**大学生上网行为分析系统设计说明书**

# 引言

## 目的

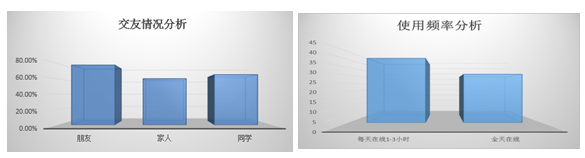
大学生通常是校园网内最活跃的网络用户,通过校园网络使用BT、P2P等软件下载网络资源,很容易造成网络堵塞和病毒传播,致使整个网络陷入瘫痪或者半瘫痪状态,严重影响校园网络教学环境正常平稳运行,故整个校园网络不仅要注意防范外部大环境网络(黑客、病毒、端口后门、木马等)对校园网设备及服务器非法的入侵,还要注意防范校园网内部用户(黑客、病毒、木马、非法入侵者等)对局域网内测试攻击。

## 背景

****（一）大学生使用社交网络数据分析****

****1、大学生社交网络软件使用分析****

根据调研问卷统计分析，几乎所有的学生都使QQ,比率达到100%， 79.8%的人使用微信， 37.6%的人使用微博，使用人人网的人数达到21.8%，另外，还有12.4%的人使用百度贴吧，5.9%的人使用陌陌。



****2、大学生社交软件交友情况分析****

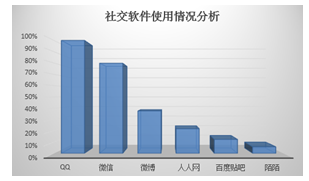
大学生的好友类型比较单一，以朋友为最多，基本局限在朋友、家人、同学之间，比例分别为78.20%、60.40%、65.70%，选择陌生人的比重较小。选择与陌生人交友的高年级的同学比例比低年级略高。

****3、大学生使用社交网络频率分析****

在调查的对象中，有40.8%的大学生表示使用社交软件的日均时间约为1-3小时。过度使用即“全天在线”的占30.7%。 令人堪忧的是，大学生在课堂上不 使 用 手 机 登陆QQ、微信、微博等的仅占到12.44%，偶尔会在课堂上使用手机登陆QQ、微信、人人等的占48.32%。

**（二）90后大学生网络行为分析**

通过调查，可以了解到，社交网络中已经渗透到大学生生活的方方面面，我们从以下两个方面进行分析。一方面，社交网络满足了当代大学生的人际交往需求，个人动态、新闻资讯、交友聊天等功能满足了大学生多方面交往与沟通需求，虚拟社交环境丰富了大学生自我展示空间，有利于大学生加强对自我的认知。



另一方面,大学生过度沉溺于社交网络，甚至在心理与生活中过度依赖导致的各方面问题也日趋明显。有些学生喜欢和网上的“陌生人”交流，反而在现实生活中不愿意与人交流，过分自闭。生活中过度依赖网络，不上网总感觉失去了什么，患得患失。很多大学生自控能力差，上课经常登录社交网站，影响了学习，成为上课的“低头族”，还有的学生痴迷网络社交平台，熬夜在平台上与朋友互动，成为晚上的“夜猫子”，等等等等，这些行为严重影响了正常的学习和生活。对于大学生而言，再享受社交网络平台提供的极速体验和快捷便利的同时也存在较多不良因素。

**（三）大学生使用社交网络易引发的不良问题**

****1.耗时耗力，影响学业****

通过调研数据我们了解到，大学生登录社交网络平台因时间和地点是不受外界控制的，因此只要条件允许，82.5%的大学生表示会在学习过程中打开社交软件或者浏览好友动态， 73.9%的大一学生都选择在上课期间会习惯性登录社交软件。学生在上课期间登录社交软件，不能集中精力听课，无法高质量的完成老师布置的任务是大学生使用网络平台的主要困扰和问题。过度依赖社交网络，耗时耗力，影响学业。

****2.判断力低，易受影响****

大学生在使用社交网络时，对于社交平台上的“灰色信息”以及鱼龙混杂的网络恶搞、网络谩骂、网络谣言等消息判断力低，社会阅历少，思维单纯，在平台使用过程中缺乏甄别能力，易感情用事受其影响。调查显示，社交平台上的信息言论会影响大多数学生的看法，其中认为此类信息可以拓展视野提供新与阅历的占到51.7%，觉得说的有一定意义可以参考的占39.6%，认为信息失真或对此表示不接受的占20.1%。

