关联分析（association analysis）或者关联规则学习（association rule learning）：在大规模数据集中寻找有趣关系的任务，包含两种形式，频繁项集（frequent item sets）和关联规则（association rules）

一个项集的支持度（support）：数据集中包含该项集的记录所占的比例

可信度或者置信度（confidence）

Apriori原理，如果一个项集是频繁的，那么他的所有子集也是频繁的，如果一个项集是非频繁的，那么他的所有超集也是非频繁的。

从逻辑研究的角度上来讲，箭头左边的集合称作前件，箭头右边的集合称为后件。

寻找关联规则从可信度的角度出发。寻找关联规则，首先从一个频繁项集开始，创建一个规则列表，然后循环此过程。

FP-growth算法，运算速度通常比Apriori算法高两个数量级，FP代表频繁模式（Frequent Pattern），需要对原始数据集扫描两遍，第一遍对所有元素相的出现次数进行计数，第二遍只针对扫描中的那些频繁元素构建FP树，在构建FP树时，读入每个项集并将其添加到一条已经存在的路径之中。

奇异值分解（Singular Value Decomposition，SVD），原始矩阵Data为m行n列，可以分解成三个矩阵U、E、V，分别是m\*m,m\*n,n\*n,矩阵E为对角矩阵，这些对角元素就称为奇异值（singular value），取奇异值中较大的部分实现对原始数据集的简化过程，一个典型的做法就是保留矩阵中90%的能量信息。奇异值就是Data\*DataT特征值的平方根。

相似度计算方式：

1. 相似度=1/(1+距离)，距离可以使欧氏距离，也可以是其他形式的距离
2. 皮尔逊相关系数（Pearson correlation）numpy中是corrcoef（），皮尔逊相关系数的取值范围是-1到1，规划到0到1之间为0.5+0.5\*corrcoef（），可以用numpy中的corrcoef（）函数进行计算
3. 余弦相似度（cosine similarity），取值范围也是-1到1之间，A、B两向量之间的余弦相似度计算：A\*B/(|A|\*|B|)，范数可以用numpy中的linalg.norm()函数进行计算

通常用于推荐引擎评价的指标是最小均方根误差（Root Mean Squared Error， RMSE），他首先计算均方误差的平均值然后去其平方根。

海量数据处理的解决方案：Hadoop，采用java语言编写，支持在大量机器上分布式处理数据，Hadoop是MapReduce框架的一个免费开源实现，MapReduce在大量节点组成的集群上运行，单个作业被分成很多小份，输入数据也被切片分发到每个节点，每个节点只在本地数据上做运算，对应的运算代码成为mapper，这个过程被称作map，每个mapper的输出通过某种方式组合，排序后的结果在被分成小份分发到各个节点进行下一步处理工作，第二部的处理阶段成为reduce阶段，对应的运行代码称为reducer，reducer的输出就是程序的最终执行结果。（在任何时候每个mapper或者reducer之间都不进行通信，每个节点只处理自己的事物，同时每份mapper的输入数据会同时分发到多个节点形成多份副本，用于事务的失效处理）

