可以得到一个解析解

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

- 其中 X 是 n 行 $p+1$ 列的特征矩阵， y 是 n 行的观测值列向量

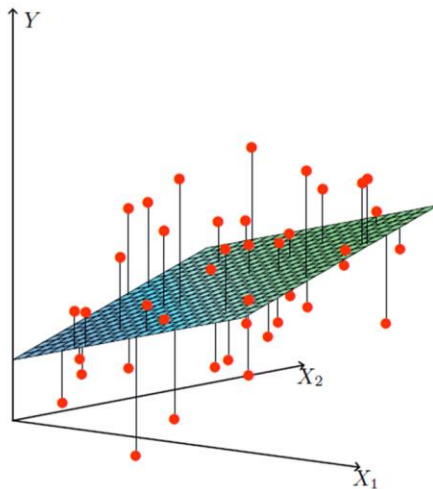


FIGURE 3.1. Linear least squares fitting with $X \in \mathbb{R}^2$. We seek the linear function of X that minimizes the loss function $\sum_{i=1}^n (y^{(i)} - \hat{y}^{(i)})^2$.



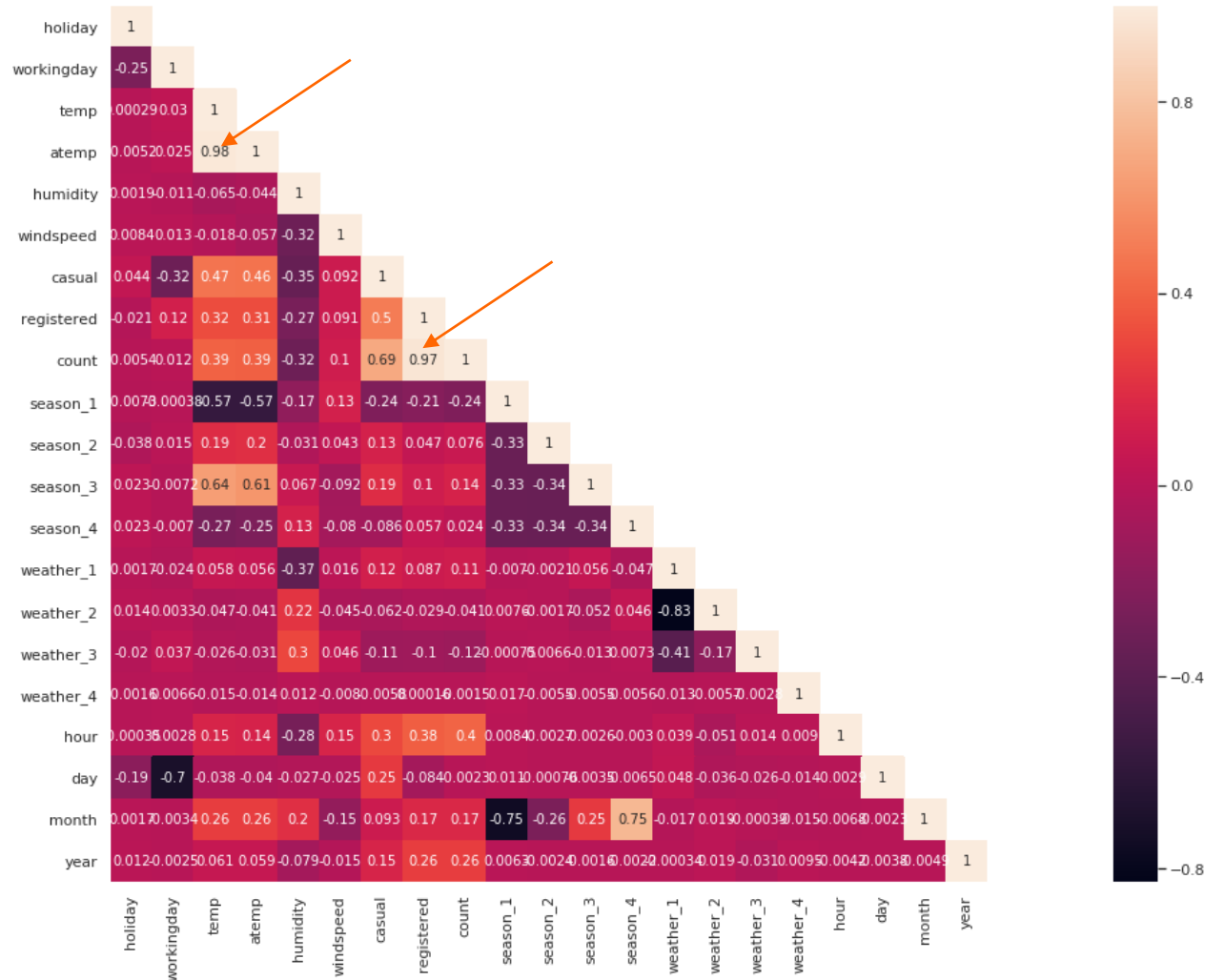
线性回归模型求解对特征的要求

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

