**数据预处理**

**归一化**

* 公式  
  X' = (x - min)/(max - min)  
  X" = X' \* (mx - mi) + mi
* API
* sklearn.preprocessing.MinMaxScaler(feature\_range=(mi, mx)...)
* MinMaxScaler.fit\_transform(X)
* X: numpy array格式的数据[n\_samples, n\_features]

返回值: 转换后的形状相同的array

* 总结  
  最大值和最小值容易受异常点影响，所以这种方法鲁棒性较差，只适合传统精确小数据场景。

**标准化**

* 公式  
  X' = (x - mean) / std
* API
* sklearn.preprocessing.StandardScaler()
* 处理之后， 对每列来说， 所有数据都聚集在均值为0附近， 标准差为1
* StandardScaler.fit\_transform(X)
* x: numpy array格式的数据[n\_samples, n\_features]

返回值: 转换后的形状相同的array

* 总结 适用于数据嘈杂的大数据场景

**数据降维**

**概念**

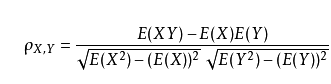
降维是指在某些限定条件下，降低随机变量（特征）的个数，得到一组“不相关”主变量的过程。

**方法**

* 特征选择
  1. Filter 过滤式：  
     (1) 方差选择法：低方差特征过滤  
     (2) 相关系数：特征与特征之间的相关程度强的过滤
  2. Embedded 嵌入式  
     (1) 决策树  
     (2) 正则化  
     (3) 深度学习
* 主成分分析

**API**

sklearn.feature\_selection

1. 低方差特征过滤
   * sklearn.feature\_selection.VarianceThreshold(threshold = 0.0)
   * 删除所有低方差特征
   * Variance.fit\_transform(X)
     + X: numpy array 格式的数据[n\_samples, n\_features]
     + 返回值: 训练集差异低于threshold的特征将被删除。默认值是保留所有非零方差特征，即删除所有样本中具有相同值的特征。
2. 相关系数
   * 皮尔逊相关系数(取值范围[-1,1]) [](https://camo.githubusercontent.com/c83bd5067b1cc8de5ac3284ac980130b452014f222691d7de46926a639d59437/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30332f355164544371794542335a67446c782e706e67)
   * from scipy.stats import pearsonr
     + x:(N,) array\_like
     + y:(N,) array\_like
     + return:(coefficient, p-value)
   * 如果相关性很高怎么办
     + 选取其中一个
     + 加权求和
     + 主成分分析

**主成分分析(PCA)**

**API**

* sklearn.decomposition.PCA(n\_components=None)
  + 将数据分解为较低维空间
  + n\_components:
    - 小数：表示保留百分之多少的信息
    - 整数：减少到多少特征
  + PCA.fit\_transform(X)
    - X:numpy array格式的数据[n\_samples, n\_features]
  + 返回值：转换后指定维度的array

**案例：探究用户对物品类别的喜好细分**

https://jupyter.dgutdev.ml/notebooks/work/personal\_folder/wjer/instacart\_from\_kaggle.ipynb

**分类算法**

**sklearn转换器和预估器**

**转换器(transformer)**

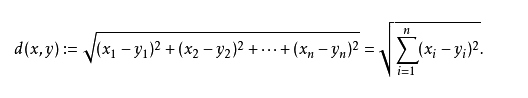
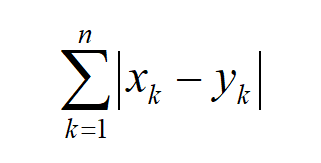
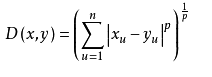
* 实例化
* 调用fit\_transformer

**预估器(estimator)**

* 实例化一个estimator
* estimator.fit(x\_train, y\_train)
* 模型评估
  + y\_predict = estimator.predict(x\_test)  
    then check if y\_test == y\_predict
  + 计算准确率  
    estimator.score(x\_test, y\_test)