宕机=死机

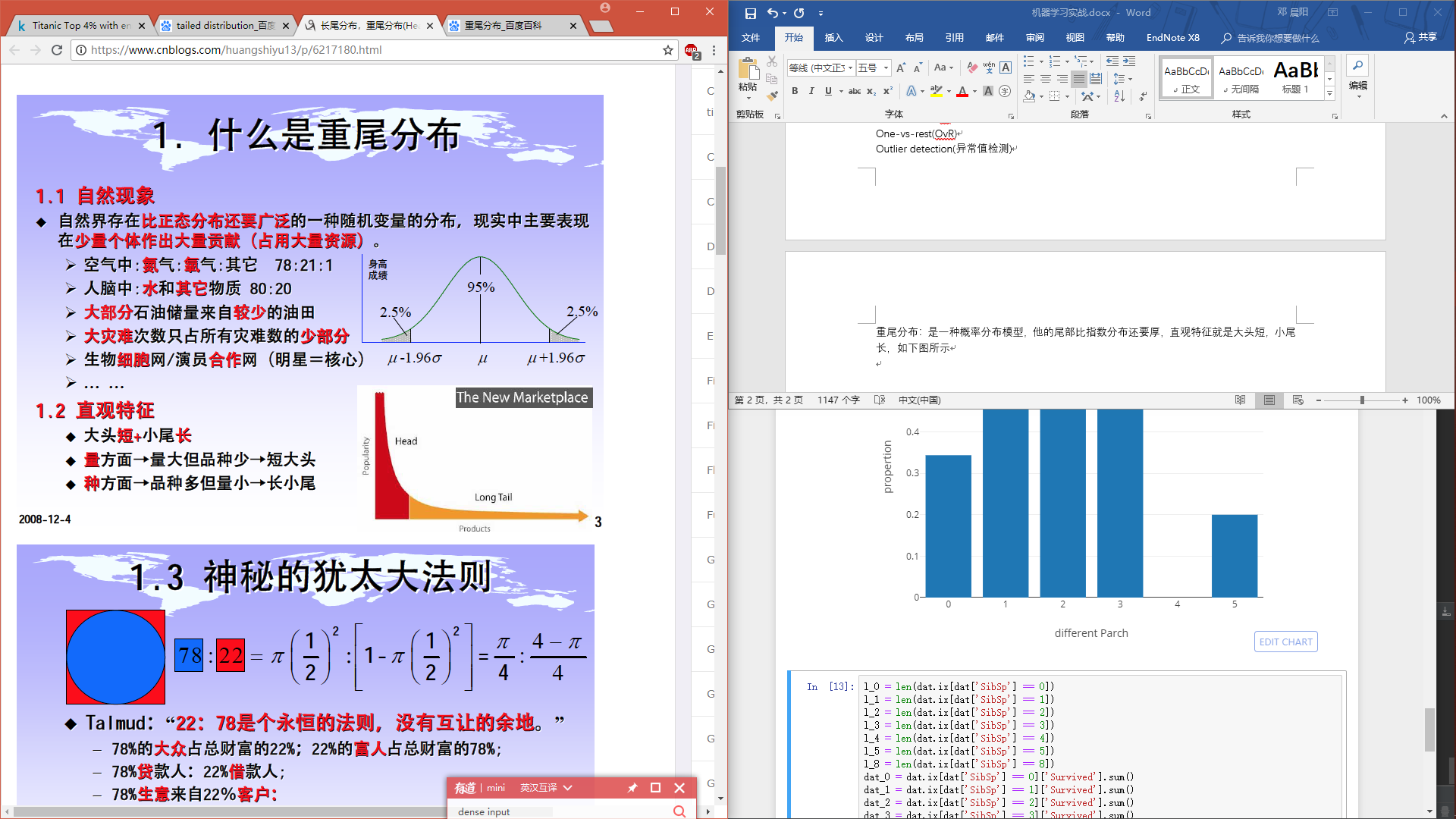
Hadoop是MapReduce框架的一个实现、

斐波那契数列是一个递归数列，除了第一个和第二个数以外，任意一个数都可以由前两个数相加得到

Python中带有yield的函数会被解释器视为一个generator，在迭代的过程之中会自动继承上一次迭代的的值

One-vs-rest(OvR)

Outlier detection(异常值检测)

重尾分布：是一种概率分布模型，他的尾部比指数分布还要厚，直观特征就是大头短，小尾长，如下图所示

重尾分布又可以分为两个子类型，分别是长尾分布（long-tailed distribution）以及次指数分布（subexponential distribution）

如果数据中存在缺失，可以使用中位数来代替，原因是中位数产生的影响不会特别大，来自于kaggle的blog，没有出处，url= https://www.kaggle.com/yassineghouzam/titanic-top-4-with-ensemble-modeling

偏度（skewness），三阶矩，是统计数据分布偏斜方向和程度的度量，是统计数据分布非对称程度的数字特征，也可以称为偏态、偏态系数。

如果偏度特别大的数据分布即使标准化之后对于比较大的值还是会产生很重的权重，所以这个人用log函数处理了一下，url = <https://www.kaggle.com/yassineghouzam/titanic-top-4-with-ensemble-modeling>

Violin plot，小提琴图，展示了任意位置的数据密度

机器学习中训练出的模型的超参数包括Ridge回归以及Lasso回归中的正则项中的正则参数（regularization parameter），又或者是决策树中需要设置的树的深度和叶子数，又或者是支持向量机中需要设置的分类错误的惩罚因子还有核函数的参数，还有可能是模型训练阶段产生的在为了使损失函数最小话过程中经常需要用的参数例如学习率、初始点等等。