- 其中X是n行p+1列的特征矩阵，y是n行的观测值列向量
- 则 $X^T X$ 是一个p+1的方阵，为使模型有解析解，要求此方阵可逆
- 方阵可逆的必要条件是x中各个列（特征）相互无关



数据相关性分析



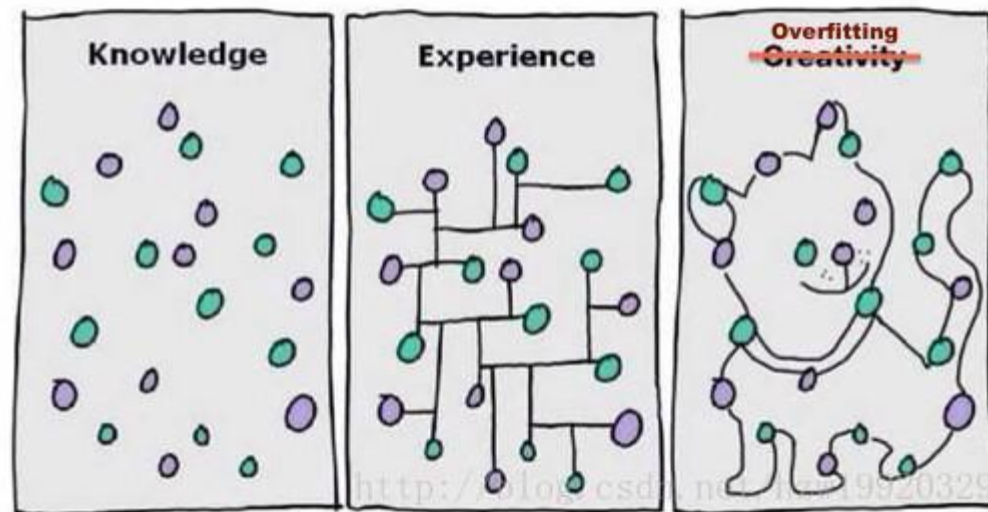
尝试使用线性回归模型

- 使用已经整理好的特征数据，训练线性回归模型
- 暂不处理有线性相关的特征
- 评估模型



正规化处理

- 解决线性回归可能存在的问题——特征线性相关等原因导致 $X^T X$ 不可逆
- 各种模型均可能存在的问题——过拟合



正规化处理思路

- 在损失函数中增加一个正则项
- 正则项和系数（ β ）有关，通过正则项来控制系数：
 - ▶ 每个系数都不是太大，这样模型不过分依赖某种特征，普适性好
 - ▶ 系数非零值个数不是太多，丢弃一些不重要的特征