**KNN算法(K-近邻算法)**

**什么是K-近邻算法**

* 存在一个样本数据集合，也称为训练样本集，并且样本集中每个数据都存在标签，即我们知道样本集中每一数据与所属分类对应的关系。输入没有标签的数据后，将新数据中的每个特征与样本集中数据对应的特征进行比较，提取出样本集中特征最相似数据（最近邻）的分类标签。
* 确定最近距离
  + 欧式距离 [](https://camo.githubusercontent.com/e225a001a5f90b3b8536905e66d88399e6150880972d9a6768423b696605c42a/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30342f6561557366397953335850646e45562e706e67)
  + 曼哈顿距离 [](https://camo.githubusercontent.com/40574a69b801330600c620adb2f81bf960020d461a0733e369f275cb20b453ef/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30342f424d684a6c51715544797a70336e5a2e706e67)
  + 明可夫斯基距离  
    [](https://camo.githubusercontent.com/4a256b4f7c2fa73f90aad9b26599326cde2783f06e04a0326fb9d45f19b7e5c0/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30342f6b53456650566278485a74516f55712e706e67)
* K的取值  
  若K过小，容易受异常点影响；若K过大，受样本不均衡的影响。
* 使用KNN前的数据处理 无量纲化的处理：标准化

**K-近邻算法API**

* sklearn.neighbors.KNeighborsClassifier(n\_neighbors=5, algorithm='auto')
  + n\_neighbors: int,可选(默认=5)
  + algorithm: {'auto', 'ball\_tree', 'kd\_tree', 'brute'}

**案例：鸢尾花种类预处**

1. 获取数据
2. 数据集划分
3. 特征工程 标准化
4. KNN
5. 模型评估

**总结**

* 缺点：K值难确定；内存开销大

**模型选择和调优**

**交叉验证**

* 基本思想就是将原始数据（dataset）进行分组，一部分做为训练集来训练模型，另一部分做为测试集来评价模型。

**超参数搜索-网格搜索**

* 一种调参手段；穷举搜索：在所有候选的参数选择中，通过循环遍历，尝试每一种可能性，表现最好的参数就是最终的结果。其原理就像是在数组里找最大值。
* sklearn.model\_selection.GridSearchCV(estimator,param\_grid,cv)
  + estimator：估计器对象
  + param\_grid：估计器参数，参数名称（字符串）作为key，要测试的参数列表作为value的字典，或这样的字典构成的列表
  + cv：整形，指定K折交叉验证
  + fit()：输入训练数据
  + score()：准确率
  + best\_score\_：交叉验证中测试的最好的结果
  + best\_estimator\_：交叉验证中测试的最好的参数模型
  + best\_params\_：交叉验证中测试的最好的参数
  + cv\_results\_：每次交叉验证的结果

**鸢尾花案例增加K值调优**

**案例：预测facebook签到位置**

* 由于找不到数据，没实践

**总结**

**朴素贝叶斯算法**

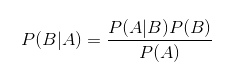
**什么是朴素贝叶斯分类方法**

* 对于给出的待分类项，求解在此项出现的条件下各个类别出现的概率，哪个最大，就认为此待分类项属于哪个类别。

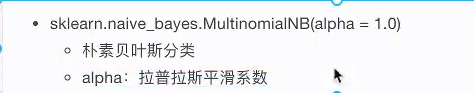
**概率基础**

**联合概率、条件概率与相互独立**

**贝叶斯公式**

* [](https://camo.githubusercontent.com/90ba72cf00060d983d75ae335ea3c5e580dd9556b4b976c345e3a8d05715e4fd/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30352f6d764269734a43667275335a6b41772e706e67)

**API**

* [](https://camo.githubusercontent.com/ac5dcd8ce8b160cc7b5b67c716a70e05d1e14128cd314fcd50d4dee4e2f937fb/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30352f71364638767a4a55745472667844422e706e67)

**案例：20类新闻分类**

1. 获取数据
2. 划分数据集
3. 特征工程 文本特征抽取
4. 朴素贝叶斯预估器流程
5. 模型评估

**朴素贝叶斯算法总结**

* 对缺失数据不敏感， 准确度高，常用于文本分类
* 对特征有关联的样本效果不是很好

**总结**

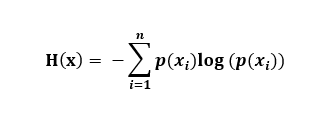
* 文本分类

**决策树**

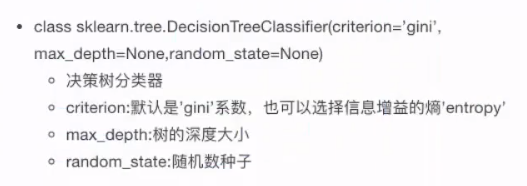
**认识决策树**

* 决策树(Decision Tree）是在已知各种情况发生概率的基础上，通过构成决策树来求取净现值的期望值大于等于零的概率，评价项目风险，判断其可行性的决策分析方法，是直观运用概率分析的一种图解法。

