位图索引的优缺点：

优点：对于在某一维度上的值域范围比较小的数据特别有用。因为很多操作都变成了算数运算，大大降低了空间和I/O开销，运算也会更快。

缺点：当值域范围特别大，或者是一个连续区间，那么这种方法就要构建一个很大的图索引列表甚至不适用了。

Apriori 和 FP-Growth：

两种方法都是根据已经得到的频繁模式一步一步减少挖掘任务和数据库，但总的来说，FP树效率更高，因为：

1.FP树不用产生候选集，不用每一步都对候选集进行测试。

2.只需要扫描两次数据库。

3.计算了局部频繁项集，构建了子FP树，不用进行模式搜索和匹配。

分布式频繁项集挖掘：

1. 在每一个分布式站点找到局部的频繁项集，令CF为他们的集合。
2. 得到CF中每一个项在每一个站点的局部支持度。
3. CF中每一个项的总的支持度是其在每一个站点的局部支持度的求和，那么在CF中满足sup>sup\_min的就是全局频繁项集。
4. 从全局频繁项集可以取得强关联规则。

分布式k-means：

1. 初始化K个初始中心点，并将其复制到所有的站点。
2. 在每一个站点进行一次聚类得到一个新的聚类中心。然后利用每一个类中的值的数量得到一个新的对于这一类的全局的加权聚类中心。然后又将其应用到所有站点。保持各个站点初始化的聚类中心一直保持一致。
3. 重复第2个步骤，知道聚类的结果不再变化或者变化很小。

数据挖掘的主要类型：

1. 特征化与区分 定期购买电脑的客户和不购买电脑的客户进行对比
2. 挖掘频繁模式、关联和相关性 提出更好的推荐方案
3. 分类预测 信用评分
4. 聚类 目标用户群体分类
5. 异常值检测 剔除异常数据

归一化方法：

1. 最大最小归一化：容易被很离谱的离群点影响，而且新输入数据可能越界
2. z分数归一化：要计算方差，麻烦点
3. 小数定标：同1

决策树应对过拟合的剪枝策略：

先剪枝：给用于决定决策方案的统计量设定阈值，对每一个节点处计算往下分类的相关增益，如果增益低于阈值则不再往下分类。

后剪枝：计算剪去一个点的代价复杂度，如果剪去能够实现较小的代价复杂度，则减去，否则不剪。

数据库和数据仓库：

1. 数据库是面向事务的设计，数据仓库是面向主题的设计
2. 数据库是为了查询数据而设计，数据仓库是为了分析数据而设计
3. 数据库一般存储在线交易数据（实时数据），数据仓库储存的是历史数据（反应历史变化）
4. 数据库设计是尽量避免冗余，数据仓库设计有意引入冗余