# 引言

从2001年世界上第一部智能手机诞生开始，智能手机的发展速度可谓是日新月异。从以诺基亚为代表的塞班一家独大，到Android，ios，WindowsPhone三足鼎立，到如今Android和ios几乎已经占领了智能手机操作系统95%以上的份额，而其中又以Android份额最多。伴随着智能手机数量数量的爆发式增长，手机的硬件也得到了极大的升级，功能也变得更加丰富。手机已经不仅仅是用来打电话，发信息，而是集通信，娱乐，办公，社交等诸多功能于一体的智能终端设备。随着手机内置的传感器越来越多，手机所能收集到的信息也越来越多，而且因为多数人都有随身携带的习惯，让手机俨然成为一个天然的“人体数据收集站”。而且，近些年机器学习热度越来越高，“情绪分析”作为其中的一个重要方向也吸引了大量研究人员的兴趣。鉴于智能手机的优势和特性，通过分析智能手机的数据来进行情绪分析，正逐渐成为机器学习的一个研究热点。

## 1.1 研究背景

1.1.1 情绪的认识

情绪是人在探索和发现周围环境时，对外界环境刺激做出的心理或生理反应，是一种综合了人的思想，感觉，和行为的状态。常见的情绪有高兴，平静，悲伤，生气等。积极的情绪可以让人精神焕发，干劲十足，消极的情绪可以让人萎靡不振，日渐消沉，人的情绪极大影响了人们的日常生活和工作的效率。因此在这种情况下，对情绪的了解显得极为重要。

人的情绪变化时通常伴随着人体的一系列生理表现，或者行为表现，比如心跳加快，血压升高，脑电波活跃，激素分泌，又比如欢呼雀跃，手舞足蹈，唉声叹气，愁眉苦脸等等。正是因为情绪变化时，人体会产生这些变化，那么，如果我们可以通过愈来愈发达的技术手段准确的获取人体的这些信息，并且准确有效，再加上科学正确的分析，我们就可以判断出用户此刻的心情。情绪分析和识别是一项多领域交叉的科研问题，目前，多个领域都在对于情绪进行相关的研究，涉及的领域有医学，心理学，神经科学，人工智能等。

智能手机在发展过程中，为了丰富功能，增强用户体验，内置了越来越丰富的传感器，比如光线传感器，温度传感器，磁力传感器，重力传感器，陀螺仪，及速度传感器等等，这为我们进行实验提供给了非常有利的条件。

之所以选用安卓手机来收集这些信息，是因为，首先，越来越多的人拥有智能手机,其中绝大部分是安卓手机（下文“手机”，“智能手机”都指代“安卓手机”），安卓手机的大规模使用保障了数据的充足，在之前是不存在这样优越的研究条件的。其次，随着手机越来越智能，功能越来越多，浏览网页，聊天阅读，拍照娱乐，移动支付等等，使用手机的场景越来越多，人们越来越离不开手机，这使得很多人有了随身携带手机的习惯。并且，手机的携带对被测试者的正常生活影响较小，这使得持续获取数据成为了可能。

机器学习经过多年的发展，相关理论已经非常成熟。随着科技水平的飞速发展，并且伴随着数据量的暴增，人们进入“大数据”时代，在这样的时代背景下，机器学习具有传统统计难以匹及的优势，因此本文选用了机器学习的方式来进行情绪的识别。

## 1.2 研究现状

目前有关人的情绪状态分析识别的研究主要有两类方向，一类是基于生理信号进行情绪识别，一类是基于非生理信号进行识别。

非生理信号的情绪识别研究主要包括基于面部表情的情绪识别，基于语音语调的情绪识别两种。

基于面部表情的情绪识别：不同的表情常常会反映不同的情绪，所以我们可以通过分析人的面部表情以及伴随的面部肌肉动作来判断人的情绪。比如嘴角上扬时，并且眼角出现皱纹，我们通常可以判断此时的情绪状态为高兴；眼睛瞪大，眉头紧皱，我们通常可以判断此时情绪状态为愤怒。基于面部表情的情绪识别既可以是基于局部特征进行的情绪分析，也可以是基于整体特征的情绪识别。前者是考虑到人在不同情绪下，人脸五官的形状，大小，以及相对位置会有差异，并以此作为情绪分析的数据特征，进行情绪分析。后者则是从整体出发，考虑不同情绪下整体面部特征的区别，提取特征的范围是整个人脸。目前基于面部表情的情绪识别已经取得一定进展，比如Facebook最近的一项专利技术就是有关面部表情的情绪识别，通过摄像头捕捉用户表情，根据捕捉到的表情推测用户当前情绪，并能够借此向用户推荐最适合当前心情的emoji表情。

基于语音语调的情绪识别：不同的语言表达方式通常也意味着不同的情绪状态，比如，人的心情愉悦时，语音会轻快，语调会上扬，心情失落时，语调会变得低沉，语调会放缓。基于语音的情绪识别研究一般多关注于基音的频率、声音携带的能量、语速快慢、表达是否流利等特征。目前，基于语音的情绪识别技术已经实现商用，如2012年成立的Beyond Verbal就是一家语音情绪识别领域领先的公司，该公司实现了通过算法识别音域变化，从而对用户的情绪进行判断，可以识别出400个复杂情绪，甚至能够识别其中的微小差别。

基于非生理信号的情绪识别研究简单易行，操作难度小，但是存在识别不准确和不可靠的问题，毕竟人是种复杂的动物，常常会为了隐藏自己的情绪而伪装自己的声音表情。另一方面，对于一些病患人群可能无法提供非生理信号情绪识别所需的特征，所以对于这一人群也不适用。

另一大类是基于生理信号的情绪识别。这类研究关注的重点通常是人体的心率，血压，脑电波等生理信号。人处在不同的情绪状态时，表现出来的生理信号也是不同的。我们们通过收集人在喜、怒、哀、乐等不同情绪下的各项生理指标，比如通过测量人体心脏跳动的频率、呼吸的频率、脑内电波的活跃程度、皮肤外表的阻抗等数据，然后与对应的情绪状态高兴还是悲伤建立对应联系，然后分析数据间的内在关系，从而建立数据模型实现对人情绪的判断。生理信息相比非生理信号而言，不易被伪装，并且通过这种方式，在获得准确信息的前提下，分析结果准确率高。但是该方法也存在着存在收集困难，如收集难度大，信号准确率无法保障，易被干扰的缺点。

情绪识别这项技术的应用前景十分广泛，比如听音乐时，如果能够识别出用户此刻的心情，进而推送给用户符合当前心情状态的歌曲，那么用户的使用体验就会上升一个台阶。再比如，在医疗护理过程中，如果可以准确分辨出患者此刻的心情，尤其是具有表达沟通障碍的患者，进而对其提供针对性的护理，无疑对于治疗过程是有极大帮助的。再比如，如果可以准确获知用户使用一款产品时的情绪状态，那么开发人员就可以针对用户使用时的情绪进行更有针对性的优化处理，从而持续提供更好的服务。

