简述机器学习和其在web安全中的应用

# 1、什么是机器学习？

想象一下我们的大脑是怎样辨别生活中的事物的，比如如何辨别一个动物是不是猪。

我们先会观察它的身形，然后会观察它的耳朵，毛发等特征，然后就会对此做出一个判断，自信地说出，这是不是只猪，或者像不像一只猪。那么我们是如何得出的判断呢？聪明的你一定有了自己的回答：“想出来的！”

“怎么想的？”

“用脑袋想的。”

“。。。”

显然，我们的大脑中应该存在一个特定的识别猪的数学模型，给大脑输入一些特征，再通过这个特定的模型就能输出结果。那么这个数学模型又是如何建立的呢？是通过我们过往大量的生活经验，即**数据**。多说一句，数学模型可以简单的理解为数学公式，比如y=f(x)，那么f就是使x映射到y的数学模型。

那么再想象一下如果在接下来的日子里，不断地有人告诉你，你之前认为是猪的其实不是猪，而是柴犬。如果你全盘接受了这些说法，那么在之后的判断中，你就会根据他人的观点而做出**调整**，你会把以前认作为猪的，认作为柴犬。这个过程我们又可以得到一个信息：我们的认知是随着生活经验的获得而不断变化的，那么用数学一点的方式说就是：我们识别猪的数学模型是通过不断的数据输入而进行调整的，是变化的，是适应数据的。

机器学习的过程和人脑学习的过程很类似。机器学习建立一个初步的数学模型，然后通过喂给模型足够多的数据，从而得到一种具备某些能力的模型。人脑学习中的生活经验，在机器学习中是数据；人脑学习中根据他人的建议做出的调整，是机器学习中的模型训练。机器学习的核心思想是：设计程序使得它可以在执行的时候提升它在某任务上的能力，而不是有着固定行为的程序。机器学习包括多种问题的定义，提供很多不同的算法，能解决不同领域的各种问题。

# 2、机器学习的最简要素

成功的机器学习有四个要素：数据、转换数据的模型、衡量模型好坏的损失函数和一个调整模型权重来最小化损失函数的算法。

* 数据。越多越好。事实上，数据是深度学习复兴的核心，因为复杂的非线性模型比其他机器学习需要更多的数据。数据的例子包括
  + 图片：例如你的手机图片，里面可能包含猫、狗、恐龙、高中同学聚会或者昨天的晚饭
  + 文本：邮件、新闻和微信聊天记录
  + 声音：有声书籍和电话记录
  + 结构数据：Jupyter notebook（里面有文本，图片和代码）、网页、租车单和电费表
* 模型。通常数据和我们最终想要的相差很远，例如我们想知道照片中的人是不是在高兴，所以我们需要把一千万像素变成一个高兴度的概率值。通常我们需要在数据上应用数个非线性函数（例如神经网络）
* 损失函数。我们需要对比模型的输出和真实值之间的误差。损失函数帮助我们决定2017年底亚马逊股票会不会价值1500美元。取决于我们想短线还是长线，这个函数可以很不一样。
* 训练。通常一个模型里面有很多参数。我们通过最小化损失函数来学这些参数。不幸的是，即使我们做得很好也不能保证在新的没见过的数据上我们可以仍然做很好。
  + 训练误差。

这是模型在评估用来训练模型的数据集上的误差。这个类似于考试前我们在模拟试卷上拿到的分数。有一定的指向性，但不一定保证真实考试分数。

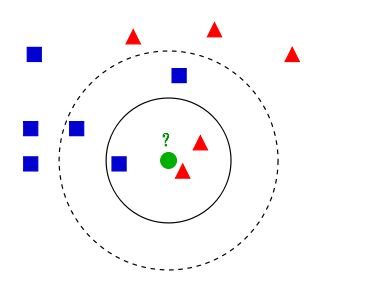
* + 测试误差。

这是模型在没见过的新数据上的误差，可能会跟训练误差不很一样（统计上叫过拟合）。这个类似于考前模考次次拿高分，但实际考起来却失误了。

# 3、常见的几种机器学习算法

1. K近邻

k近邻算法是一种基本分类和回归方法。K近邻算法，即是给定一个训练数据集，对新的输入实例，在训练数据集中找到与该实例最邻近的K个实例，这K个实例的多数属于某个类，就把该输入实例分类到这个类中。（这就类似于现实生活中少数服从多数的思想）根据这个说法，咱们来看下引自维基百科上的一幅图：