岭回归

- 从不满足秩的角度看，给 $\mathbf{X}^T\mathbf{X}$ 增加一个小的常量，使其可逆

$$\beta = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

- 从避免过拟合的角度看，增加的常量实际上是在损失函数上增加了二范（L2）正则项（系数的平方和）

$$\sum_{i=1}^n (y^{(i)} - \hat{y}^{(i)})^2 + \lambda \|\theta\|_2^2$$

- 系数平方和越大，损失越大，因此模型倾向于训练出各个值都比较小的系数



Lasso回归

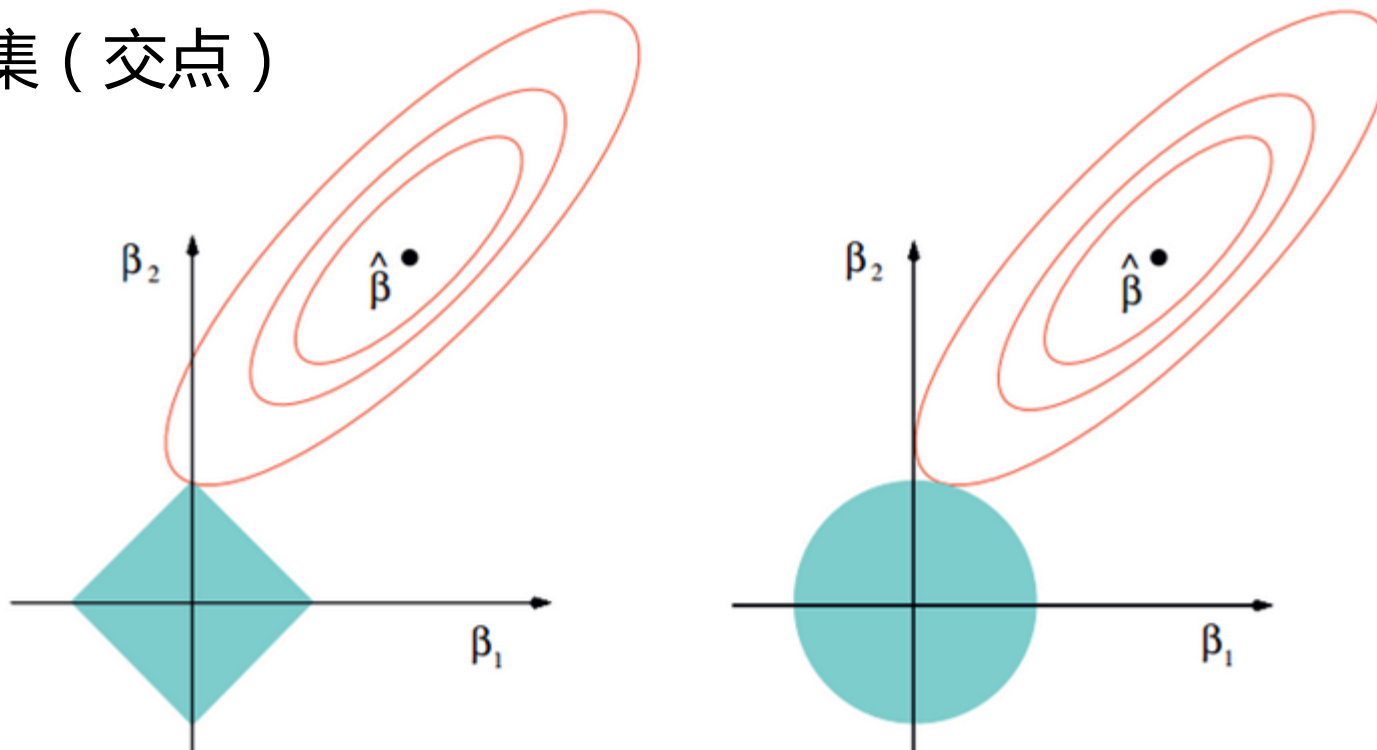
- 如果把岭回归加到损失函数的二范正则项替换为一范正则项（L1范数，系数的绝对值之和），就得到了Lasso回归

$$\sum_{i=1}^n (y^{(i)} - \hat{y}^{(i)})^2 + \lambda \|\theta\|_1$$



几何解释

- 优化过程中，不同beta取值可能得到相同的函数结果，即函数等值线
- 正则项可以看作约束空间，最优解是等值线和约束空间的交集（交点）



训练岭回归的Lasso回归

- 新增的正则项带来了一个模型参数（超参）
- 使用不同的lambda值
- 确定较好的lambda值
- 训练并评估岭回归、Lasso回归，对比普通线性回归

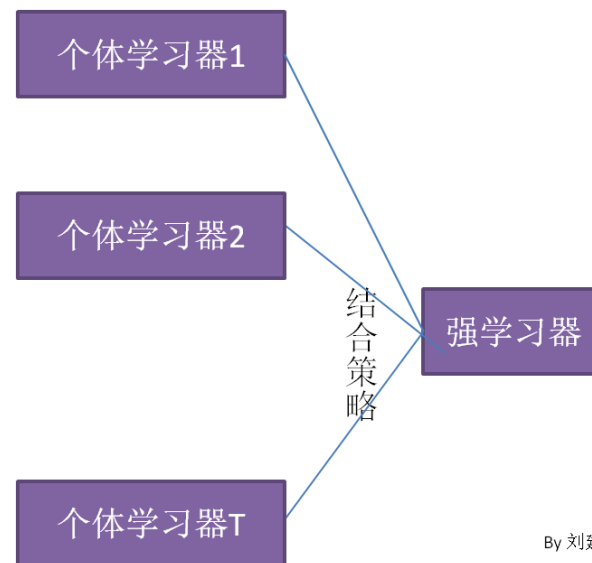


GBRT与XGBOOST



集成学习ensemble learning

- 试图训练一个强大的学习器很容易遇到瓶颈
 - ▶ 模型太复杂训练耗时且困难
 - ▶ 复杂度太高导致过拟合
 - ▶ 模型研究投入巨大
- 集成学习的思路是组合（ensemble）若干简单学习器来较好地完成任务