## 1.3 本文的研究内容

下文主要研究基于机器学习的安卓移动用户情绪分析系统的设计与实现，即将从安卓设备采集来的数据，通过经典的机器学习算法进行分析，从而预测出使用者现在的心情状态。

下面对后边章节将要讨论的内容进行简要介绍。

第二章主要介绍了用到的技术和工具软件。

第三章介绍了情绪分类模型的构建原理，主要讲了六种机器学习的分类模型。

第四章详细介绍了实验过程中用户数据收集以及对收集来的原始数据的预处理操作。

第五章详细介绍了在构建情绪分类模型过程前进行的特征工程，包括了特征提取、相关性分析、特征选择、主成分分析等。

第六章介绍了情绪分析模型的构建，主要包括K近邻分类模型、SVC分类模型、朴素贝叶斯分类模型三种情绪分类模型。

第七章对实验结果进行了准确性测试和性能比较。

# 第二章 相关技术及使用工具

## 2.1 机器学习

机器学习开始于上个世纪五六十年代，近年来随着社会的发展，技术的进步，CPU的计算能力越来越强，并且数据量激增，这为机器学习提供了极为便利的条件，各个领域都对机器学习产生了浓厚的兴趣。机器学习是一门多学科多领域交叉学科，涉及到统计学、概率论、计算机科学等多门学科，主要通过对大量数据的分析，来使机器越来越智能，从而模拟甚至实现人类的学习行为。

机器学习分为有监督学习和无监督学习两大类。通过对已有的带标签的数据的分析，分析出数据的特征，并将这些特征与标签建立对应关系，训练出一个最优模型，然后通过这个模型对新的数据进行判断分类，这就是有监督学习。无监督学习，是给定一组不带有标签的数据，通过算法自己分析数据间的内在关系，然后把它们分成不同的类别，也就是聚类问题。

机器学习的应用前景十分广阔，在互联网、金融、医疗、教育等方面都有广泛应用，比如商品推荐，股票预测，AI看病，以及著名的阿尔法狗等等。随着机器学习的发展，机器学习也正在改变着我们的生活。

## 2.2 Pandas 和 Numpy

Pandas是一个Python软件包，提供快速，灵活和富有表现力的数据结构

Pandas非常适合许多不同类型的数据：

1. 具有异构类型列的表格数据，如在SQL表格或Excel电子表格中

2. 有序和无序（不一定是固定频率）时间序列数据。

3. 具有行列标签的任意矩阵数据（均匀类型或不同类型）

4. 任何其他形式的观测/统计数据集。（数据实际上不需要标记为放置在Pandas数据结构中）

而且，Pandas可以轻松处理数据中的缺失数据，插入和删除数据，数据对齐，标签切片等等，是一款强大、灵活的开源数据分析/操作工具。

NumPy是用Python进行科学计算的基础软件包。它包含以下内容：

1. 一个强大的N维数组对象

2. 复杂的（广播）功能

3. 用于集成C / C ++和Fortran代码的工具

4. 有用的线性代数，傅里叶变换和随机数能力

除了明显的科学用途外，NumPy还可以用作通用数据的高效多维容器，可以定义任意数据类型，在Python科学计算中应用十分广泛。

## 2.3 scikit-learn

scikit-learn简称sklearn，于2007年问世，是Python最重要的机器学习库之一，常被用于机器学习和数据挖掘等应用中。

sklearn依赖于matplotlib、NumPy和 SciPy，内置了丰富的机器学习算法，有效提高了机器学习的效率。此外，sklearn内置的大量的标准数据集也为开发者节省了不少获取数据和处理数据的时间。而且sklearn文档完善，API丰富，上手难度小，颇受开发人员的喜爱。

sklearn库主要包含了分类（Classification），回归（Regression），聚类（Clustering），降维（Dimensionality reduction），模型选择（Model selection），预处理（Preprocessing）六大功能。分类，即识别对象属于哪个类别，包含的算法有K最近邻，支持向量机分类，决策树，朴素贝叶斯，随机森林等，常用于垃圾邮件检测，图像识别等方向。回归，最主要是预测与对象相关联的连续属性，多应用于药物反应，股价预测等方面，主要包括线性回归，多项式回归，支持向量回归等算法。聚类就是将对象根据数据间的特征划归为不同分类，常用方法有K均值、mean-shift等。跟分类不同的是，分类是用带标签的数据训练出模型，然后判断新数据哪种类别，聚类的数据是不带有标签的，是完全根据算法分析数据中之间的相似性来对数据进行自动归类。降维，就是降低样本特征的维度，常用来提高计算效率，或者进行可视化。模型选择即比较、验证、选择参数和模型，目标是通过参数调整提高模型精度。预处理则是对数据进行一些操作，如提取数据特征、归一化、标准化、白化、去均值化、二值化等操作来满足计算需求。

## 2.4 Anaconda 和 PyCharm

Anaconda 是一个用于科学计算的 Python 发行版， 它包括250多种流行的数据科学软件包，比如大名鼎鼎的numpy和pandas，以及适用于Windows，Linux和MacOS的conda软件包和虚拟环境管理器。Conda使安装，运行和升级复杂的数据科学和机器学习环境（如Scikit-learn，TensorFlow和SciPy）变得简单快捷。

PyCharm是一款开发者常用的Python集成开发工具，由Jetbrains公司开发完成。PyCharm除了提供了调试，语法高亮，代码跳转等一些基本功能，还提供了智能代码完成，代码检查，即时错误突出显示和快速修复，以及自动代码重构和丰富的导航功能，而且PyCharm内置了多种集成的调试器和测试运行器。除了Python之外，PyCharm还为各种Python Web开发框架，特定模板语言等提供良好的支持。PyCharm还与IPython Notebook集成，具有交互式Python控制台，并支持Anaconda以及多个科学软件包，包括Matplotlib和NumPy等。实际项目的开发过程中，通过使用PyCharm IDE可以极大地提升开发者的开发效率。

# 第三章 情绪分析系统实现原理

## 3.1 数据预处理

直接收集获取到的数据往往存在一些缺失值、无效值、异常值等等，因为数据分析时，有些模型的构建对于数据比较敏感（比如朴素贝叶斯，比如决策树），对数据要求比较高，所以难以直接使用这些数据进行数据特征的提取以及后续的数据分析。常见的数据预处理一般包括数据清洗、数据集成和数据变换。

### 3.1.1 数据清洗

1. 孤立值处理：筛选出数据中与周围数据的范围明显不符的异常值，即孤立值，分析其产生原因，对于错误数据，通常采取空值替换，直接删除，平均值修正等处理。

2. 缺失值处理：我们收集来的大量数据难免有属性缺失的情况存在，我们一般的处理方法有中位数替代法，平均值替代法，频率最高值替代法，默认值替代法，邻近值替代法或者根据需要直接删除属性缺失的行或者列。

3. 噪声平滑：对于收集来的数据中的噪声问题，通常先采取分箱方法，将数据按照一定规律排好序，然后依据等深分箱原则或者等宽分箱原则将数据分成几组，然后根据需要，选择边界平滑，均值平滑，中值平滑中的一种或者几种对数据进行处理，即使用每组中的边界值、平均值或者中值代替同一组的数据。

### 3.1.2 数据集成

数据集成是把多组源数据融合成一组数据，这多组源数据可能来自多个不同的数据库或者不同的文件，所以集成的过程中要消除数据不一致和数据冗余。数据不一致主要表现为属性名称不一致或者数据矛盾。数据冗余一般是同一属性名称多次出现或者属性间存在线性关系。一般通过相关性分析来消除属性间线性相关。

### 数据变换

数据变换主要是把数据标准化，消除量纲的影响，比如把属性按比例缩小，把属性放到一个。常用的数据变换方法有最小最大规范化、Z-score标准化、对定量特征二值化。

## 3.2 特征选择与特征提取

通常我们收集来的大量原始数据，数据之间的特征并不是很明显，我们需要对收集来的数据进行特征提取，使之具有统计意义（如平均值，最大值，最小值，方差等）或者将数据转换为具有明显物理意义的特征（如几何特征，纹理特征，Gabor等）。而特征选择则是从已有的数据特征中选择出最能代表该组数据的一组数据特征，该组特征是已有数据特征集的子集。特征选择通过选择能够提供更高的准确性的特征来帮助研究人员创建精确的预测模型，同时需要更少的数据。