****3.信息泄露，存在安全隐患****

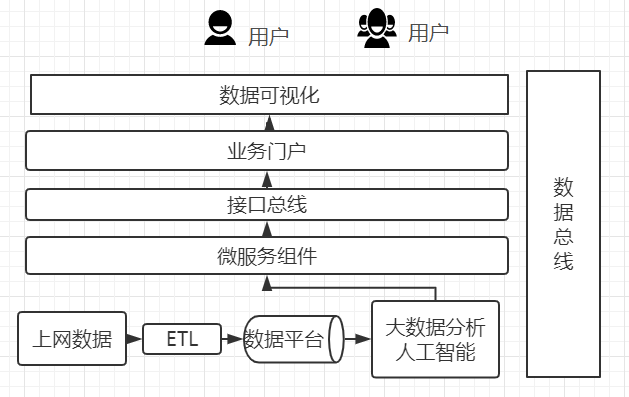
大部分社交网站要求用户在注册时使用真实姓名并提供身份证信息、手机号码、所在地市、甚至个人真实照片等。而且多数社交网络的APP手机端应用软件在安装时要求读取个人信息等相应权限，大学生自我保护不够，为了能正常使用，一般不会阻止此类行为。使用社交网络平台的大学生个人信息会被这些平台软件获悉并收集，并根据学生经常搜集的信息情况进行归类并可能进行恶意侵扰，安全隐患非常大。 甚至有些社交软件如“微信”等在发表个人状态时的“定位”功能，极易暴露大学生的个人行踪，引发的造成大学生身体、心灵受到伤害的新闻也屡见不鲜。

# 总体设计

1. 需求概述

系统主要包含一下功能：

1. 对学生上网流量进行分析；
2. 对学生每天上网时长进行分析；
3. 对学生应用进行分析；
4. 对学生关注新闻进行分析；
5. 对学生检索文字进行分析；
6. 业务架构



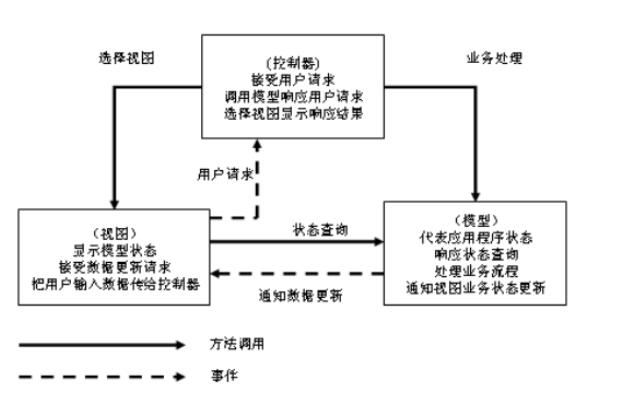
业务架构图

如上图所示,对大学生上网数据进行ＥＴＬ（extract提取、transform转换、load加载）后，将整理好的数据加载到数据平台进行大数据和人工智能的分析。

通过微服务接口将业务数据进行可视化展示。

1. 系统设计模式

系统采用目前最为流行的MVC设计模式，即Model-View-Controller，即把一个应用的输入、处理、输出流程按照Model、View、Controller的方式进行分离，这样一个应用被分成三层--模型层、视图层、控制层。



视图(View)：  
　　视图是用户看到并与之交互的界面。对老式的Web应用程序来说，视图就是由HTML元素组成的界面，在新式的Web应用程序中，HTML依旧在视图中扮演着重要的角色，但一些新的技术已层出不穷，它们包括Adobe Flash和象XHTML，XML/XSL，WML等一些标识语言和Web services。如何处理应用程序的界面变得越来越有挑战性。MVC一个大的好处是它能为你的应用程序处理很多不同的视图。在视图中其实没有真正的处理发生，不管这些数据是联机存储的还是一个雇员列表，作为视图来讲，它只是作为一种输出数据并允许用户操纵的方式。

模型 (Model)：  
模型表示企业数据和业务规则。在MVC的三个部件中，模型拥有最多的处理任务。例如它可能用象EJBs和ColdFusion Components这样的构件对象来处理数据库。被模型返回的数据是中立的，就是说模型与数据格式无关，这样一个模型能为多个视图提供数据。由于应用于模型的代码只需写一次就可以被多个视图重用，所以减少了代码的重复性。

控制器(Controller)：  
　　控制器接受用户的输入并调用模型和视图去完成用户的需求。所以当单击Web页面中的超链接和发送HTML表单时，控制器本身不输出任何东西和做任何处理。它只是接收请求并决定调用哪个模型构件去处理请求，然后确定用哪个视图来显示模型处理返回的数据。

MVC处理过程：

    首先控制器接收用户的请求，并决定应该调用哪个模型进行处理；

    然后模型根据用户请求进行相应的业务逻辑处理，并返回数据；

    最后控制器调用相应的视图来格式化模型返回的数据，并通过视图呈现操作结果。

MVC的具体实现：

* 用SpringMVC类封装与用户互动的数据元素(Model)；
* 用Action类实现动作处理，链接转向。实现MVC中的控制器功能(Console)；
* 借助标准的JSP JSTL标签和EXTJS2.0+AJAX来完成数据呈现(视图)，实现MVC中的展现视图功能(View)；
* 说明：Spring MVC(ver2.5.6)已完全解耦前端界面与后台JAVA应用，同时实现前端界面中的域无缝地对接和后台控制器Control层中的对象属性自动映射匹配，极大地方便了开发和维护；

1. 数据库设计

* [数据库](http://lib.csdn.net/base/mysql" \o "MySQL知识库)的设计范式是数据库设计所需要满足的规范，满足这些规范的数据库是简洁的、结构明晰的，同时，不会发生插入（insert）、删除（delete）和更新（update）操作异常。反之则是乱七八糟，不仅给数据库的编程人员制造麻烦，而且面目可憎，可能存储了大量不需要的冗余信息。数据库的设计按照数据库三范式来设计。