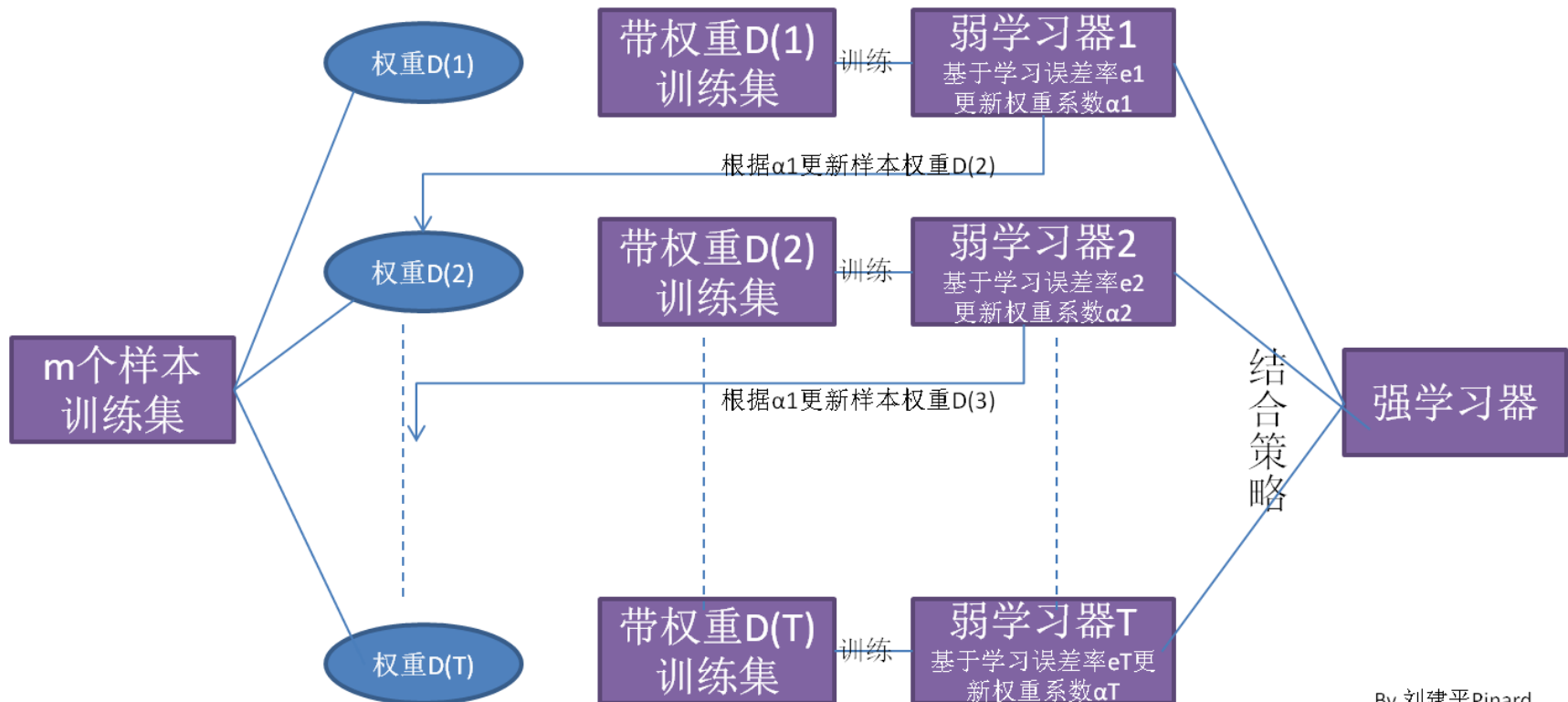


By 刘建平Pinard



Boosting

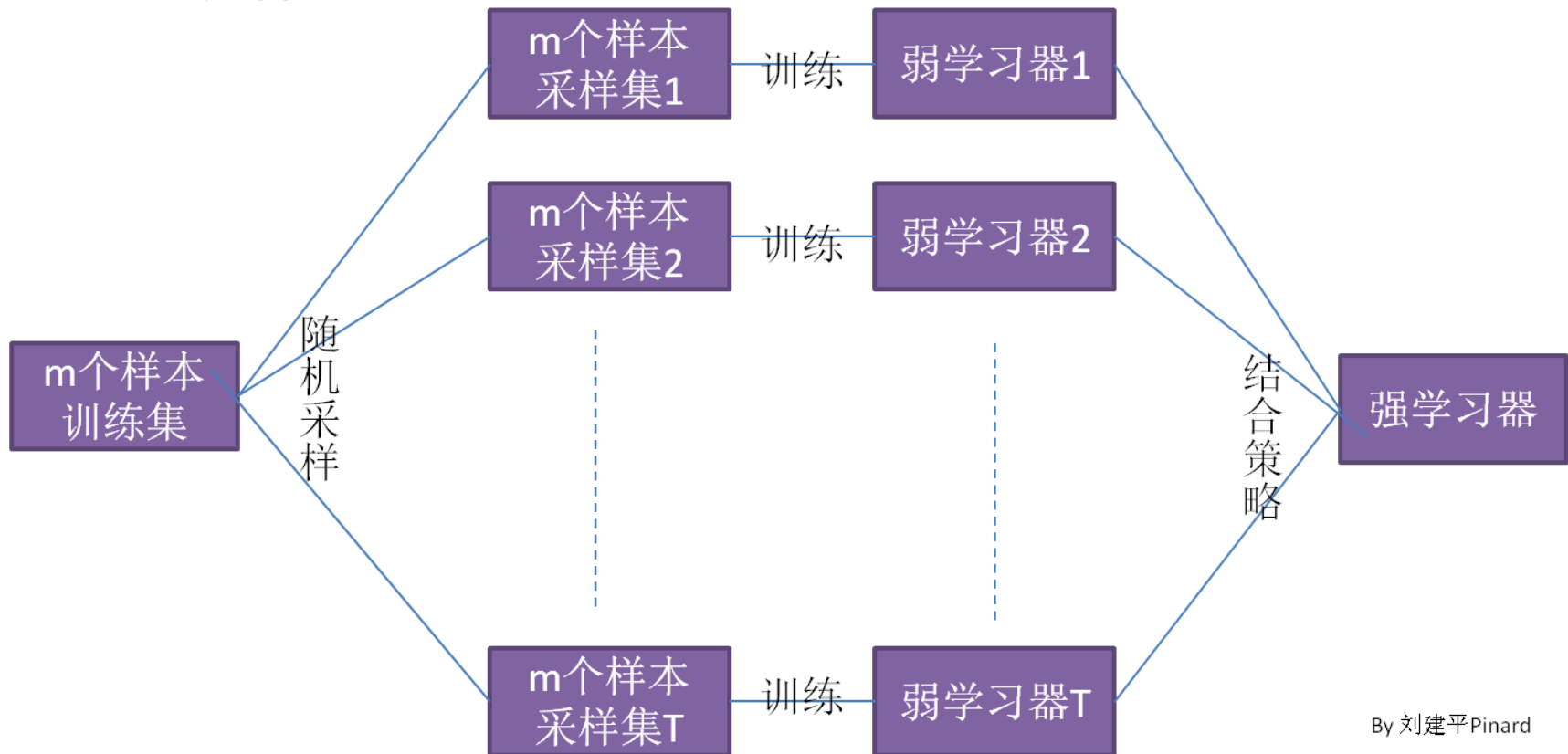
- 后面的学习器重点学习前面学习器犯错的数据
- AdaBoost、各种Boost Tree（如Gradient Boosting Tree）