如上图所示，有两类不同的样本数据，分别用蓝色的小正方形和红色的小三角形表示，而图正中间的那个绿色的圆所标示的数据则是待分类的数据。这也就是我们的目的，来了一个新的数据点，我要得到它的类别是什么？好的，下面我们根据k近邻的思想来给绿色圆点进行分类。

* 如果K=3，绿色圆点的最邻近的3个点是2个红色小三角形和1个蓝色小正方形，少数从属于多数，基于统计的方法，判定绿色的这个待分类点属于红色的三角形一类。
* 如果K=5，绿色圆点的最邻近的5个邻居是2个红色三角形和3个蓝色的正方形，还是少数从属于多数，基于统计的方法，判定绿色的这个待分类点属于蓝色的正方形一类。

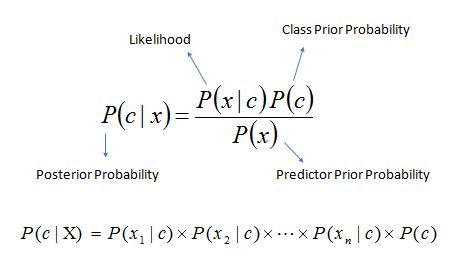
从上面例子我们可以看出，k近邻的算法思想非常的简单，也非常的容易理解，那么我们是不是就到此结束了，该算法的原理我们也已经懂了，也知道怎么给新来的点如何进行归类，只要找到离它最近的k个实例，哪个类别最多即可。

1. 朴素贝叶斯。

朴素贝叶斯分类器是一种基于贝叶斯定理的弱分类器，所有朴素贝叶斯分类器都假定样本每个特征与其他特征都不相关。举个例子，如果一种水果其具有红，圆，直径大概3英寸等特征，该水果可以被判定为是苹果。尽管这些特征相互依赖或者有些特征由其他特征决定，然而朴素贝叶斯分类器认为这些属性在判定该水果是否为苹果的概率分布上独立的。

朴素贝叶斯分类器很容易建立，特别适合用于大型数据集，众所周知，这是一种胜过许多复杂算法的高效分类方法。

贝叶斯公式提供了计算后验概率P(X|Y)的方式：



其中，

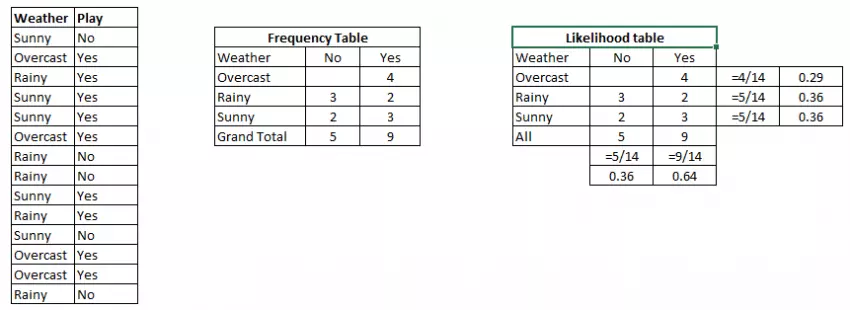
* P(c|x)是已知某样本(c，目标)，(x，属性)的概率。称后验概率。
* P(c)是该样本“c”的概率。称先验概率。
* P(x|c)是已知该样本“x”，该样本“c”的概率。
* P(x)是该样本“x”的概率。

### 朴素贝叶斯算法的分类流程

让我举一个例子。下面我设计了一个天气和响应目标变量“玩”的训练数据集（计算“玩”的可能性）。我们需要根据天气条件进行分类，判断这个人能不能出去玩，以下是步骤：

步骤1：将数据集转换成频率表；

步骤2：计算不同天气出去玩的概率，并创建似然表，如阴天的概率是0.29；



步骤3：使用贝叶斯公式计算每一类的后验概率，数据最高那栏就是预测的结果。

问题：如果是晴天，这个人就能出去玩。这个说法是不是正确的？

P(是|晴朗)=P(晴朗|是)×P(是)/P(晴朗)

在这里，P(晴朗|是)= 3/9 = 0.33，P(晴朗)= 5/14 = 0.36，P(是)= 9/14 = 0.64

现在，P(是|晴朗)=0.33×0.64/0.36=0.60，具有较高的概率。

朴素贝叶斯适合预测基于各属性的不同类的概率，因此在文本分类上有广泛应用。

## (3) 决策树

### 算法思想

决策树（decision tree）是一个树结构（可以是二叉树或非二叉树）。其每个非叶节点表示一个特征属性上的测试，每个分支代表这个特征属性在某个值域上的输出，而每个叶节点存放一个类别。