**第一范式（1NF）无重复的列**

所谓第一范式（1NF）是指数据库表的每一列都是不可分割的基本数据项，同一列中不能有多个值，即实体中的某个属性不能有多个值或者不能有重复的属性。如果出现重复的属性，就可能需要定义一个新的实体，新的实体由重复的属性构成，新实体与原实体之间为一对多关系。在第一范式（1NF）中表的每一行只包含一个实例的信息。简而言之，第一范式就是无重复的列。

**第二范式（2NF）属性完全依赖于主键 [ 消除部分子函数依赖 ]**

如果关系模式R为第一范式，并且R中每一个非主属性完全函数依赖于R的某个候选键， 则称为第二范式模式。

第二范式（2NF）是在第一范式（1NF）的基础上建立起来的，即满足第二范式（2NF）必须先满足第一范式（1NF）。第二范式（2NF）要求数据库表中的每个实例或行必须可以被惟一地区分。为实现区分通常需要为表加上一个列，以存储各个实例的惟一标识。这个惟一属性列被称为主关键字或主键、主码。

**第三范式（3NF）属性不依赖于其它非主属性 [ 消除传递依赖 ]**

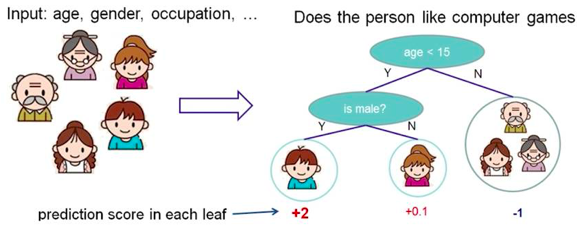
如果关系模式R是第二范式，且每个非主属性都不传递依赖于R的候选键，则称R为第三范式模式。满足第三范式（3NF）必须先满足第二范式（2NF）。第三范式（3NF）要求一个数据库表中不包含已在其它表中已包含的非主关键字信息。

# 程序描述

**（一）算法**

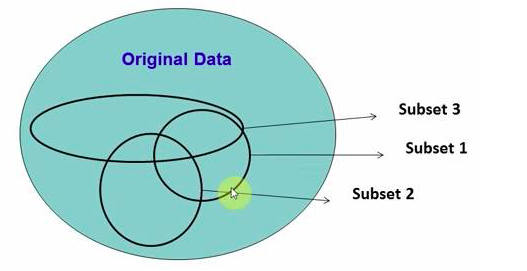
## **1. 决策树**

根据一些 feature 进行分类，每个节点提一个问题，通过判断，将数据分为两类，再继续提问。这些问题是根据已有数据学习出来的，再投入新数据的时候，就可以根据这棵树上的问题，将数据划分到合适的叶子上。