By 刘建平Pinard

Bagging

- 前述Boosting需要串行训练，而Bagging可以并行
- 如随机森林等

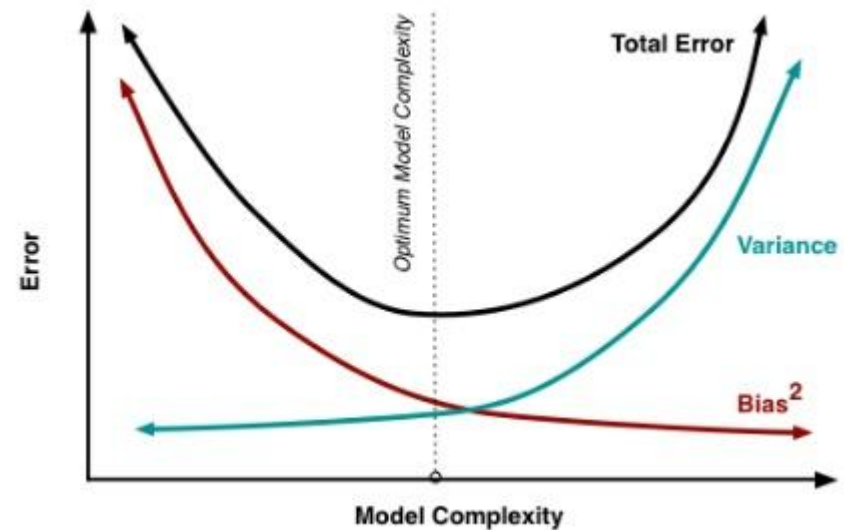
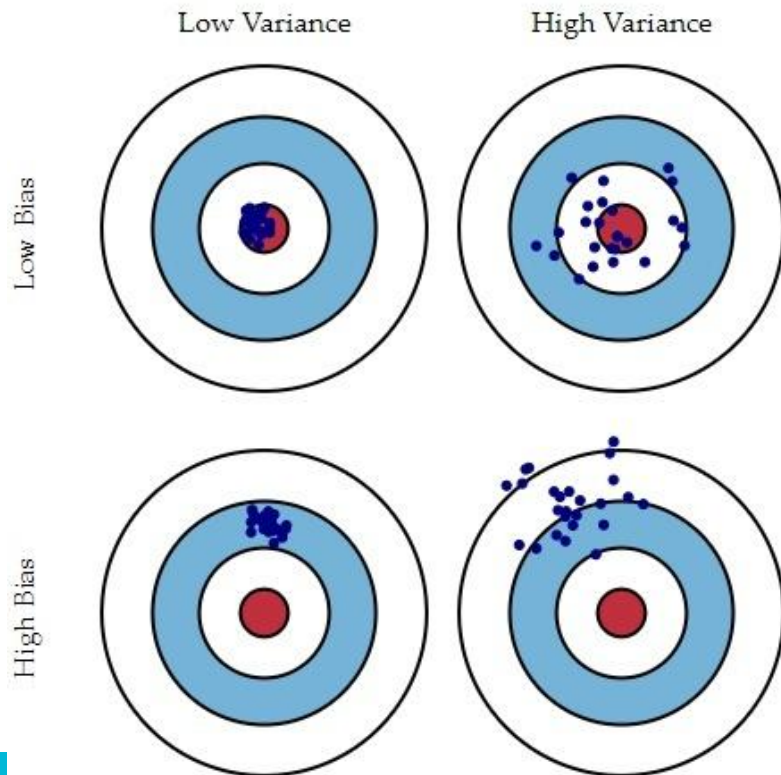


