**K近邻**：算法采用测量不同特征值之间的距离的方法进行分类。   
优点：   
1.简单好用，容易理解，精度高，理论成熟，既可以用来做分类也可以用来做回归；   
2.可用于数值型数据和离散型数据；   
3.训练时间复杂度为O(n)；无数据输入假定；   
4.对异常值不敏感   
缺点：   
1.计算复杂性高；空间复杂性高；   
2.样本不平衡时，预测偏差比较大。如：某一类的样本比较少，而其它类样本比较多。   
3.一般数值很大的时候不用这个，计算量太大。但是单个样本又不能太少 否则容易发生误分。 KNN每一次分类都会重新进行一次全局运算。  
4.最大的缺点是无法给出数据的内在含义。

5、k值大小的选择。

**朴素贝叶斯**   
优点：   
1.生成式模型，通过计算概率来进行分类，可以用来处理多分类问题， 朴素贝叶斯对结果解释容易理解。  
2.对大数量训练和查询时具有较高的速度。即使使用超大规模的训练集，针对每个项目通常也只会有相对较少的特征数，并且对项目的训练和分类也仅仅是特征概率的数学运算而已。对小规模的数据表现很好，适合多分类任务，适合增量式训练，算法也比较简单。   
缺点：   
1.对输入数据的表达形式很敏感，   
2.由于朴素贝叶斯的“朴素”特点，所以会带来一些准确率上的损失。   
3.需要计算先验概率，分类决策存在错误率。

**决策树**   
优点：   
1.概念简单，计算复杂度不高，可解释性强，输出结果易于理解；   
2.数据的准备工作简单， 能够同时处理数据型和常规型属性，其他的技术往往要求数据属性的单一。   
3.对中间值得确实不敏感，比较适合处理有缺失属性值的样本，能够处理不相关的特征；   
4.应用范围广，可以对很多属性的数据集构造决策树，可扩展性强。决策树可以用于不熟悉的数据集合，并从中提取出一些列规则 这一点强于KNN。   
缺点：   
1.容易出现过拟合；   
2.对于那些各类别样本数量不一致的数据，在决策树当中,ID3算法信息增益的结果偏向于那些具有更多数值的特征。  
3. 信息缺失时处理起来比较困难。 忽略数据集中属性之间的相关性。

**Svm**   
优点：   
1.可用于线性/非线性分类，也可以用于回归，泛化错误率低，计算开销不大，结果容易解释；   
2.可以解决小样本情况下的**[机器学习](http://lib.csdn.net/base/2" \o "机器学习知识库" \t "http://blog.csdn.net/u013369277/article/details/_blank)**问题，可以解决高维问题 可以避免神经网络结构选择和局部极小点问题。   
3.SVM是最好的现成的分类器，现成是指不加修改可直接使用。并且能够得到较低的错误率，SVM可以对训练集之外的数据点做很好的分类决策。   
缺点：对参数调节和和函数的选择敏感，原始分类器不加修改仅适用于处理二分类问题。

**Logistic回归：**根据现有数据对分类边界线建立回归公式，依次进行分类。   
优点：实现简单，易于理解和实现；计算代价不高，速度很快，存储资源低；   
缺点：容易欠拟合，分类精度可能不高