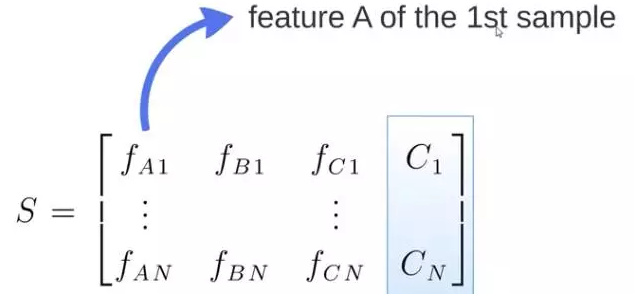


## **2. 随机森林**

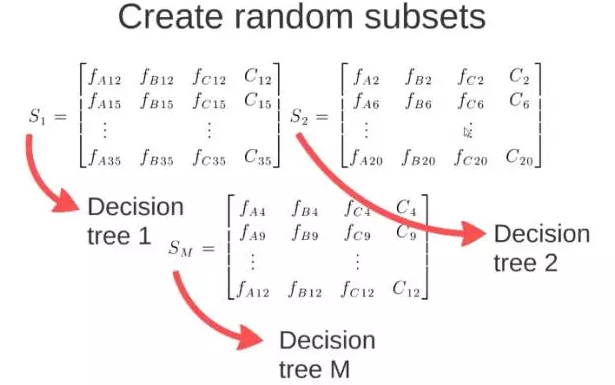
在源数据中随机选取数据，组成几个子集



S 矩阵是源数据，有 1-N 条数据，A B C 是feature，最后一列C是类别

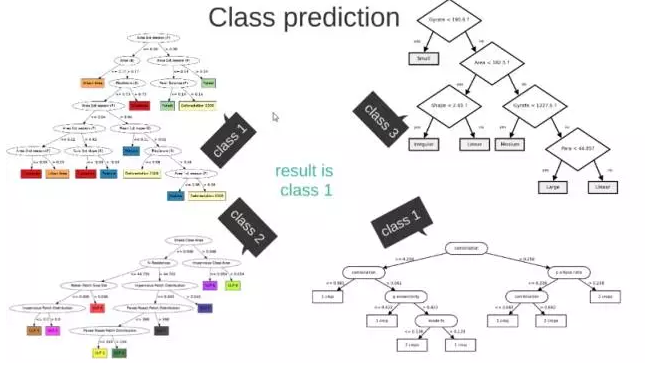


由 S 随机生成 M 个子矩阵



这 M 个子集得到 M 个决策树

将新数据投入到这 M 个树中，得到 M 个分类结果，计数看预测成哪一类的数目最多，就将此类别作为最后的预测结果



## **3. 逻辑回归**

当预测目标是概率这样的，值域需要满足大于等于0，小于等于1的，这个时候单纯的线性模型是做不到的，因为在定义域不在某个范围之内时，值域也超出了规定区间。

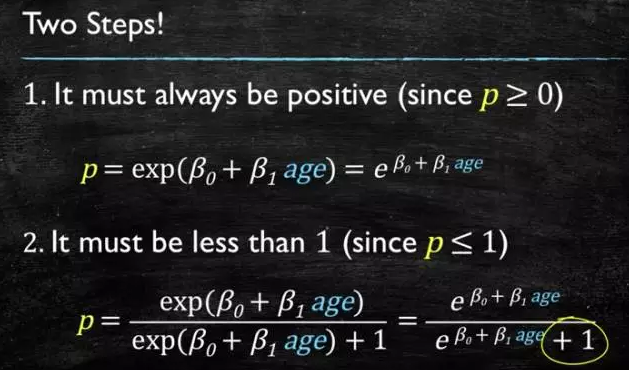


所以此时需要这样的形状的模型会比较好

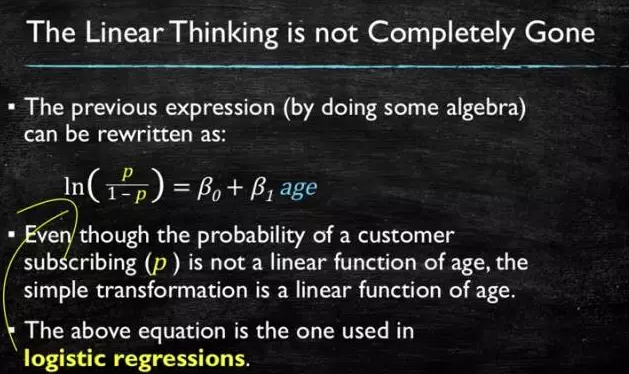


那么怎么得到这样的模型呢？

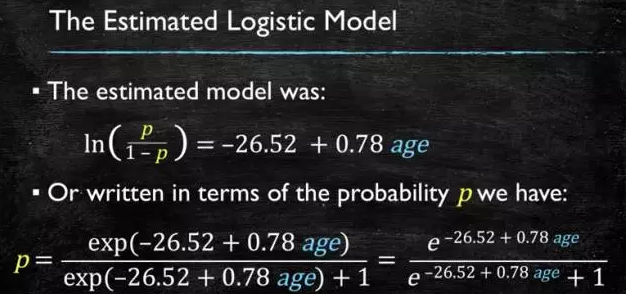
这个模型需要满足两个条件 大于等于0，小于等于1  
大于等于0 的模型可以选择 绝对值，平方值，这里用 指数函数，一定大于0  
小于等于1 用除法，分子是自己，分母是自身加上1，那一定是小于1的了



再做一下变形，就得到了 logistic regression 模型



通过源数据计算可以得到相应的系数了



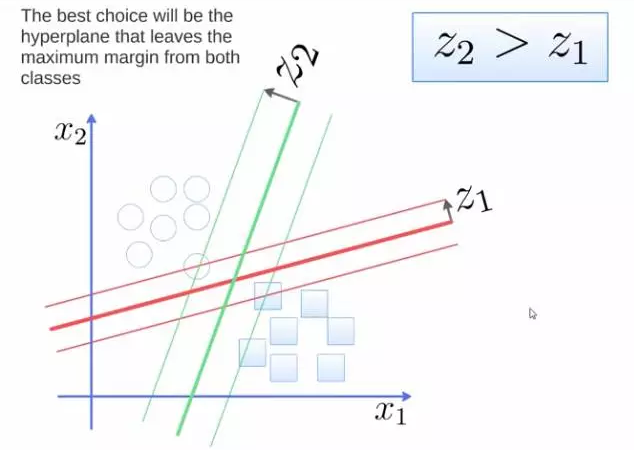
最后得到 logistic 的图形



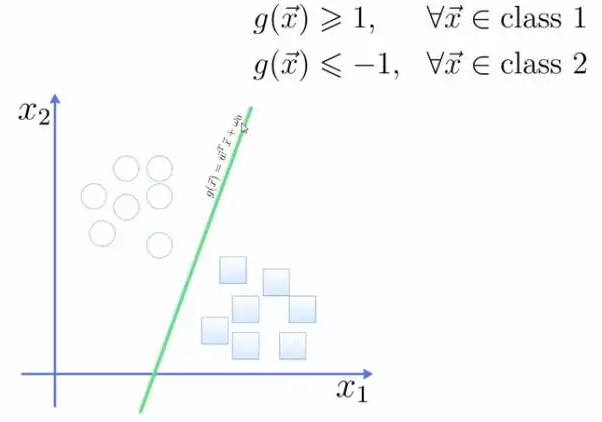
## **4. SVM**

support vector machine

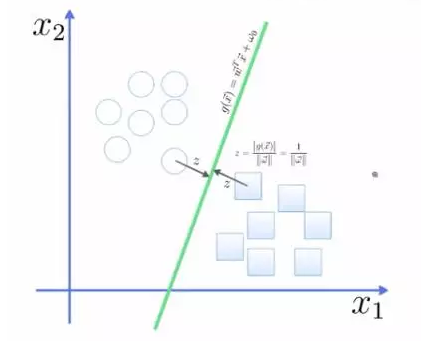
要将两类分开，想要得到一个超平面，最优的超平面是到两类的 margin 达到最大，margin就是超平面与离它最近一点的距离，如下图，Z2>Z1，所以绿色的超平面比较好