通过特征提取有可能发现更有意义的隐性变量，从而对数据有更深的理解。

特征选择方法则可以用来识别和去除不需要预测模型准确性的数据中不需要的，不相关的和冗余的属性，降低噪声的影响，减少数据的存储，但是一定程度上也存在着降低模型准确性的可能。

### 3.2.1 特征提取

实际的研究过程中，特征提取主要包含两部分内容，一个是时域特征的提取，另一个是频域特征的提取。对于这两种特征一般采用滑动窗口的方式进行特征提取，时域特征提取的特征一般包括最大值、最小值、众数、均值、过均值点个数、方差（或者标准差）、相关系数等。频域特征一般需要提取直流分量DC，幅度，功率谱密度PSD等。

均值：即一组数据中的平均数，

方差：一组数据中每个样本与该组数据平均值的偏离程度，

滑动窗口：滑动窗口有窗口大小和移动步长两个变量。一次处理的数据量被称为窗口大小。窗口大小一般设置为2a，步长设置为窗口大小的一半，其中传感器采样频率为a Hz。实际情况中，为了在提取频域特征时，可以顺利的进行傅里叶变换，窗口大小通常选择2的n次方（n为正整数），即，

窗口大小 =

加速度合成：加速度传感器通常有x，y，z三个轴，每个轴上有各自的值，代表加速度的三个分量。在实际情况当中，除了个别需要单独读取每个轴的数据的特殊情况，一般可以将三轴的分量合并成一个加速度值，之后的特征提取、模型训练全部用合成后的值进行操作，这样可以在降低计算复杂度的同时，不损失精度。

其中，a代表合成后的加速度，，，分别代表x，y，z三轴上的分量加速度。

众数：一组数据中出现次数最多的数就叫做众数。

最大值：一组数据中最大的数，其他值都小于等于该值。

最小值：一组数据中最小的数，其他值都大于等于该值。

范围：最大值和最小值的差值就是范围。

过均值点个数：一个窗口内超过均值点的个数。

相关系数：又称皮尔逊系数，用来表示两个变量之间的相关性。

其中，表示y与z的协方差，，分别代表y和z的方差。

幅度：就是变换后数据的绝对值。

直流分量：非正弦周期信号进行傅里叶变换后，频率为零的分量。

### 3.2.2 特征选择

特征选择算法有三大类： Filter方法， Wrapper方法和Embedded方法。

Filter方法：为不同的特征分别赋予权重，即所谓的“打分”，通过不同的“打分”来表示不同特征的重要性，然后按照“打分”排序。常用的方法有卡方检验法，信息增益法，相关系数法。

Wrapper方法：把特征的选择过程看作是搜索问题，然后评估比较不同的组合。搜索过程可能是井然有序的，比如最佳搜索，也可能是随机的，例如随机爬山算法，也可能使用启发式算法，如前向和后向遍历来添加和删除特征。常用方法有递归特征消除算法。

Embedded方法：选出模型创建过程中对模型的准确性有最大的贡献的特征。最常见的方法是正则化方法，正则化方法也称为惩罚方法，它将额外的约束引入预测算法（如回归算法）的优化中，从而将模型偏向低复杂度（更少的系数），正则化算法的例子是LASSO，Elastic Net和Ridge回归。

## 3.3 主成分分析（PCA）

### 3.3.1 主成分分析概述

在一般的研究分析过程中，常常需要对反映研究对象特征的多个属性进行大规模的数据收集来进行规律的发现和寻找。毫无疑问，样本容量大，属性多可以为研究和分析过程提供给更多的有用信息，为分析数据中的规律提供便利。但与此同时，也带来了数据收集整理工作量增加等问题，而且多维度数据的特征提取极容易造成维度危机，给分析工作造成极大挑战。而如果只分析单个指标，又会导致分析是孤立的，缺少全局性和综合性。单纯的为了降低工作量去减少分析指标，很容易得出错误的分析结果。因此，需要找到一个有效方法，在尽可能少丢失原属性带有的信息的同时，降低需要分析的属性个数，以达到既能全面分析所收集的数据，又能在一定程度上降低工作量的目的。由于各属性之间不是完全孤立存在的，这就为我们用较少的属性表现全体属性所携带的规律提供了可能。主成分分析就是这样一种降维方法。主成分分析在日常研究中应用十分广泛，在统计学，地理学，经济学，社会学等多个领域都有应用，是一种常用的多元数学分析方法。

### 3.3.2 PCA降维原理

一些高维度数据，数据中的不同属性之间，常常有一定的关联。为了解释降维原理，以淘宝的交易数据为例，假设订单中有商品编号，购买数量，商品单价，商品总价四个属性，那么明显可以看出商品总价可以由单价和数量的关系反映出来，所以这项属性就是冗余的。删除掉这项属性，也就是降低了维度，并不会损失数据的信息，但却可以有效降低数据信息的分析、整理、计算过程中的工作量，有效降低后面构建模型时模型所需的空间时间开销。我们降维的目的就是将近乎无损的数据信息用尽可能少的属性或维度表现出来，最后的效果就是各属性间不存在线性关系。

PCA降维是通过把高维数据投射到低维子空间进行降维，比如把二维平面的点投射成线，数据集中的值由两个值变成一个值表示，这就实现了二维数据的降维。同理，把三维数据投射到一个平面，就实现了三维数据到二维数据的降维操作。n维数据集以此类推，通过映射，将n维数据降维到k（k<=n）维子空间。

降维步骤：

假设原数据是m行n维，首先，由原数据生成n行m列的矩阵。然后对训练集中的数据特征进行归一化处理，然后计算协方差矩阵，接下来要求出协方差矩阵的特征向量和特征值，这里要用到奇异值分解算法。最后一步，进行降维计算，通过获得的降维矩阵，就可以将高维数据映射到低维子空间了。

经过降维之后的数据，在低维下更容易处理，更容易使用。相关特征，特别是重要特征更能在数据中明确的显示出来；如果只有两维或者三维的话，更便于可视化展示。此外，应用PCA进行特征降维还可以去除数据噪声，降低算法的开销。

## 3.4 机器学习分类器

### 3.4.1 k-NearestNeighbor算法

K最近邻 (k-Nearest Neighbors，KNN) 算法工作原理：一组每个数据都带有标签的数据集，被称为样本集。样本集中的数据和其对应分类是已知的。输入样本集后，通过对样本集数据特征进行分析，训练出模型。等再输入不带标签的新数据时，提取出新数据的数据特征，与训练好的数据模型进行比对，从训练集中提取出k个与新数据最相似的样本的标签，选出这k个数据里面比例最高的标签作为新数据的标签，从而实现分类的目的。通常情况下，k的取值不大于20。 K最近邻的主要过程：