**决策树分类原理详解**

* 信息论基础
  + 信息是用来消除随机不定性的东西
  + 信息熵  
    [](https://camo.githubusercontent.com/7221c2d627493d629587844ca822bcd5a5a7b9b56d91062183a71e1bfdc550e1/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30352f5163654c6c70524f6b6d75666e43722e706e67)
  + 条件熵
  + 信息增益 = 信息熵 - 条件熵

**决策树API**

* [](https://camo.githubusercontent.com/7d3216abc1b017191e71f451863c174ddef23ff1499a9b394e637679c65c66f4/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30352f454a6278563738476d326c545852492e706e67)

**案例：泰坦尼克号乘客生存预测**

* <http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/titanic.txt>
* 获取数据
* 数据处理 缺失值处理 特征值->字典类型
* 准备好特征值和目标值
* 划分数据集
* 特征工程：字典特征抽取
* 决策树预估器流程
* 模型评估

**决策树可视化**

* export\_graphviz(estimator, out\_file="iris\_tree.dot", feature\_names=iris.feature\_names)
* <http://webgraphviz.com/>

**决策树总结**

* 优点：可视化
* 缺点：容易过拟合

**总结**

**随机森林**

**什么是集成学习方法**

* 使用一些（不同的）方法改变原始训练样本的分布，从而构建多个不同的分类器，并将这些分类器线性组合得到一个更强大的分类器，来做最后的决策。

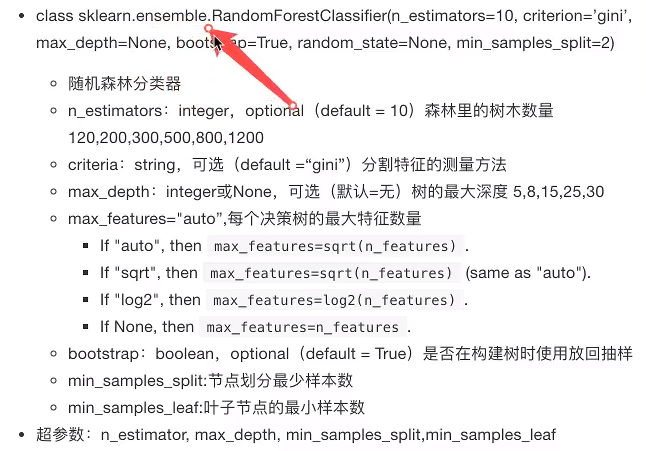
**什么是随机森林**

* 随机森林就是通过集成学习的思想将多棵树集成的一种算法，它的基本单元是决策树，而它的本质属于机器学习的一大分支——集成学习（Ensemble Learning）方法。

**随机森林原理过程**

* 随机
  + 特征随机
  + 训练集随机 bootstrap抽样

**API**

[](https://camo.githubusercontent.com/a11d4c33d349539215d4cd5ed35b846d6e5602275a0a3f1942799da9aa1ac23d/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30352f5343316c4d4f563447335a6b7962682e706e67)

**随机森林预测案例**

**总结**

* 极好的准确率
* 有效的运行在大数据集上， 处理具有高维特征的样本，不需要降维
* 能够评估各个特征在分类问题上的重要性

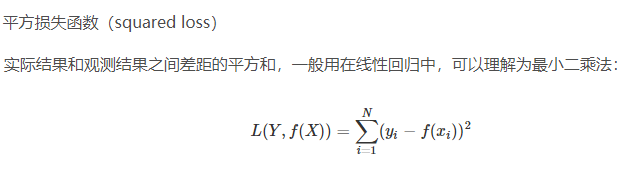
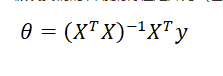
**回归与聚类算法**