超参数调优算法的实现：

1. 格搜索（Grid Search），顾名思义，格搜索便是将超参数的取值范围划分成一个个格子，然后对每一个格子所对应的值进行评估，选择评估结果最好的格子所对应的超参数值。例如，对于决策树叶子节点个数这一超参数，可以将值划分为这些格子：10，20，30，…，100，…；又如正则化因子这一超参数，一般使用指数值，那么可以划分为：1×e-5，1×e-4，1×e-3，…，1。有时可以进行猜测对格子进行搜索去获得最优的超参数。如，当从第一个开始，发现效果较差，第二个好了一点，那么可以第三个可以取最后一个。格搜索较为简单并且可以进行并行化。
2. 随机搜索（Random Search），在论文 “RandomSearch for Hyper Parameter Optimization” (Bergstra and Bengio)中，已经验证了随机搜索是一个简单而有效的方法。它是格搜索的变种。相比于搜索整个格空间，随机搜索只对随机采样的那些格进行计算，然后在这中间选择一个最好的。因此随机搜索比格搜索的代价低。随机搜索有个缺点，即其可能找不到最优的点。但是前面的那篇论文已经证明，随机采样60个点的性能已经足够好了。从概率的角度来说，对于任何的分布的样本空间若存在最大值，那么随机采样60个点中的最大值位于整个样本空间top 5%的值的集合中的概率达到95%。
3. 智能搜索（Smart Search），除了前面的两种搜索算法，还可以利用智能搜索算法，但是相对于前面的两种方法，智能搜索算法最大的缺点便是不能并行化。它的处理过程是一个序列，并只处理一部分候选点，然后对当前点进行评估，决定下一个点。智能搜索的目的是只对一部分点进行评估从而节省调优时间。

可以看出，智能搜索需要时间去计算下一个评估的点，于是相对于前面的方法，可能需要更多的时间。因此只有在对点进行评估所用的时间大于决定下一个需要评估的点的时间时才有意义。当然智能搜索算法也需要自己的超参数，因此也需要调优。有时好的智能搜索算法超参数可以确保智能搜索快于随机搜索。

目前有超参数三个智能调优算法：derivative-free optimization, Bayesian optimization和random forest smart tuning。derivative-free优化算法采用启发式来决定下一个计算的点；Bayesian和random forest优化算法都是创建一个响应函数曲面模型，由模型决定下一步需要计算的点。Jasper Snoek[x9] 等使用高斯过程对响应函数进行建模。Frank Hutter[x10] 等使用回归随机森林来近似这个响应曲面。Misha Bilenko[x11] 等使用Nelder-Mead来进行超参数调优。

不同类别区域边界叫做决策面（decision boundaries），判别式函数（discriminant function），线性判别分析（Linear Discriminant Analysis，LDA），二次判别分析（Quadratic discriminant analysis，QDA），LDA线性判别式分析，也叫做Fisher线性判别分析（Fisher Linear Discriminant，FLD），通过将高维的模式样本投影到最佳鉴别矢量空间，以达到抽取分类信息和压缩特征空间维数的效果

排名前十的数据挖掘算法，C4.5决策树，K-均值（K-mean），支持向量机（SVM），Apriori，最大期望算法（EM），PageRank算法、AdaBoost算法、k-近邻算法（kNN）、朴素贝叶斯算法（NB），分类回归树（CART）

不同的分类器组合结果称为集成算法（ensemble method）或者元算法（meta-algorithm），随机森林（random forest）是bagging算法中的一种，也可以称为Random Trees。

构建决策树的方法有很多种：其中一种是ID3方法，如机器学习实战中第三章所采用的方法，利用香农熵这一指标来量化数据的不一致性，无法直接处理数值型数据，在构建决策树的过程中通常使用递归的方法，不断地将数据切分成小数据集，直到所有目标变量完全相同，或者数据不能在切分为止，是一种贪心算法。同时在处理数据过程中容易破坏连续型变量的内在性质。

而在决策树构建过程中若果是用二元切分法则易于调整并且处理连续性特征，CART树就是使用二元切分来处理连续性变量的方法，在该过程中使用数据的方差来代替ID3决策树中的香农熵衡量数据的一致性

梯度提升树（Gradient Boosting Decision Tree）,梯度提升（gradient boosting）属于boost算法的一种，他的每一次计算都是为了减少上一次的残差（residual），与boost中对正确错误的样本进行加权不同。

Extremely randomized trees,极端随机树，简称为Extra-Trees或者ET，与随机森林的不同在于ET使用所有的训练样本得到每颗决策树，也就是每颗决策树应用的是相同的训练样本。第二点在于随机森林在一个随机子集内得到最佳分叉属性，而ET是完全随机的得到分叉值？？对于极端随机树中的某一棵决策树，他的最佳分叉属性是随机选择的，因此他的预测结果往往是不准确的，但是多可决策树组合在一起，就可以达到很好的预测结果。

PCA（Principle component analysis）

Machine learning algorithms：classification， regression， clustering， dimensionality reduction，

Precision:精确率，[tp/(tp+fp)]

Recall:召回率，[tp/(tp+fn)]

F-measure：

当β等于1的时候就是F1-Measure

Standardization（标准化）：将原始数据转化成均值为0方差为1的高斯分布

Normalization（归一化）：归一化是将单个样本缩放到具有单位范数的过程