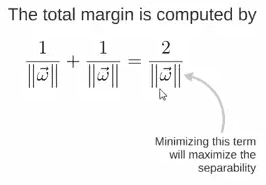
将这个超平面表示成一个线性方程，在线上方的一类，都大于等于1，另一类小于等于－1



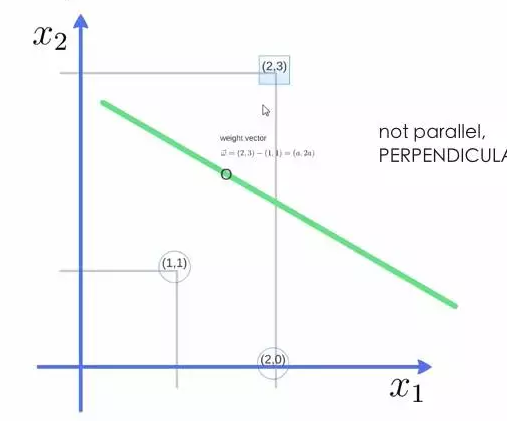
点到面的距离根据图中的公式计算



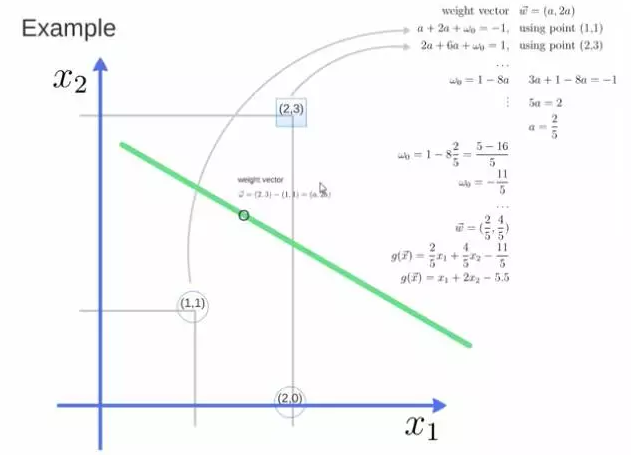
所以得到 total margin 的表达式如下，目标是最大化这个 margin，就需要最小化分母，于是变成了一个优化问题



举个栗子，三个点，找到最优的超平面，定义了 weight vector＝（2，3）－（1，1）



得到 weight vector 为（a，2a），将两个点代入方程，代入（2，3）另其值＝1，代入（1，1）另其值＝-1，求解出 a 和 截矩 w0 的值，进而得到超平面的表达式。



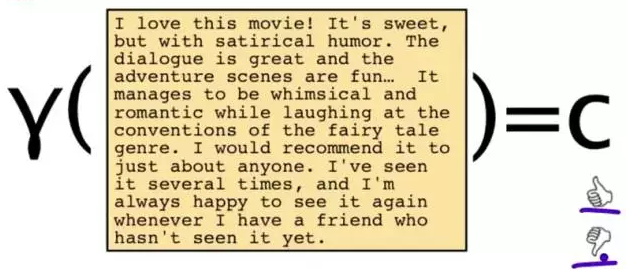
a 求出来后，代入（a，2a）得到的就是 support vector

a 和 w0 代入超平面的方程就是 support vector machine

## **5. 朴素贝叶斯**

举个在 NLP 的应用

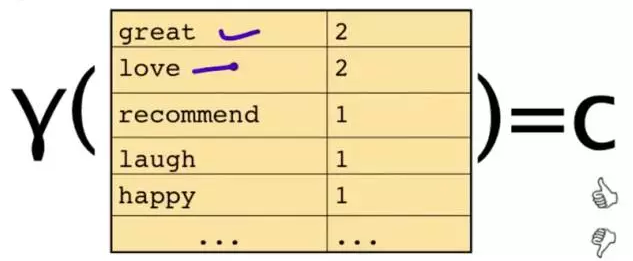
给一段文字，返回情感分类，这段文字的态度是positive，还是negative