1、计算测试对象到训练集中每个对象的距离

2、按照距离远近排序，这个距离可以是欧式距离，马氏距离，曼哈顿距离。

3、选取与当前测试对象最近的k的训练对象，作为该测试对象的邻居

4、统计这k个邻居的类别频率

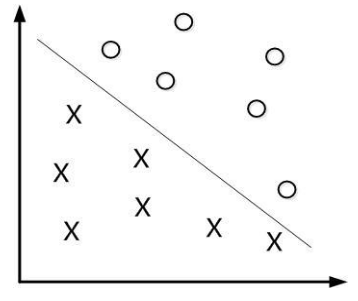
5、k个邻居里频率最高的类别，即为测试对象的类别

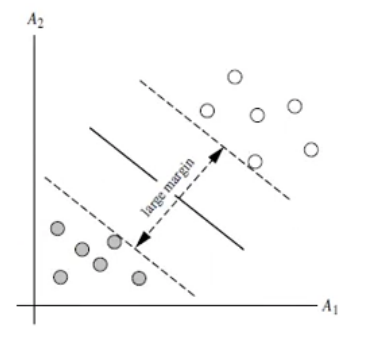
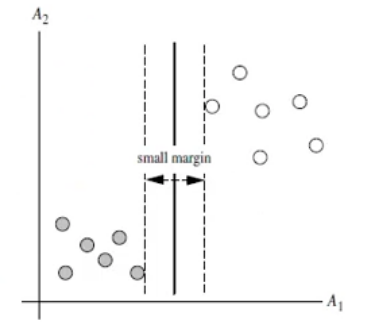
k值选取较小时，也就意味着使用待分类点周围较少的邻居点进行预测，比如极端值k=1,那么也就是由待分类点最近的一个点来对该点进行分类，如果一旦该邻居点是噪声点，那么就会对预测结果造成较大误差。K值选取较大时，可以有效降低噪声的影响，但是容易导致分类界限不明显。k的具体取值一般与数据情况有关，不合理的k值会在一定程度上产生过拟合或者欠拟合的问题。具体的k值选择可以通过经验判断或者交叉验证来确定。

### 3.4.2 SVM 算法

SVM（Support Vector Machine）,也被称为支持向量机。支持向量机是一种二分类模型，处理的数据可以分为三类：线性可分，近似线性可分，线性不可分。SVM的分类原理是寻找一个最优的超平面将数据进行分类，使边界最大。在二维空间上，这个超平面是一条直线，在三维空间上，这个超平面是一个平面，在更高维的空间上，是一个超平面。根据数据可分的三种情况，SVM有以下几种分类方式：

如果数据线性可分，那么选择硬间隔方法使边界最大化，通过学习线性分类器来完成这一分类过程。硬间隔对应于线性可分数据集，可以将所有样本正确分类，也正因为如此，受噪声样本影响很大。

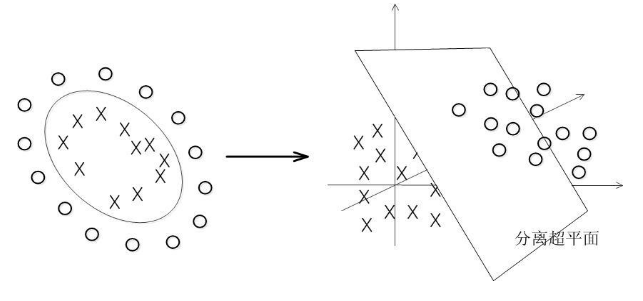




如果数据近似线性可分，那么选择软间隔方法使边界最大化，通过学习线性分类器完成这一分类过程。软间隔对应于通常情况下的数据集（近似线性可分或线性不可分），允许一些超平面附近的样本被错误分类，从而提升了泛化性能。因为不是所有情况都需要把点全部分对的，有时候样本点中存在一些本来就是错误的数据，也就是噪声，学习过程中如果学习了这些噪声，就会出现过拟合的情况，降低模型预测的准确性。这个过程需要加入惩罚因子C，使得点被错分的情况更合理。



如果数据线性不可分，那么就通过核函数以及软间隔最大化的方式，学习非线性分类器实现这一分类过程。当数据线性不可分的时候，SVM通过将低维数据向高维空间转化实现线性可分。



对于线性不可分的数据集的任意两个实例：，。选取特定映射f之后，使得f()与f()在高维空间中线性可分，运用上述（近似）线性可分问题的求解方法，可以发现目标函数和分类决策函数只涉及内积<f(), f()>。由于高维空间中的内积计算非常复杂，我们可以引入核函数K(, )=<f(), f()>，因此内积问题变成了求函数值问题，由高维运算变成了低维运算，有效降低了计算复杂度。而这个过程中，我们不需要知道映射关系是怎样的，只需要将核函数应用到原数据上就可以了。常用的核函数有：

多项式核函数：多项式核函数应用于将低维数据向高维空间转化，缺点是当多项式阶数比较高时，核函数计算复杂度会非常高，甚至无法计算。

线性核函数：主要用于线性可分的情况，其参数较少，运算复杂度低，通常我们选择线性核函数作为首先尝试的核函数来看看效果，如果准确率不够再换其他的。

高斯径向基核函数：也是将低维数据转换到高维空间，但本身参数相比多项式核函数要少，所以在大样本和小样本时都有很好的性能。使用最广泛的一种核函数，在不确定用哪种核函数时，推荐使用高斯径向基核函数。

sigmoid核函数：

### 3.4.3 朴素贝叶斯算法

朴素贝叶斯算法一种简单而且效果比较不错的弱分类器，其理论基础是概率论中的贝叶斯理论。朴素贝叶斯算法虽然构建简单，分类效果却很优秀，甚至比许多复杂算法还要高效，尤其是在大型数据集，表现更佳。之所以说其朴素，是因为朴素贝叶斯算法是基于各个样本特征相互独立的假设的。举个例子，比如一个男生具有长得高，皮肤白，性格好的特点，我们得出结论该男生受女生喜欢，虽然可能这些特征之间具有一定的关联，或者相互依赖，但在朴素贝叶斯算法看来，这些特征在判断男生是不是受女孩喜欢的问题上，特征之间是相互独立的，并且对事件的影响是相同的，即权重相同。

先验概率：在事件发生前，基于历史事件统计，或者背景常识，或者人的经验判断得出的事件可能发生的概率。比如，天空中阴云密布，历史上天空中出现阴云密布的情况会下雨的可能性是70%，那么70%就是先验概率。

后验概率：在事件发生后，依据事件发生的结果反推该事件是由某因素引起的概率，即执果寻因。举个例子，中午吃了苹果，下午肚子疼，我们想算一下肚子疼是由吃苹果导致的概率，这就是后验概率。

贝叶斯理论：

事件X在事件Y发生的条件下的概率，与事件Y在事件X发生的条件下的概率是不一样的，贝叶斯公式就是用来描述这种关系的。

其中P(X|Y)是在Y发生的情况下X发生的概率。X，Y都是事件，并且P(Y)不为0。P(X)是事件X发生的先验概率，P(X|Y)是B的后验概率。

当上式被如下表示，，y表示类变量，X是特征向量：

由朴素贝叶斯的朴素假设，也就是每个特征变量之间相互独立，即有：

所以，前面式子又可以表示为：

由于输入数据与分母是常量相关，于是进一步推导：

所以，我们的目的是选择出类变量y的所有可能值的中使概率最大的那个可能值，可以用公式表示为：

最后，通过与的计算得出结果。

朴素贝叶斯分类器具有算法逻辑简单，实现难度小的的优点，而且因为朴素贝叶斯分类器建立在特征相互独立的假设下，只涉及二维存储，所以分类过程中时间复杂度低。朴素贝叶斯分类器在样本特征比较少，并且特征之间相关性小时，具有较为良好的分类效果，而且理论上，相比其他分类方法，朴素贝叶斯算法误差率最低。但实际应用中，情况往往比较复杂，样本特征比较多的时候往往难以保证特征间相互独立的假设，分类效果不好。

### 3.4.4 决策树算法

决策树应用十分广泛，既可以用于回归也可以用于分类。分类问题中，决策树采用树形结构对样本的属性进行分类，既可以处理离散（if-then）的特征空间，也可以处理连续的特征空间，只需将连续空间通过阈值化变为if-then形式即可。决策树由边和结点构成，内结点代表属性和特征，外结点代表类别。边代表判别的规则，即if-then规则。决策树主要分为特征选择、生成、剪枝三步。根据特征选择度量方式不同，分为ID3、C4.5、CART三种决策树算法，它们对应的度量方式分别是信息增益（Information gain）、增益比率（gain ratio）、基尼指数（Gini index）。