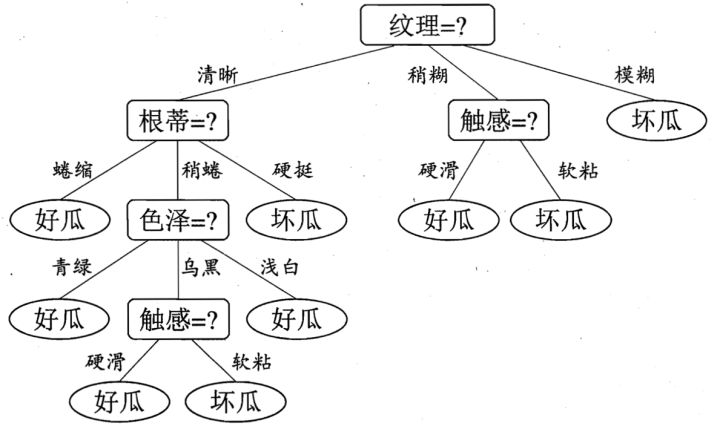
使用决策树进行决策的过程就是从根节点开始，测试待分类项中相应的特征属性，并按照其值选择输出分支，直到到达叶子节点，将叶子节点存放的类别作为决策结果。

总结来说：

决策树模型核心是下面几部分：

* 结点和有向边组成
* 结点有内部结点和叶结点俩种类型
* 内部结点表示一个特征，叶节点表示一个类

决策树表示如下：

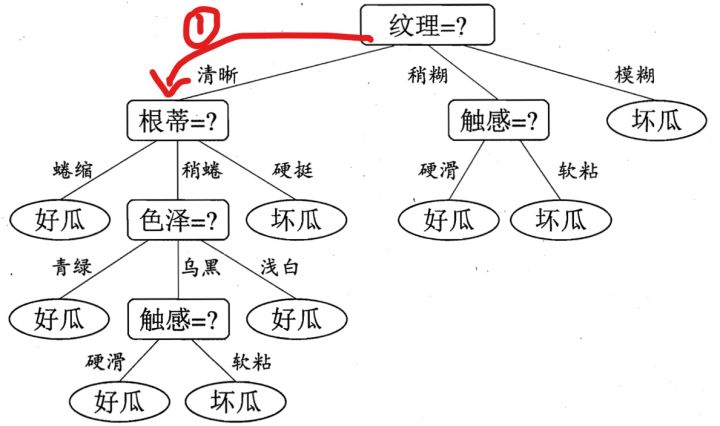


决策树代表实例属性值约束的合取的析取式。从树根到树叶的每一条路径对应一组属性测试的合取，树本身对应这些合取的析取。理解这个式子，比如上图的决策树对应表达式为：  
（（纹理=清晰）∧ (根蒂=蜷缩) ∨（（纹理=清晰）∧ (根蒂=稍蜷) ∧（色泽=乌黑）∧（触感=硬滑））∨ ..........(纹理=模糊)

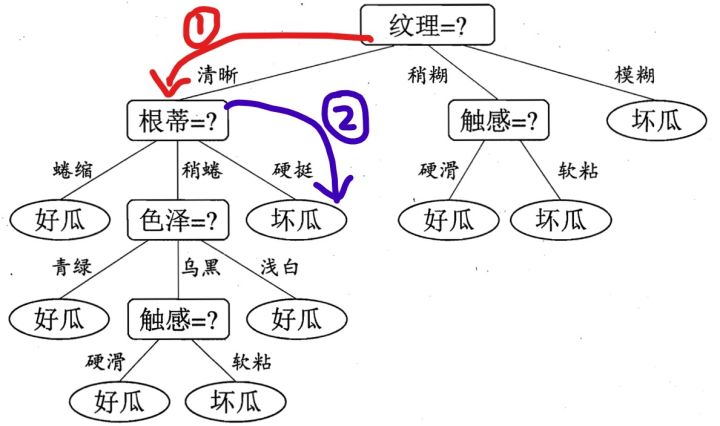
假如我现在告诉你，我买了一个西瓜，它的特点是纹理是清晰，根蒂是硬挺的瓜，你来给我判断一下是好瓜还是坏瓜，恰好，你构建了一颗决策树，告诉他，没问题，我马上告诉你是好瓜，还是坏瓜？