为了解决这个问题，可以只看其中的一些单词

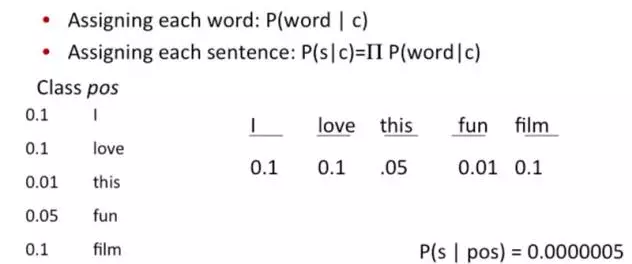


这段文字，将仅由一些单词和它们的计数代表



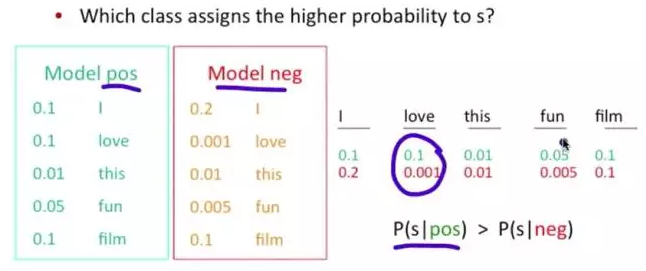
原始问题是：给你一句话，它属于哪一类

通过 bayes rules 变成一个比较简单容易求得的问题



问题变成，这一类中这句话出现的概率是多少，当然，别忘了公式里的另外两个概率

栗子：单词 love 在 positive 的情况下出现的概率是 0.1，在 negative 的情况下出现的概率是 0.001

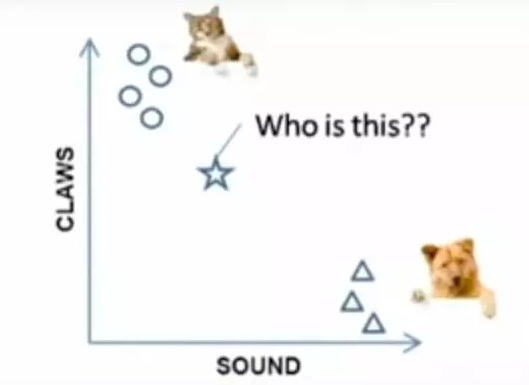


## **6. K最近邻**

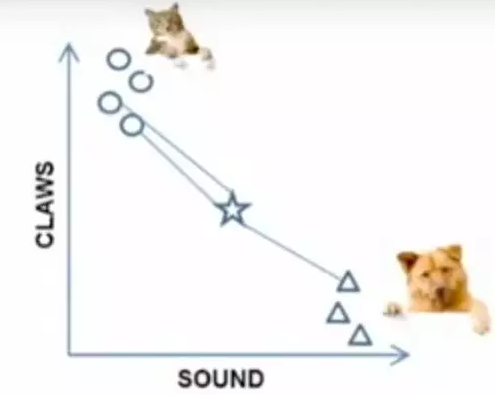
k nearest neighbours

给一个新的数据时，离它最近的 k 个点中，哪个类别多，这个数据就属于哪一类

栗子：要区分 猫 和 狗，通过 claws 和 sound 两个feature来判断的话，圆形和三角形是已知分类的了，那么这个 star 代表的是哪一类呢

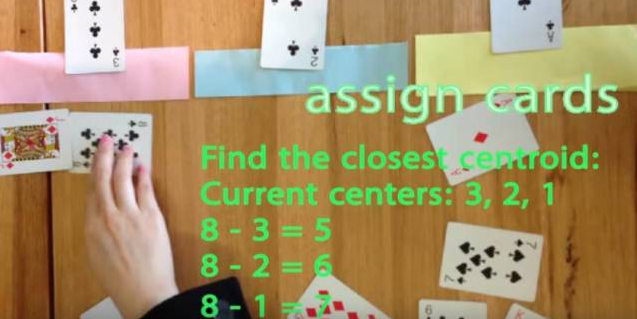


k＝3时，这三条线链接的点就是最近的三个点，那么圆形多一些，所以这个star就是属于猫



## **7. K均值**

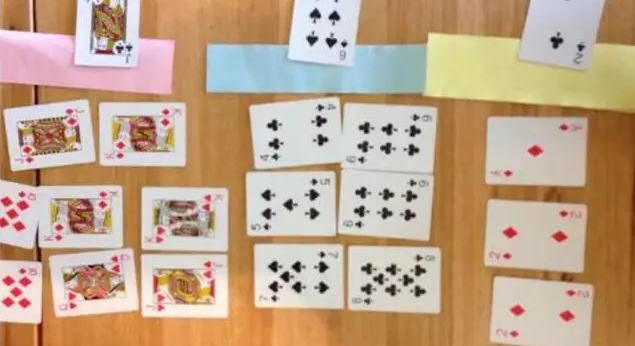
想要将一组数据，分为三类，粉色数值大，黄色数值小  
最开心先初始化，这里面选了最简单的 3，2，1 作为各类的初始值  
剩下的数据里，每个都与三个初始值计算距离，然后归类到离它最近的初始值所在类别