信息增益：

首先是熵的定义，

变量的不确定性越大，熵也越大。

信息增益，即信息获取量（例如，通过a作为结点来分类获取了多少信息）：

依次比较各个属性的信息增益，选择最大的那个属性作为根结点，然后对于后面的结点，依次重复这个过程，直至最后给定结点的所有样本属于同一类或者没有剩余属性可以进一步划分样本，迭代停止。

增益比率：

其中代表的是a属性的x种情况中某一种在样本中的比例，即。增益比率，也就是某属性增加的信息熵与某属性自有信息上的比率。选择增益比率高的那个作为根结点。

基尼指数：

基尼指数（基尼不纯度）= 样本被选中的概率 \* 样本被分错的概率

样本集分为K类，表示选中的样本在k类别中的比例，此处看作样本是k类别的概率，则这个样本被分错的概率是(1-)

属性a的基尼系数，

选择基尼系数低的属性作为根结点。

决策树的优势在于它的数据形式非常容易理解，而且能够给出数据间的内在关系。除此之外，决策树计算复杂度不高，对中间值的缺失不敏感，可以处理不相关的数据特征。而且，决策树对数据数据形式要求简单，不必像其他分类方法一样统一数据属性，既可以是数值型，也可以是标称型。但缺点是容易拟合过度，处理连续变量效果不好，类别较多时，错误会增加的比较快等。

### 3.4.5 Adaboost算法

Boosting是一种集成学习技术，也被称为增强学习法，可以通过训练出多个预测精度较低的弱分类器，然后把这些组合起来成为一个分类精度高的强分类器。在难以直接构造出高预测精度分类器的情况下，这种方法无疑为研究人员提供了一种行之有效的算法设计思路。其中Adaboost是boosting算法中一种十分成功的应用。

Adaboost算法，即自适应增强算法，是一种迭代算法。通过对同一数据集迭代训练不同的分类器，每次找到一个最优的分类器，然后下一次迭代时增大前一个分类器错误分类样本的权值，减小正确分类样本的权值。最后将得到的多个最优的分类器组合起来就得到一个强分类器。

Adaboost算法实现步骤：

1、初始化各样本数据的权值。假设有N个训练数据，第一次开始迭代时，各个样本被赋予相同的权值

2、对数据集迭代训练弱分类器。如果训练过程中，某个训练数据被准确分类，那么在下一次迭代过程中，降低该训练数据的权值，同时提高被错误分类的训练数据的权值。一次迭代过程完成后，使用权重值更新后的训练数据集进行下一次迭代，构造新的弱分类器。如此迭代下去，完成整个训练过程。

3、集成各个弱分类器构建一个新的强分类器。为了让分类准确率高的弱分类器发挥更大的作用，按照分类过程中各个弱分类器的误差大小情况，为各个弱分类器分配权重。误差率越小的分类器，在构建强分类器的过程中，所占权重越高，否则，所占权重越小。这样，一个强分类器就构建完成了

Adaboost是一种简单有效的分类算法，很好地利用了不同弱分类器进行级联，并且在构建过程中充分考虑了不同分类器的权重问题，分类结果精度高。主要缺点有，分类精度可能会受数据不平的影响而下降，时间复杂度高，弱分类器的数目也就是迭代次数不易确定。

# 第四章 数据处理过程

本实验主要研究内容是，以Android设备为信息收集载体，通过获取Android用户的运动信息、手机使用情况、用户情绪状态，并以此进行分析，探索发现用户的各项信息与情绪状态的内在联系，并建立模型，实现通过各项数据识别用户情绪。实验步骤主要包括：数据的获取、数据预处理、特征提取、PCA降维、训练模型与预测、参数调优、结果对比。

## 4.1 用户数据收集

本实验由10名使用Android设备的同学参与志愿活动，进行数据的收集。每日的数据收集分为早中晚三部分，在早上8：00，中午12：00，下午6：00，分别进行一次用户情绪录入，并与此同时进行离散采样，每隔一分钟进行一次采样，每次采样时间持续十分钟。志愿者的心情部分主要包含高兴、平静、难过、愤怒四种基本情绪类型，由志愿者凭感受主动录入。主要采样信息为Android设备各项传感器数据和设备基本情况信息。其中需要采样的传感器部分包括加速度传感器、方向传感器、陀螺仪传感器、磁场传感器、重力传感器、线性加速度传感器、GPS传感器、光线传感器、的数据。设备情况信息的采集部分包括手机的网速情况和情景模式。收集到的数据以txt格式存储，保存在用户手机内存卡根目录。整个采集过程持续两星期。

收集的各项信息数据格式如下：

（加速度x，加速度y，加速度z，方向x，方向y，方向z，陀螺仪x，陀螺仪y，陀螺仪z，磁场x，磁场y，磁场z，重力x，重力y，重力z，线性加速度x，线性加速度y，线性加速度z，GPS经度，GPS纬度，光照，网速，情景模式，情绪状态）

将收集到的原始数据转换成excel表格，情况如下：







## 4.2 用户数据预处理

我们收集来的大量原始数据，是不能直接被我们使用的。收集来的原始数据有可能会存在包含大量噪声，数据不完整（比如有的属性可能缺失，或者不确定），数据不一致（比如在不同表中的同一属性名称不一致或者数据矛盾），度量单位不一致等问题。可能产生的原因有，比如收集过程中设备使用方法不正确，用户操作不当，或者设备出现故障和异常，信息收集过程受到干扰和中断，再比如数据收集后数据存储不当，工作人员误操作等等。这就需要我们通过科学的数据预处理技术对数据进行清洗，剔除坏值，非法值，异常值，填补空值等操作，来消除数据中的噪声，保证数据数据一致性及数据完整性，保证收集来的数据有效和可用。除此之外，我们还需要数据进一步进行集成，规约，变换等操作，使得数据满足我们研究分析的要求。

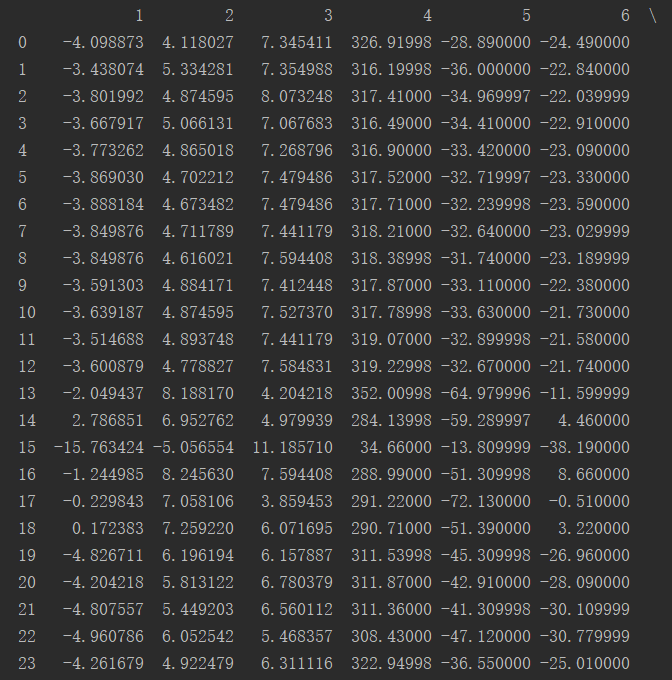
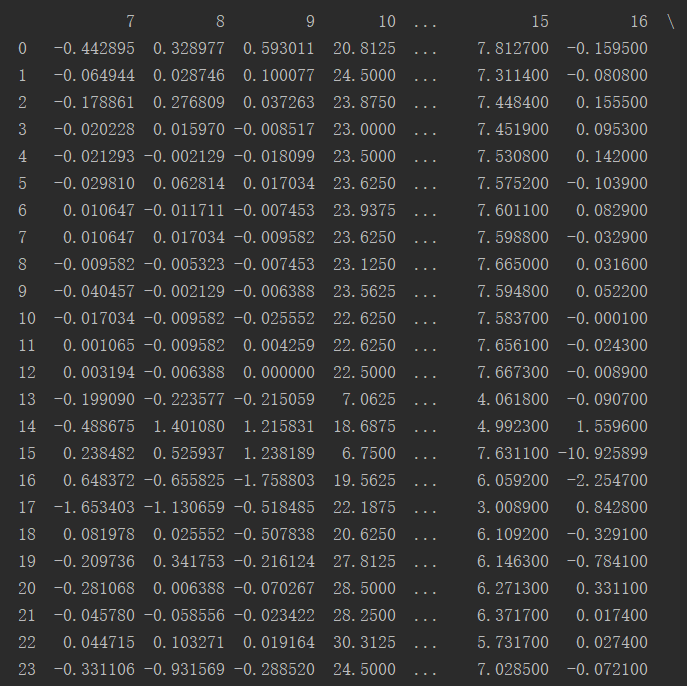
数据的预处理可以显著提高数据的质量，同时可以有效地提高后续过程中数据分析的效率。Python中提供了强大的pandas和numpy库，我们使用这两个科学的数据分析库进行数据的预处理操作。