**线性回归**

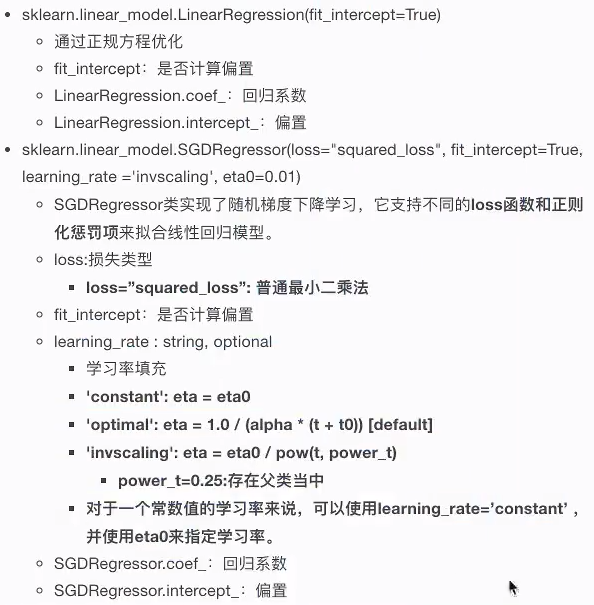
**线性回归的原理**

* 在统计学中，线性回归(Linear Regression)是利用称为线性回归方程的最小平方函数对一个或多个自变量和因变量之间关系进行建模的一种回归分析。

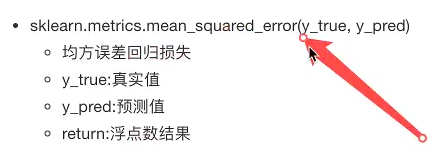
**线性回归的损失和优化原理**

* 目标：求模型参数，一开始不知道，只能随意假设一组参数，不断地靠近真正的一组参数
* 如何衡量真实值与实验值的差异：损失函数/cost/成本函数/目标函数  
  [](https://camo.githubusercontent.com/6004c0fee8cea0089cca25e4d8ed06085f0f2c5e6dc4c0ba770afb6524167329/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30362f47737944767a53644d436a494c54482e706e67)
* 优化损失
  + 优化方法
    - 正规方程  
      [](https://camo.githubusercontent.com/bc26a08093883e2f4a139cd6891fd35541c3b8f675a230e715c69f67c9475d82/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30362f4c7a4b7473633155504a4275465a762e706e67)
    - 梯度下降

**线性回归API**

* API  
  [](https://camo.githubusercontent.com/00e696b44ddfbd68c28e8d2eef68e2e046fffd1faa72f2f76763972c27a1a0b9/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30362f695762776c4b50756f4759633543552e706e67)

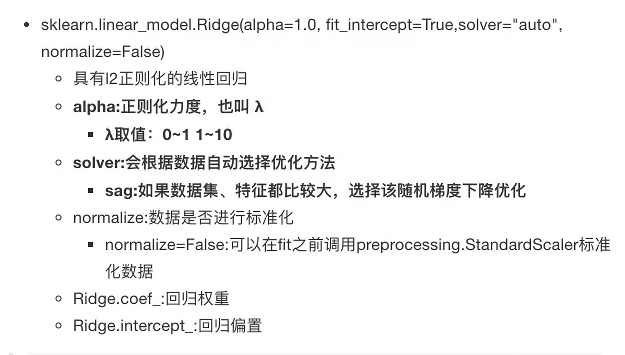
**波士顿房价预测**

1. 获取数据集
2. 划分数据集
3. 特征工程：无量纲化-标准化
4. 预估器流程
5. 模型评估 [](https://camo.githubusercontent.com/cbb7ec6e3ca080d030e078929a27958c33ed3ab2760a47ec283d5e8ffae21696/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30362f336142595765555879766f637378682e706e67)

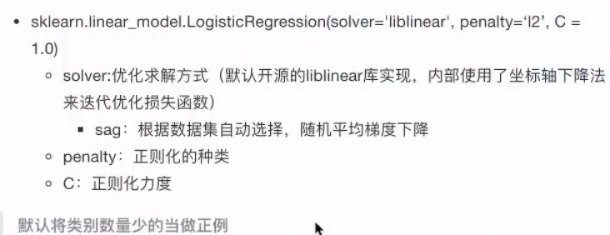
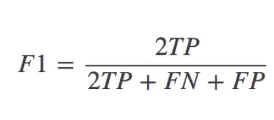
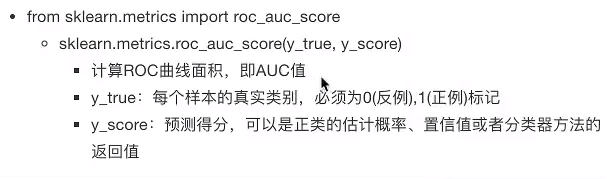
**欠拟合与过拟合**

* 参考博客：<https://www.cnblogs.com/zhhfan/p/10476761.html>

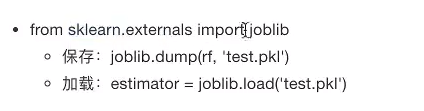
**线性回归的改进——岭回归**

* 岭回归就是带有L2正则化的线性回归
* API  
  [](https://camo.githubusercontent.com/1c351551c1dbbb34a2aab776b58fcbc664294bf846275d6032ada23588fecac0/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30372f706971666e6d6f4a654935474868632e706e67)