By 刘建平Pinard



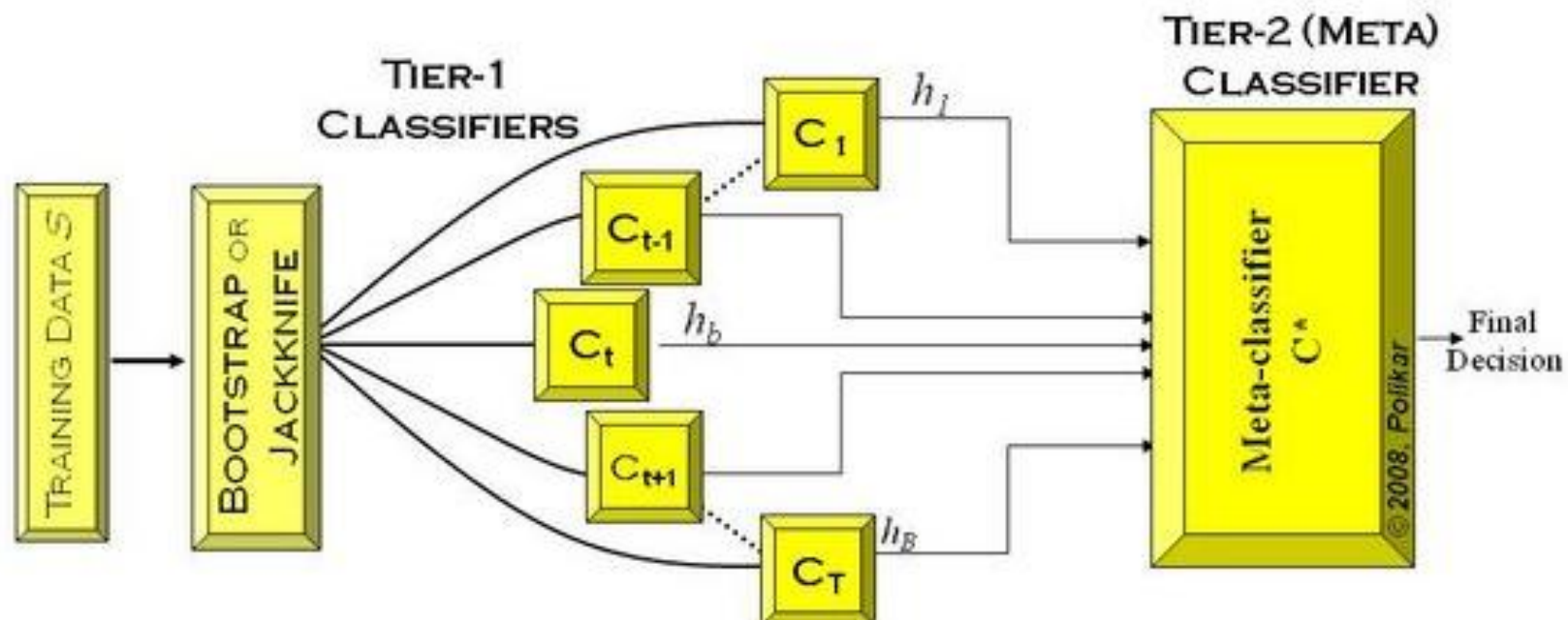
Bagging和Boosting的选择

- Bagging对多个模型进行平均，增强整体鲁棒性（var小）
- Boosting模型间不是独立的，增强整体拟合能力（bias小）
- 详细说明参见：<https://zhuanlan.zhihu.com/p/45213397>



Stacking

- 集成时可以用简单的平均、投票等方法，也可以训练一个新的模型用于集成——统称Stacking
- 分类问题的实践中Tier2一般是单层逻辑回归



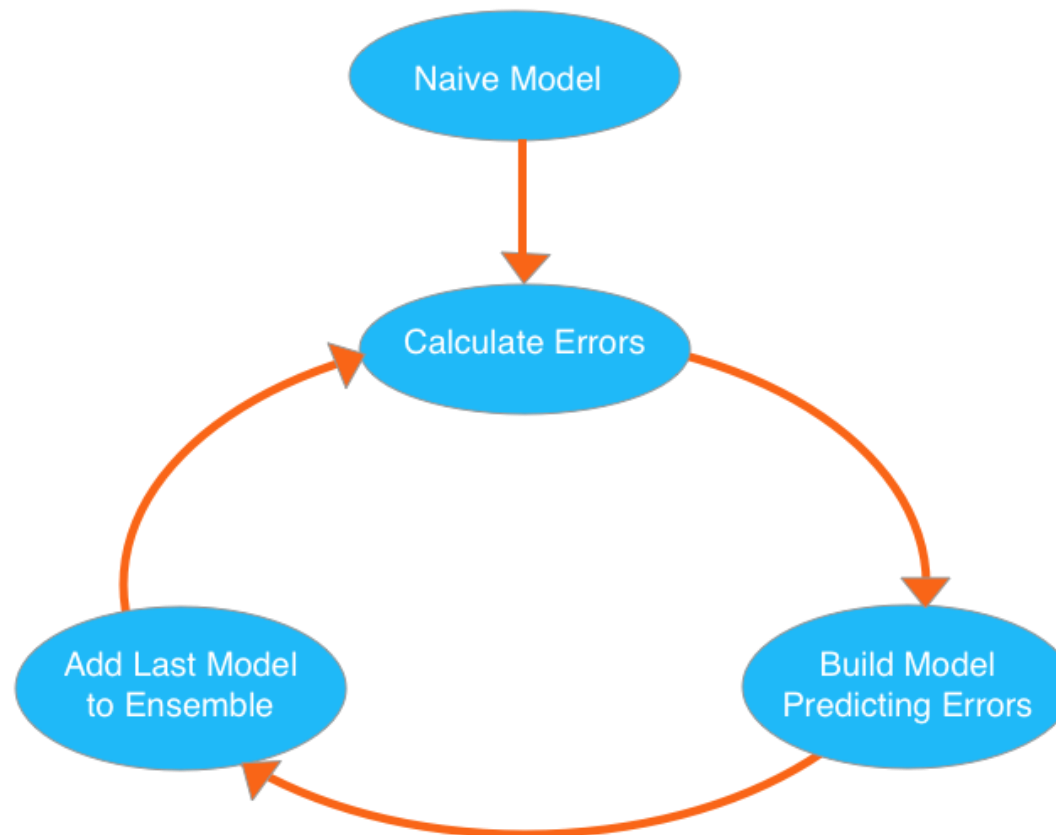
渐进梯度回归树 (Gradient Boost Regression Tree , GBRT)