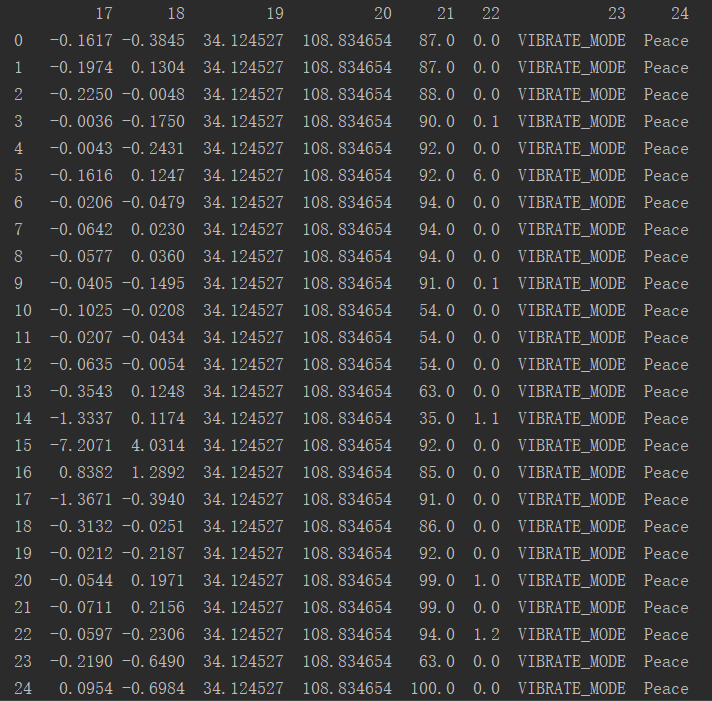
### 4.2.1 数据集成

数据集成是把多组源数据融合成一组数据，这多组源数据可能来自多个不同的数据库或者不同的文件，所以集成的过程中要消除数据不一致和数据冗余。数据不一致主要表现为属性名称不一致或者数据矛盾。数据冗余一般是同一属性名称多次出现或者属性间存在线性关系。一般通过相关性分析来消除属性间线性相关。此处主要介绍相关的数据拼接和合并，有关属性间的相关性造成的数据冗余在后面单独会有介绍。

本实验由于是从多个用户收集信息，每个用户收集的信息保存在不同的文件中，所以需要将多份数据合并在一起。我们首先将多份文件通过pandas读文件方法将txt文件读入，分别存储为不同的DataFrame（Pandas中的一种特殊数据结构，表现形式为二维数组），然后将不同的DataFrame拼接在一起。其中需要注意的是，防止数据不一致和数据冗余，不同文件的数据格式要保持一致，同一个属性在不同的文件中是否有不同属性名。因为数据分几次收集，前期收集到一部分数据后，又进行了数据收集工具的改进，数据格式存在一定的调整，在进行拼接时，我们使用pandas调整了属性顺序，然后进行拼接操作。

数据拼接后结果：



### 4.2.2 缺失值处理

通过numpy的isnan方法（numpy中的查询空值的方法，返回DataFrame的空值和非空值情况）发现，收集来的数据中明显存在一些缺失值，可能造成这一现象原因可能有用户忘记填写、应用被关闭或者后台清理、手机关机或没电等，也有可能是应用出现bug，未能正常收集数据。如果有连续多个缺失值，应当采取的操作是将连续空缺的几组数据删除。如果是个别属性不连续的出现缺失，我们一般的处理方法有中位数替代法，平均值替代法，频率最高值替代法，默认值替代法，邻近值替代法或者根据需要直接删除属性缺失的行或者列。考虑到人的心情在一段时间内是相对稳定的，所以各项数据也应该是基本稳定的，在这里我们采取同组的平均值进行缺失值的替换。

## 4.3 数据归一化

收集来的数据因为单位不统一、量纲不统一，是无法直接用来分析的。因为分析时往往不清楚各个属性对于结果的影响，所以一般假设各个属性对于结果的影响是相同的，即权重相同。但如果因为量纲不统一，就可能导致样本之间数量级不统一，有的数据很大，有的数据很小，直接进行分析的话就会导致数值大的属性对结果影响过高，会影响到模型的准确性，所以需要把样本划归到一个特定区间。本实验中的数据处理部分也应用到了归一化操作，但按照实验顺序，该操作在后面才会被执行，所以详细介绍见第五章的5.6小节。

# 第五章 特征选择与特征提取

## 5.1 数据属性观察

通过对收集来的数据进行直观分析，初步判断用户数据中的GPS经纬度信息不具有明显变化，造成这一现象的原因可能有用户群体大多是在实验室进行毕业设计的大四同学，一天中待在实验室的时间较长，活动范围有限，基本上存在变化的时刻集中出现在中午就餐时刻，但由于手机内置的GPS传感器精度有限，无法明显捕捉到这样小范围的移动。然后，提取出每组数据中的GPS经纬度信息，使用MATLAB进行绘图，显示出的情况与人为判断基本一致，近乎集中于两三个点。鉴于这种情况，我们判断这两个属性对于后面模型构建以及预测分析不具备太大的参考价值，故选择直接删除这两项。

图：GPS经纬度情况

## 5.2 属性的相关性分析

为防止数据线性相关造成的数据冗余，本实验对初始获得的24个变量中，可能出现线性相关的变量之间，进行了相关性分析。

相关性的计算方法：

属性A和属性B的相关性计算，

如果，则有A与B正相关。

如果，则有A与B负相关。

如果，则有A与B相互独立。

如果很大，那么就说明A，B相关性很强，可以删除其中一个。

## 5.3 加速度合成

本实验中使用到的加速度类传感器有加速度传感器和线性加速度传感器，每种传感器都有x，y，z三个轴，因为本实验中不需要区分具体的方向，所以把两类加速度传感器的x，y，z三轴的分量数据（即加速度x，加速度y，加速度z，线性加速度x，线性加速度y，线性加速度z六个属性），分别进行合成，合成后形成两个新的加速度属性——加速度和线性加速度，特征合成以后的操作都使用新特征进行操作，不再使用旧特征，并将旧属性从数据列表中剔除。将这几项特征合成后，使用合成后的值进行操作，可以有效减少训练模型时的时间复杂度，而且不损失精度。