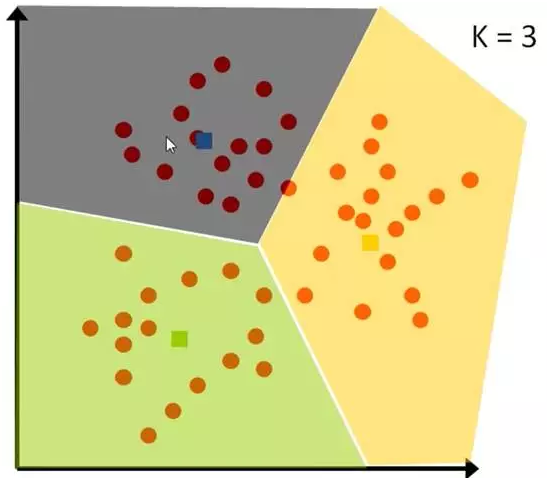


分好类后，计算每一类的平均值，作为新一轮的中心点



几轮之后，分组不再变化了，就可以停止了



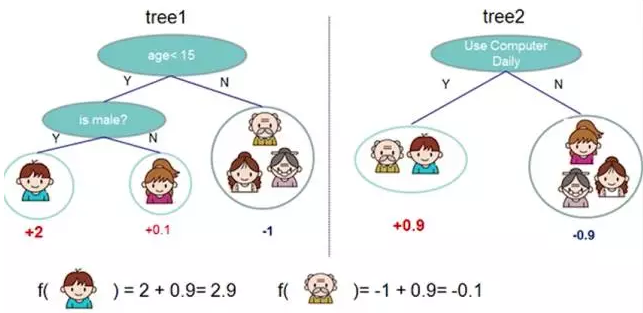


## **8. Adaboost**

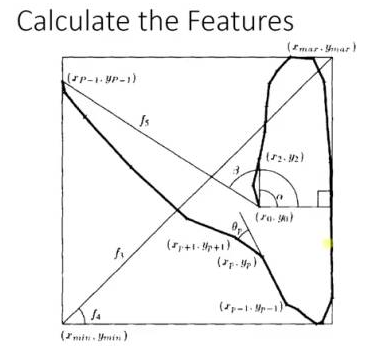
adaboost 是 bosting 的方法之一

bosting就是把若干个分类效果并不好的分类器综合起来考虑，会得到一个效果比较好的分类器。

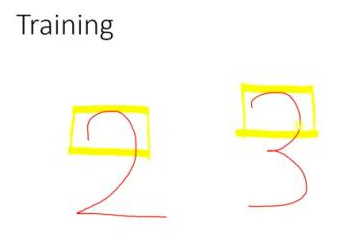
下图，左右两个决策树，单个看是效果不怎么好的，但是把同样的数据投入进去，把两个结果加起来考虑，就会增加可信度



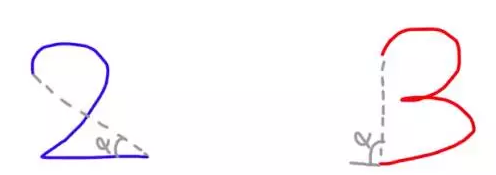
adaboost 的栗子，手写识别中，在画板上可以抓取到很多 features，例如 始点的方向，始点和终点的距离等等



training 的时候，会得到每个 feature 的 weight，例如 2 和 3 的开头部分很像，这个 feature 对分类起到的作用很小，它的权重也就会较小



而这个 alpha 角 就具有很强的识别性，这个 feature 的权重就会较大，最后的预测结果是综合考虑这些 feature 的结果

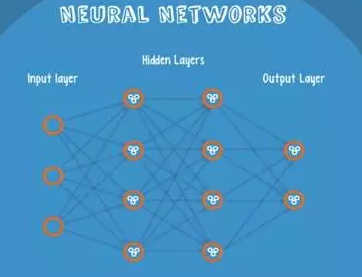


## **9. 神经网络**

Neural Networks 适合一个input可能落入至少两个类别里

NN 由若干层神经元，和它们之间的联系组成  
第一层是 input 层，最后一层是 output 层

在 hidden 层 和 output 层都有自己的 classifier



input 输入到网络中，被激活，计算的分数被传递到下一层，激活后面的神经层，最后output 层的节点上的分数代表属于各类的分数，下图例子得到分类结果为 class 1

同样的 input 被传输到不同的节点上，之所以会得到不同的结果是因为各自节点有不同的weights 和 bias

这也就是 forward propagation



1. **接口设计**
2. 通用请求参数

每个请求都要携带的参数，用于描述每个请求的基本信息，后端可以通过这些字段进行接口统计，或APP终端设备的统计，一般放到header或url参数中。

|  |  |
| --- | --- |
| 字段名称 | 说明 |
| version | 客户端版本version，例：1.0.0 |
| token | 登陆成功后，server返回的登陆令牌token |
| os | 手机系统版本（Build.VERSION.RELEAS）例：3.4，3.5 |
| from | 请求来源，例：android/ios/h5 |
| channel | 渠道信息，例：com.wandoujia |
| appid | APP唯一标识，有的公司一套server服务多款APP时，需要区分开每个APP来源 |
| net | APP当前网络状态 |

2.请求Path

原则：在以下命名规范的基础上尽量保持良好的可读性，见名知意。另外这里需要额外提下restful规范，个人理解restful规范是通过path表示当前请求的资源，通过method表示当前请求的操作动作（post=增，delete=删，put=改，get=查），例：GET /userinfo/{id}，通过这个path就可以清楚的知道当前请求的意图是根据id获取用户信息，而APP开发中很多时候一个页面是需要同时获取，如，用户，订单，营销各种信息，这时候就很难用一个path来表示当前请求的真正意图，restful规范就很难得到实现，故本文介绍的接口设计方法，只区分get和post，通过path命名定义请求行为，