- 提出之初就和SVM一起被认为是泛化能力 (generalization)较强的算法
- 近些年更因为被用于搜索排序的机器学习模型而引起大家关注
- 优点：
 - ▶ 天然就可处理不同类型的数据 (各种各样的features , 几乎可用于所有的回归问题 (线性/非线性))
 - ▶ 预测能力强
 - ▶ 对空间外的异常点处理很健壮 (通过健壮的loss函数)
- 缺点：
 - ▶ 扩展性不好 , 因为boosting天然就是顺序执行的 , 很难并行化



原理简介

Boosting Decision Tree迭代过程中，假设我们前一轮迭代得到的强学习器是 $f_{t-1}(x)$ ，损失函数是 $L(y, f_{t-1}(x))$ ，我们本轮迭代的目标是找到一个回归树模型的弱学习器 $h_t(x)$ ，让本轮的损失 $L(y, f_t(x)) = L(y, f_{t-1}(x) + h_t(x))$ 最小。也就是说，本轮迭代找到的决策树，要让样本的损失函数尽量变得更小。



简单原理——年龄预测算法为例

- 根据各特征训练第一个树
- 假设对样本A，真实年龄为18岁，预测结果为12岁
- 因 $18-12=6$ ，误差（残差）为6，借此训练第二棵树：
 - ▶ 输入特征不变，但用于计算损失的目标为6（残差）
 - ▶ 对于样本A，第二棵树输出应该尽量接近6
 - ▶ 整体输出为两棵树输出之和
- 如果第二棵树输出为5，则还有残差 $18-12-5=1$ ，训练第三棵树
 - ▶ 目标为1（残差）
 - ▶ 整体输出为三棵树输出之和



GBRT尝试

- 集成学习完整文档：

<https://scikit-learn.org/stable/modules/ensemble.html>

- 使用GBRT重新训练
- 评估结果



XGBoost简单原理

- 非常简化地说，XGBoost是GBT的一种**高效系统实现**，其与GBT比较大的不同就是目标函数的定义
- 下图为损失函数，红框为正则项，红圈为常数项
- XGBoost对l进行泰勒展开，取前三项做一个**近似**
- 最终的目标函数只依赖于每个数据点的在误差函数上的一阶导数和二阶导数

- 目标 $Obj^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + \text{constant}$

- 用泰勒展开来近似我们原来的目标

- 泰勒展开: $f(x + \Delta x) \simeq f(x) + f'(x)\Delta x + \frac{1}{2}f''(x)\Delta x^2$
- 定义: $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$, $h_i = \partial_{\hat{y}^{(t-1)}}^2 l(y_i, \hat{y}^{(t-1)})$

$$Obj^{(t)} \simeq \sum_{i=1}^n \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + \text{constant}$$

进一步了解

- XGBoost还进行了大量工程实践调优，参考
 - ▶ <https://blog.csdn.net/a819825294/article/details/51206410>
 - ▶ <https://www.zhihu.com/question/41354392/answer/98658997>
- XGBoost超参增加，训练调优相对困难
- 详细介绍和调优思路参考：
 - ▶ <https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/>



XGBoost尝试

- Python中XGBoost是一个独立的库

<https://xgboost.readthedocs.io/en/latest/>

- 调用接口和其他sklearn模型类似
- 尝试使用GridSearchCV调优超参
- 评估最终结果



参考资料目录

- Kaggle项目，RMSLE 0.3194（随机森林、Ada boost、bagging、SVR和K临近）：
<https://www.kaggle.com/rajmehra03/bike-sharing-demand-rmsle-0-3194>
- Kaggle项目，前10%，RMSLE 0.102（随机森林、线性回归、GB）
<https://www.kaggle.com/viveksrinivasan/eda-ensemble-model-top-10-percentile#Missing-Values-Analysis>
- Kaggle项目，前10%，XGB，无评分（RMSLE大概0.317）
<https://www.kaggle.com/miteshyadav/comprehensive-eda-with-xgboost-top-10-percentile>
- Kaggle讨论
<https://www.kaggle.com/c/bike-sharing-demand/notebooks>
- XGB讲解：
<https://www.kaggle.com/dansbecker/xgboost>
- Kaggle XGB库使用教程
<https://www.kaggle.com/zj0512/using-xgboost-with-scikit-learn?scriptVersionId=20724404>
- Scikit-learn GB文档
<https://scikit-learn.org/stable/modules/ensemble.html#gradient-boosting>
- GBDT，中文原理：
<https://zhuanlan.zhihu.com/p/36339161>



谢谢！

请现场提交反馈调查