加速度合成公式：

其中，a代表合成后的加速度，，，分别代表x，y，z三轴上的分量加速度。

## 5.4 数据特征提取

直接收集来的数据特征可能数据关系不够明显，考虑到人的心情是由一段时间的一个平均状态来表现的，所以对收集来的数据，采用分箱技术进行切片分组，以10为单位分组，对每一组数据求出平均值、最大值、最小值、方差作为新的特征。

最大值：一组数据中其他值都小于等于数据中的某个值，这个值就是最大值。

最小值：一组数据中其他值都大于等于数据中的某个值，这个值就是最小值。

均值：即一组数据中的平均数，

方差：一组数据中每个样本与该组数据平均值的偏离程度，

## 5.5 特征选择

特征是否发散：如果一个特征不发散，例如方差接近于0，也就是说样本在这个特征上基本上没有差异，这个特征对于样本的区分并没有什么用。

## 5.6 数据标准化

数据标准化主要用来消除量纲的影响，比如把属性按比例缩小，把属性放到一个特定的区间。我们收集到的数据包括运动类信息、手机状态类信息、环境信息三大类，其中运动类信息包含了加速度、方向、陀螺仪、重力等传感器数据，手机状态类信息主要包含了网速，环境信息主要包含了光强、磁场强度等数据，经过特征选择和特征提取后，形成了新的48维特征向量，其中的数据量纲存在巨大差异。在本实验中我们假设每种特征都是与人的情绪状态相关的，并且对情绪状态的影响系数是相同的，为了防止某一特征所占的权重过高，从而过多的影响模型的构建，我们需要对数据进行标准化操作。在本实验中，我们选择了Z-Sorce标准化将数据进行了规范化处理。常见的标准化操作还有最大最小规范化，以及对定量特征进行二值化。

1. Z-score标准化：

根据属性的均值和方差来对属性进行规范化，一般在最大最小化规范化出现异常数据时使用。

其中的和分别为属性A的均值和方差。

2. 最小最大规范化：

已知属性区间，将属性的取值范围由[old\_min,old\_max]映射到[new\_min,new\_max]

该种方法保留了原来数据中存在的关系，但如果将来遇到超过目前属性[old\_min,old\_max]取值范围的数值，将会引发错误。

3. 对定量特征二值化：

对定量特征进行二值化之前，需要预先设定一个阈值a，如果比阈值大就赋值为1，如果比阈值小就赋值为0。

## 5.7 主成分分析

经过前面的数据预处理和特征工程，我们将原数据整理成了的样本集，为了体现整个过程的科学性和严谨性，也为了进一步从中已有的样本中提取对结果影响比较大的有关变量，减小无关变量和噪声的影响，以及降低计算量，我们对获得的新样本集进行了主成分分析操作。通过调用sklearn.decomposition模块的PCA方法，生成一个PCA实例。PCA降维可以将数据降到指定维数，但是考虑到将数据样本降低越多固然会降低更多的计算量，但是也可能导致预测准确度过低的问题，所以一般降维时多选择指定降维后的最小精度（即保证降维后，新数据集保留的原数据集信息在指定精度以上）或者设置为“mle”，由PCA函数自动确定降低的维数。因为本实验中不知道降低的维数对于精度的影响，所以不容易指定维数，本实验中选择设置为“mle”方式，PCA函数自动选择最优的降维处理。经过PCA降维处理后，得到了新的46维样本集。

## 5.8 本章小结

经过前面的数据集成、缺失值处理保证了数据的一致性和完整性，通过属性的相关性分析消除了线性相关的变量，通过加速度合成将重要性不高的属性进行了合并，有效降低了数据的冗余和计算复杂度，特征选择和特征提取减少了无用属性，从原有数据中提取出了最能代表数据特征的属性集合，有效降低了噪声带来的影响，保证了后面模型训练的精度。我们收集来的原始数据涉及23个属性，经过这部分的处理，融合成了新的48维特征向量，然后经过主成分分析进行进一步降维，降低运算的复杂度，得到了46维特征向量。

# 第六章 构建情绪分析模型

通过前面对数据进行数据预处理操作，我们得到了一个干净的、有效的样本集，经过特征选择和特征提取操作，我们获得了最能表现样本特征的特征属性，接下来就需要通过这些特征构建情绪的分析模型。从本质上说，这属于机器学习中有监督学习部分的分类问题，所以我们的目的也就是选择一种合适的分类器，将情绪准确地分类。

机器学习中的分类算法主要有K近邻算法、支持向量机算法、朴素贝叶斯算法、决策树算法、随机森林算法、Adaboost算法。本实验主要选取了K近邻算法、支持向量机算法、朴素贝叶斯算法构建情绪分析模型并进行分类。

## 6.1 K近邻分类模型

K近邻分类模型是基于K近邻算法构建的。K近邻算法原理是通过比较待测点与带有标签的样本点的距离，选择K个与待测点距离最近的样本点，统计这K个点中，选择比例最高的标签作为待测点的标签。K近邻算法具有理论成熟、简单好用、测试准确率较高、对异常值不敏感等优点。而且考虑到K近邻算法要求数据量不能太大，否则会导致计算量过大，也不能数据量太小，这样会容易导致误分，本实验数据量刚好满足这样的要求，基于上述考虑，本实验中选择用K近邻分类模型来构建第一个情绪分析模型。

本实验中将收集来的数据按照80%和20%的比例随机分成两部分，80%部分用来作为训练集，20%部分作为测试集。

K近邻算法中最重要的部分是K值的选择，不同的K值会对K近邻模型的分类准确度产生较大影响。通常情况K值不应大于20，而且为避免在通过K个标签分类时产生相同比例的标签，K值一般选择奇数值。具体的K值选择可以通过经验判断和交叉验证选择。本实验中，我们通过交叉验证选择K值。该步骤通过sklearn.model\_selection模块的cross\_val\_score方法实现，通过设置K值的范围，本实验中将该范围设置为了1~31，然后比较不同K值下分类准确率的变化，并通过matplotlib模块的pyplot方法将不同K值下的分类准确率用折线图的方式绘制出来，选择出准确率最高的K值。

然后，通过sklearn.neighbors模块的KNeighborsClassifier方法生成一个K近邻分类器。首先，将K值设置为前一步通过交叉验证获得的K值。然后设置K近邻算法的实现方式。K近邻算法的实现方式有枚举实现、KD树实现。枚举实现，即暴力实现，通过挨个搜索待测点距离每个样本点的距离，然后选出K个最近邻，这种方式计算量较大，只适合小数据样本。KD树实现方式没有直接计算待测点距离每个样本点的距离，而是先把数据存储进一个KD树，此处的K是指K个特征，根据构建好的KD树模型再进行距离的计算，可以有效减少计算量，提高分类的效率。此处我们设置参数algorithm为auto，即由分类器自动选择效率最高的算法。默认情况下，K个近邻的权重是相同的，但实际情况中可能不是这样的，通常距离待测点越近的样本点的标签越具有参考性，本实验中，我们按照K个近邻距离反比为K个近邻赋予权重。

经过前面步骤，我们已经获得了一个K近邻分类器，然后传入训练集X\_train和训练集标签y\_train，即可由分类器训练出K近邻分类模型。

## 6.2 SVC分类模型

SVC分类模型是基于支持向量机算法构建的。支持向量机算法原理是，寻找能够将训练数据集进行正确划分，并且使得几何间隔最大的最优超平面。但是支持向量机理论上仅支持二分类，而情绪具有多种类型，将数据划分为不同情绪类型就是多分类问题。支持向量机解决多分类问题的策略是将多个二分类器组合起来实现一个多分类器，本实验中我们使用了基于支持向量机实现的SVC（C-Support Vector Classification）来解决情绪的多分类问题。

