1. 为了测试机器学习的效果，通常有两个集合：训练集和测试集。
2. K-紧邻算法：采用测量特征值之间距离的方法来分类。它的工作原理是：存在一个样本数据集合，称为训练样本集，而且样本集中每个数据都存在标签。输入没有标签的新数据时，将新数据的特征与样本集数据对应的特征进行比较，然后提取样本特征值最相似的前k个分类标签，将出现次数最多的分类，作为新数据的分类。
3. K-紧邻算法的优点：精度高，对异常数据不敏感。缺点：计算的时间和空间复杂度高。
4. K近邻算法的伪代码：对于未知类型属性的每个点依次执行如下操作：1）计算已知类别数据集中的点与当前点之间的距离。2）按照距离递增排序。3）选取与当前距离点自最小的前K个点。4）确定前k个点所在类别的出现概率。5）返回前k个个点出现概率最高的类型作为当前点的预测分类。
5. 给定点(x1,x2,x3,…,xn)和（y1,y2,y3,…,yn），则他们的距离定义为Math.sqrt( (x1-y1)^2 + (x2-y2)^2 + … + (xn-yn)^2 ).在此距离公式中，值较大的通常所占的权重也大。可以通过公式 newValue = (oldValue – minValue)/(maxValue-minValue)将其归一到[0,1]区间内。
6. 监督学习：如果需要确定目标变量的类型，使用监督学习。如果目标变量是离散的，则使用分类器算法；如果是连续性数值，则使用回归算法。
7. 无监督学习：如果将数据分为离散的组，则使用聚类算法。如果还要评估数据与每个分组的相似程度，则需要使用密度估计算法。
8. 单位矩阵I：对角线为1，其余为0的矩阵。AI=IA=A. AB = BA = I,B是A的逆矩阵。
9. 分类Xi的信息定义为： l(Xi) = -log2P(Xi) P(Xi)是该分类的概率。 信息熵 H=-∑P(Xi)\* log2P(Xi).熵越大，信息越混乱。
10. 决策树算法：对数据选择使信息增益最大的属性去分割（可以分割成多份），分割后递归分割子节点，直到没有属性再去分割或者所有的实例均在相同的类。
11. P(A|B) = P(AB)/P(B) = P(B|A)\*P(A)/P(B)
12. 贝叶斯决策理论：选择概率大的那个。朴素贝叶斯常常用于文本分类。首先准备数据（数据包括文档和文档的类别），创建一个集合词典，将文档转换为一个特征向量（特征向量长度与词典的长度相同），每个值为1或者0,1表示词典指定位置的词包含在文档中，0表示不包含。这里不考虑词在单个文档中出现的次数。
13. 朴素贝叶斯算法：对于已知特征属性(F1,F2,…….Fn),决定它属于哪个类（C1,C2,……,Cm中的一个），因为贝叶斯决策理论是选取概率最大的那个，因此分类算法转换为求Max(P(Ci|F1,F2,……,Fn)) (i从1到m)。P(Ci|F1,F2,……,Fn) = P（Ci）\*P（F1,F2,……,Fn|Ci）/P(F1,F2,……,Fn)。分母对所有的i从1到m是不变的，因此只需计算分子。P(Ci)只需计算Ci类所占的次数除以所有分类的次数即可。P(F1,F2,……,Fn|Ci)计算比较复杂，我们假设F1,F2,……,Fn是完全不相关的，这个假设也是朴素贝叶斯算法被称为朴素的原因。P(F1,F2,……,Fn|Ci) = P(F1|Ci)\*P(F2|Ci)\*……\*P(Fn|Ci).计算P(Fj|Ci)只需用Ci类中包含Fj发生的次数除以Ci类总次数即可。
14. 以文档分类为例说明，首先我们有一个词典，词典中有n个词[w1,w2,…,wn]。将文档d转换为一个向量[f1,f2,…,fn].fi=1如果文档中包含wi，否则fi=0.文档有k个分类C1,C2,…,Ck.我们有一个新文档我们仅列举词出现的特征f’1,f’2,…,f’m(m<=n).则我们求新文档的分类，只需求max P(Ci|f’1,f’2,…,f’m) i从1到k。P(Ci|f’1,f’2,…,f’m)=P(Ci)\*P(f’1,f’2,…,f’m|Ci)/P(f’1,f’2,…,f’m).在求max时，分母相同，所以可以忽略。P(Ci)表示类Ci发生的概率。使用Ci发生的文档次数/所有文档的个数。P(f’1,f’2,…,f’m|Ci)表示在类Ci发生的情况下，f’1,f’2,…,f’m发生的概率。假设f’1,f’2,…,f’m完全不相关，则P(f’1,f’2,…,f’m|Ci)= P(f’1|Ci)\*P(f’2|Ci)\*……\*P(f’m|Ci)。 P(f’m|Ci)表示在类Ci发生的情况下，文档包含f’m单词的概率=f’m在Ci发生时出现文档的单词次数/ci发生时包含的所有单词数。当然这里没有考虑单词在单个文档中出现的次数。
15. 逻辑回归： sigmoid函数 sigmoid(x) = 1/(1+e^-x) sigmoid函数在点(0,0.5)中心对称，在(0,1)严格单调递增。逻辑回归在每个特征乘以一个回归系数，然后将结果相加，之后将结果带入sigmoid函数，如果大于0.5，将分为1类，小于0.5将归入0类。逻辑回归将是一个二值分类算法(只能分为两个类中的一个)。重点是求回归系数。假设我们有特征(x1,x2,……,xn),通过计算得到回归系数(w0,w1,w2,……,wn),计算x=w0\*1+w1x1+w2x2+……+wnxn，带入sigmoid即可求二值分类。
16. 利用梯度上升求最大值，梯度下降求最小值。定义移动的步长为a，函数f的梯度为△f。梯度上升回归系数w:=w+a△f。梯度下降回归系数w:=w-a△f。算法重复计算w直到计算了指定的次数或者w到达了误差允许的范围。算法如下：1）将回归系数初始化为1(回归系数的长度一般是特征的长度加1)；2）计算数据集的梯度△f，使用w:=w+a△f更新回归系数；3）重复2）n次或者直到w到达了误差允许的范围
17. 梯度上升算法每次更新回归系数都需要遍历整个数据集，当特征很多时，计算复杂度太高了。随机梯度上升算法使用一个样本点来更新回归系数，适用于样本很多时。因为可以在新样本到来时对分类器增量式更新，因此随机梯度上升算法是一个在线学习算法。。算法如下：1）将回归系数初始化为1(回归系数的长度一般是特征的长度加1)；2）对每个样本，计算数据集的梯度△f，使用w:=w+a△f更新回归系数；3）重复直到样本用完或者直到w到达了误差允许的范围
18. 梯度等于△f = y-h(x) x是真实的属性值，y是真实的分类值。 h(x)是预测值，h(x)等于w0+w1\*x1+w2\*x2+……+wnxn.梯度是计算真实类别与预测类别的差值。
19. Libsvm是最好的现成的分类器。MathNet是.net的数学库。