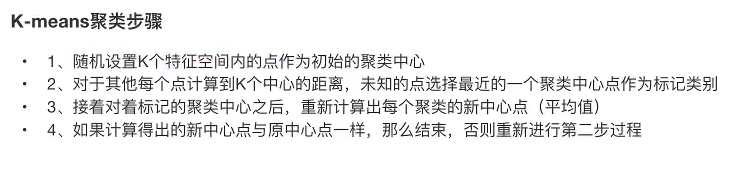
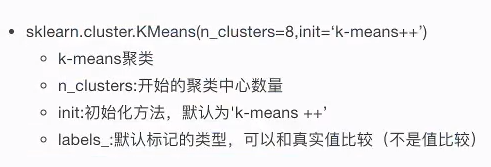
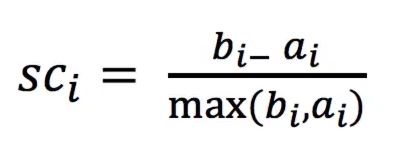
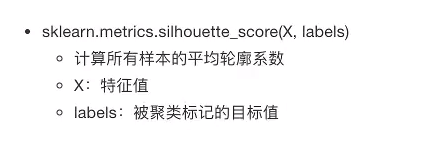
**分类算法——逻辑回归与二分类**

* 线性回归的输出就是逻辑回归的输入
* sigmoid函数
* API  
  [](https://camo.githubusercontent.com/fe2f73cce7da38c352b9d4977204f1055b8a7d33ba28f39203132808dadbba34/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30372f6547744238537935557572646d676b2e706e67)
* 良/恶性乳腺癌预测
  + <http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/>
  + 流程
    1. 获取数据 加上names
    2. 数据处理 处理缺失值
    3. 数据集划分
    4. 特征工程 无量纲化-标准化
    5. 逻辑回归预估器
    6. 模型评估
* 精确率与召回率
  + 混淆矩阵
  + 精确率
  + 召回率
  + F1-score  
    [](https://camo.githubusercontent.com/92c2042345dbfca7444a00ecd629fed6c077b48388276e678fc4c75a23a3c554/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30372f4550386e58655a5467534c564473552e706e67)
  + API  
    [](https://camo.githubusercontent.com/3b985a6ae74d963ab8bf885ac5b3219eebcc48886c5447fb4c5ab02c4d73e571/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30372f79764a3670356a4e53676b734258432e706e67)
* ROC曲线和AUC指标
  + TPR = TP/(TP+FN) 召回率
  + FPR = FP/(FP+TN)
  + API [](https://camo.githubusercontent.com/55404364079557ea5b07f966c60340c257a236a46b6ebb9b8bec163353abab7f/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30372f697248474a54584c63736f623750552e706e67)

**模型保存与加载**

* API  
  [](https://camo.githubusercontent.com/2e1c9208d611c7ec4ead4b7f785da3f37ab6a529085a42093212ac05352bef4f/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30372f5644664632366d76516779476958432e706e67)

**无监督学习——K-means算法**

* 没有目标值-无监督学习
* 无监督学习包含的算法
  + 聚类
    - K-means(K均值聚类)
  + 降维
    - PCA
* K-means  
  [](https://camo.githubusercontent.com/03b071ca5f9adc05ed0746416ca83171d2ff08904737cc24b659fef297c0630e/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30372f6342645a4658696b533659754c76322e706e67)
* API  
  [](https://camo.githubusercontent.com/b445beaab711e70edbd5aac8bd87080480b72113062351f3e592ffea76a5cf31/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30372f556a494e4161747764785465634b572e706e67)
* 轮廓系数  
  [](https://camo.githubusercontent.com/2645029d63bc7da3cd0d41fd2ee5a5e621bb36792c39fa2a198667c5aa8bbb8b/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30372f6a614e71577268736b5434426f70332e706e67)
* 轮廓系数API  
  [](https://camo.githubusercontent.com/4bae36fca6c710aac3e4ef86fd23d05bd44e78023cbefaac39c14356a6e97aee/68747470733a2f2f692e6c6f6c692e6e65742f323031392f31302f30372f57723952545967344d7336765561322e706e67)

**总结**

* 线性回归
* 过拟合与欠拟合
* 岭回归
* 逻辑回归
* 保存与加载模型
* K-Means