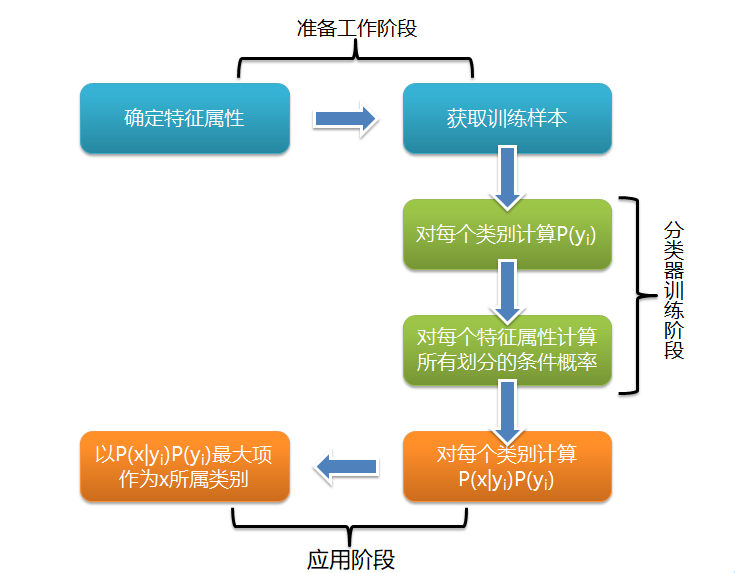
首先，我们将收集来的数据按照80%，20%比例划分为训练集和测试集。

然后我们通过sklearn.svm模块的SVC方法生成一个SVC分类器。因为SVC是基于支持向量机算法的，所以使用了支持向量机解决多分类问题的策略，一对一（one-against-one）和一对多（one-against-rest）。一对一策略就是在任意两类样本之间构建一个支持向量机，K个类别的样本之间就要构造个支持向量机。然后对未知样本进行分类时，选择得票数高的类型作为未知样本的类型。由于这种方式需要构建的支持向量机太多，影响了模型构建的效率，所以本实验中，我们选择一对多（one-against-rest）的策略构建情绪分析模型，即依次将某个情绪类型的样本归为一类，然后将剩余的类别归为一类，这样4个情绪类型的样本就构造出了4个SVM，分类时将未知样本分类为具有最大分类函数值的那类。使用SVC方式进行情绪分类，实验中首先选择了线性核函数作为核函数，因为线性核函数简单，计算量小，而且具有不错的分类效果。因为考虑到情绪相关的数据可能比较复杂，简单的线性核函数可能并不能得到最好的分类效果，实验中还选择了多项式核函数和高斯径向基核函数对情绪数据进行分类。

最后，将训练集数据及训练集对应的标签传入构造好的支持向量机分类器中，由分类器进行训练，就得到了基于支持向量机的情绪分类模型。

## 6.3 朴素贝叶斯分类模型

朴素贝叶斯分类模型基于朴素贝叶斯算法，这个模型比较简单。朴素贝叶斯算法基于各个属性对于结果都有影响，并且权重相同的假设。根据计算方式的不同，朴素贝叶斯有三种实现，高斯朴素贝叶斯、多项式分布朴素贝叶斯、伯努利分布朴素贝叶斯。本实验中选择使用这三种朴素贝叶斯算法进行模型构建。

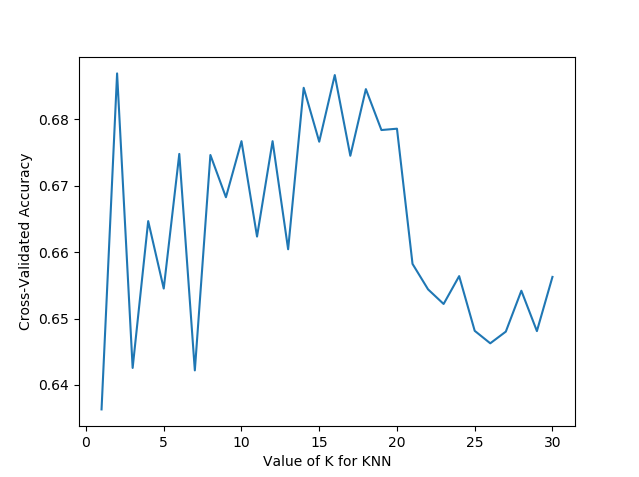
基于朴素贝叶斯分类模型的主要构建步骤如下图，

首先，我们将收集来的数据按照80%，20%比例划分为训练集和测试集。然后，我们通过sklearn.naive\_bayes模块的GaussianNB、MultinomialNB、BernoulliNB生成三个朴素贝叶斯分类器，接下来，将已经得到的训练集数据和训练集标签数据分别传入三个分类器，就得到了三种朴素贝叶斯分类模型。

# 第七章 运行与测试

## 7.1 K近邻算法交叉验证k值-准确率

下图显示了通过交叉验证得到的不同K值下，K近邻分类模型准确率变化情况：



## 7.2 准确性测试

准确性测试主要是针对不同情绪分析模型，输入相同的样本，比较输出结果与真实结果。

下图是不同分类器对于最后10个数据的预测值与真实值：



对于训练集最后的10个数据，K近邻、SVC(linear)、SVC(poly)、SVC(rbf)、高斯朴素贝叶斯、多项式朴素贝叶斯、伯努利朴素贝叶斯六个情绪分析模型，分别识别正确了6个、6个、8个、9个、5个、8个、3个，与于预期效果基本一致。

下图是对于整个测试集，不同的模型识别准确率比较：

## 7.3 性能测试

性能测试部分主要测试的是，对于同一个样本集，采用不同方式构造情绪分类模型并进行识别整个过程的耗时情况。训练集是一个50046的数据集，测试集大小为100。

下图是构建不同情绪分类模型的耗时情况：

# 第八章 总结与展望

本文针对智能手机日渐普及以及人们越来越关注自己的情绪的情况，通过收集Android用户数据信息，采用机器学习常见分类器如K近邻分类器、SVC分类器、朴素贝叶斯分类器构建了情绪分析模型，实现了对Android用户的情绪识别，来帮助人们随时随地了解自己的情绪状态。

本文提出了一种基于机器学习的安卓移动用户情绪分析方法，对于数据的收集过程、处理过程和分析模型的构建都进行了详细的分析和研究，并进行了准确性测试和性能测试。在数据收集过程，针对传感器的选取、收集的数据种类等方面进行了一定的分析。数据收集完成之后，对数据进行了详细的预处理，包括对异常值、缺失值产生原因的分析以及相关的处理，并对不同的原始数据集进行了集成。将数据进行了基本的整理之后，对数据进行了详细的特征提取和特征选择。此过程中，对属性进行了直观观察，进行了相关性的分析，加速度的合成，对特征进行了选择。考虑到一些数据可能表达的数据关系不明显，我们又将数据进行了特征提取操作，为使数据满足后面计算需要，又对数据进行了归一化操作。由于新样本特征集维度比较高，为避免维度灾难，我们又对数据进行了PCA降维。然后，利用机器学习算法中三种常见分类器，通过调整参数使之适合我们的样本集，然后训练出了三种情绪分析模型。最后，对这三种情绪分析模型进行了准确性测试和性能测试，验证的结果表明模型对于情绪的识别效果良好，三种模型的识别准确率均超过60%，尤其是采用“rbf”核方法的SVC分类模型准确率最高，达到了72%。