判断步骤如下：

根据纹理特征，已知是清晰，那么走下面这条路，红色标记：



好的，现在咋们到了第二层了，这个时候，由决策树图，我们看到，我们需要知道根蒂的特征是什么了？很好，他也告诉我了，是硬挺，于是，我们继续走，如下面蓝色所示：



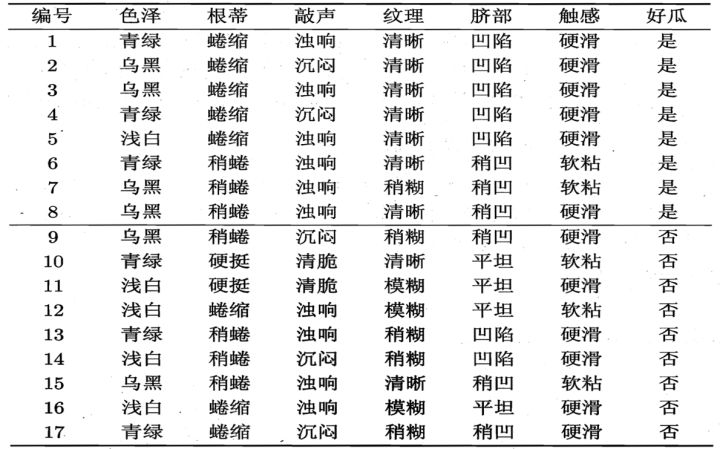
此时，我们到达叶子结点了，根据上面总结的点，可知，叶子结点代表一种类别，我们从如上决策树中，可以知道，这是一个坏瓜！于是我们可以很牛的告诉他，你买的这个纹理清晰，根蒂硬挺的瓜是坏瓜。

### 回归源头

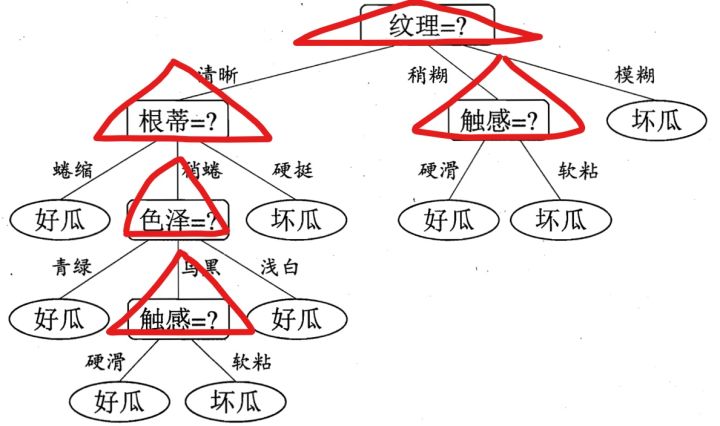
根据上面例子，非常容易直观的得到了一个实例的类别判断，只要你告诉我各个特征的具体值，决策树的判定过程就相当于树中从根结点到某一个叶子结点的遍历。每一步如何遍历是由数据各个特征的具体特征属性决定。