操作行为 Method Path

查找 GET getXxx

增加 POST addXxx/submitXxx

修改 POST modifyXxx

删除 POST delXxx

示例：

操作行为 Method Path

获取用户信息 GET getUserInfo

修改密码 POST modifyPwd

登陆 GET login

3.响应数据

字段名称 说明

code 响应状态码，200：成功；非200：失败

msg 请求失败时的message

time 服务端时间戳，单位：毫秒。用于同步时间

data 数据实体

code=200时，msg=登陆成功/修改成功/提交成功；如果需要Toast，可以直接使用msg。

code!=200时，msg=错误提示信息；比如login接口，”账号或密码错误”，”账号不存在”类似这些的业务提示文案放在msg字段，客户端直接Toast就可以了。不过需要提醒后端同学，错误提示不能自己觉的什么合适就提示什么，要按需求文档来提供，或和PM确认。

4.出错处理设计

**2xx-确定**  
客户端请求已成功。  
201-已创建。  
202-已接受  
203-非权威性信息。  
204-无内容。  
205-重置内容。  
206-部分内容。  
**3xx-重定向**  
客户端浏览器必须采取更多操作来实现请求。 例如，浏览器可能不得不请求服务器上的不同的页面，或通过代理服务器重复该请求。 301-对象已永久移走，即永久重定向。  
302-对象已临时移动。  
304-未修改。  
307-临时重定向。  
**4xx-客户端错误**  
发生错误，客户端似乎有问题。  
例如，客户端请求不存在的页面，客户端未提供有效的身份验证信息。 400-错误的请求。  
401-访问被拒绝。IIS定义了许多不同的401错误，它们指明更为具体的错误原因。这些具体的错误代码在浏览器中显示，但不在IIS日志中显示：  
这些状态代码表示请求可能出错，妨碍了服务器的处理。  
401.1-登录失败。  
401.2-服务器配置导致登录失败。  
401.3-由于ACL对资源的限制而未获得授权。  
401.4-筛选器授权失败。  
400 （错误请求） 服务器不理解请求的语法。  
404 （未找到） 服务器找不到请求的网页。  
405 （方法禁用） 禁用请求中指定的方法。  
406 （不接受） 无法使用请求的内容特性响应请求的网页。  
407 （需要代理授权） 此状态代码与 401（未授权）类似，但指定请求者应当授权使用代理。  
408 （请求超时） 服务器等候请求时发生超时。  
409（冲突） 服务器在完成请求时发生冲突。服务器必须在响应中包含有关冲突的信息。  
410 （已删除） 如果请求的资源已永久删除，服务器就会返回此响应。  
411 （需要有效长度） 服务器不接受不含有效内容长度标头字段的请求。  
412（未满足前提条件） 服务器未满足请求者在请求中设置的其中一个前提条件。  
413 （请求实体过大） 服务器无法处理请求，因为请求实体过大，超出服务器的处理能力。  
414 （请求的 URI 过长） 请求的 URI（通常为网址）过长，服务器无法处理。  
415 （不支持的媒体类型） 请求的格式不受请求页面的支持。  
416 （请求范围不符合要求） 如果页面无法提供请求的范围，则服务器会返回此状态代码。  
417 （未满足期望值） 服务器未满足”期望”请求标头字段的要求。  
**5xx（服务器错误）**  
这些状态代码表示服务器在尝试处理请求时发生内部错误。 这些错误可能是服务器本身的错误，而不是请求出错。  
500 （服务器内部错误） 服务器遇到错误，无法完成请求。  
501 （尚未实施） 服务器不具备完成请求的功能。例如，服务器无法识别请求方法时可能会返回此代码。  
502 （错误网关） 服务器作为网关或代理，从上游服务器收到无效响应。  
503 （服务不可用） 服务器目前无法使用（由于超载或停机维护）。通常，这只是暂时状态。  
504 （网关超时） 服务器作为网关或代理，但是没有及时从上游服务器收到请求。 505 （HTTP 版本不受支持） 服务器不支持请求中所用的 HTTP  
协议版本。  
**Http展示状态码结果**  
\*\*200 - 服务器成功返回网页，客户端请求已成功。  
302 - 对象临时移动。服务器目前从不同位置的网页响应请求，但请求者应继续使用原有位置来进行以后的请求。  
304 -  
属于重定向。自上次请求后，请求的网页未修改过。服务器返回此响应时，不会返回网页内容。  
401 - 未授权。请求要求身份验证。  
对于需要登录的网页，服务器可能返回此响应。  
404 - 未找到。服务器找不到请求的网页。  
2xx - 成功。表示服务器成功地接受了客户端请求。  
3xx - 重定向。表示要完成请求，需要进一步操作。客户端浏览器必须采取更多操作来实现请求。例如，浏览器可能不得不请求服务器上的不同的页面，或通过代理服务器重复该请求。  
4xx - 请求错误。这些状态代码表示请求可能出错，妨碍了服务器的处理。  
5xx - 服务器错误。表示服务器在尝试处理请求时发生内部错误。 这些错误可能是服务器本身的错误，而不是请求出错。