好的，可能有人要问了，说了这么多，给你训练数据，你的决策树是怎么构建的呢？没有树，谈何遍历，谈何分类？于是构建决策树也就成为了最重要的工作！

比如，给我下面训练数据，我如何构建出决策树

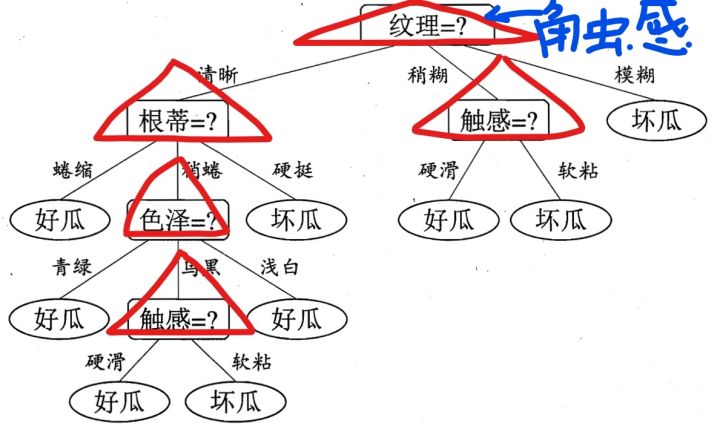


我们可以从上面决策树看出，每一次子结点的产生，是由于我在当前层数选择了不同的特征来作为我的分裂因素造成的。比如下图用红色三角形表示选择的特征：



每一层选择了指定的特征之后，我们就可以继续由该特征的不同属性值进行划分，依次一直到叶子结点。

看起来一切很顺利！但是细心的小伙伴可能会问了，为什么在第一次选择特征分裂的时候，不选择触感呢？而是选择纹理，比如如下：

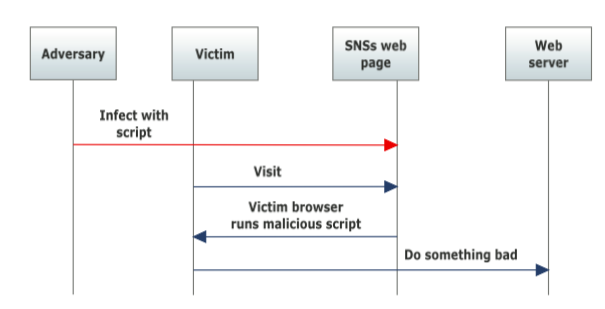


不换成触感，或者其它特征为什么也不行呢？为什么选择的是纹理，这是以什么标准来选择特征的？这就是我们要说的决策树的关键步骤是分裂属性。

所谓分裂属性就是在某个节点处按照某一特征属性的不同划分构造不同的分支，其目标是让各个分裂子集尽可能地“纯”。尽可能“纯”就是尽量让一个分裂子集中待分类项属于同一类别。

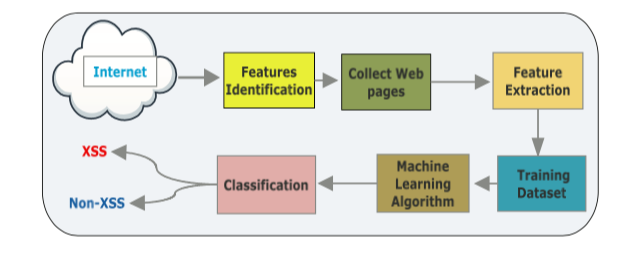
# 4、机器学习识别XSS攻击

常见的XSS攻击如下图所示：



不难看出，XSS攻击的特征主要存在于它所携带的脚本代码中。所以，我们可以把XSS攻击识别问题，转化为对脚本代码的一个二分类问题。那么如何做一个好的分类呢？如果用传统的方法，基于特定规则进行分类，得到的结果往往差强人意，这时机器学习便派上了用场。

机器学习方法识别XSS攻击被证明有更高的准确性和适用性，整个识别的过程如下图所示。



下面介绍下用机器学习算法识别XSS攻击的具体步骤。

1. 准备数据集

黑样本可以从<http://www.xssed.com/> 中爬取，白样本可以从其他正常的http请求中获得

1. 选取特征

根据Nayeem和Shailendra Rathore的论文，特征的选择可以有70种之多，这里我仅选择几个方便提取的特征：

* URL长度
* HTML中含有有害关键词的个数
* HTML中脚本代码的长度
* HTML中不可打印的字符串个数

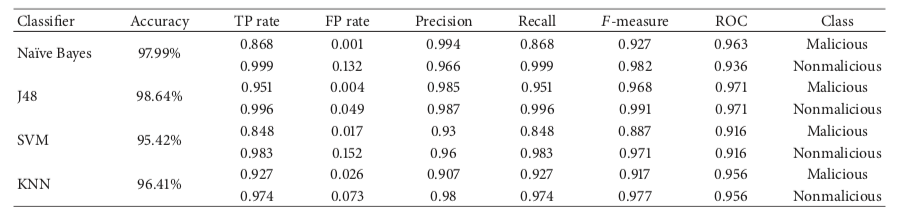
1. 选择模型

可以选择朴素贝叶斯，K近邻或者SVM等机器学习算法

1. 训练模型

使用Weka可以很容易得完成模型建立，具体请检索Weka

1. 验证模型



可以看出，应用不同算法得到的结果差异不大，总的来看机器学习对于xss攻击识别的准确率和召回率都是很好的。

# 5、小结

本文首先简述了什么是机器学习，机器学习的主要步骤和要素。之后介绍了常见的几种机器学习算法，并在文章的最后以识别XSS攻击为例，叙述了机器学习算法在web安全中应用的过程。因为时间仓促，以及本人知识储备有限，很多介绍原理性的部分也是摘自他人的知乎专栏，所以本文仅起到抛砖引玉的作用，供各位和我一样的初学者参考。

# 参考文献：

[1] <https://www.zhihu.com/question/19960417> 作者：论智

[2] <https://zhuanlan.zhihu.com/p/26703300> 作者：忆臻

[3] Shailendra Rathore, “XSSClassifier: An Efficient XSS Attack Detection Approach Based on Machine Learning Classifier on SNSs ”

[4] Nayeem Khan, “Defending Malicious Script Attacks Using Machine Learning Classifiers”

[5] Monika Rohilla, “XSS Attack: Detection and Prevention Techniques”