**数据分析与处理部分实践作业 4#**

# PCA降维算法实现

## PCA降维算法介绍

### PCA模型

#### 1.1.1.1 PCA基本概念

主成分分析也称主分量分析，旨在利用降维的思想，把多指标转化为少数几个综合指标（即主成分），其中每个主成分都能够反映原始变量的大部分信息，且所含信息互不重复。这种方法在引进多方面变量的同时将复杂因素归结为几个主成分，使问题简单化，同时得到的结果更加科学有效的数据信息。在实际问题研究中，为了全面、系统地分析问题，我们必须考虑众多影响因素。这些涉及的因素一般称为指标，在多元统计分析中也称为变量。因为每个变量都在不同程度上反映了所研究问题的某些信息，并且指标之间彼此有一定的相关性，因而所得的统计数据反映的信息在一定程度上有重叠。主要方法有特征值分解，SVD，NMF等。

#### 1.1.1.2 PCA基本原理

PCA的基本原理就是将一个矩阵中的样本数据投影到一个新的空间中去。对于一个矩阵来说，将其对角化即产生特征根及特征向量的过程，也是将其在标准正交基上投影的过程，而特征值对应的即为该特征向量方向上的投影长度，因此该方向上携带的原有数据的信息越多。

#### 1.1.1.3 PCA求解方法

求得一个k维特征的投影矩阵，这个投影矩阵可以将特征从高维降到低维。投影矩阵也可以叫做变换矩阵。新的低维特征必须每个维都正交，特征向量都是正交的。通过求样本矩阵的协方差矩阵，然后求出协方差矩阵的特征向量，这些特征向量就可以构成这个投影矩阵了。特征向量的选择取决于协方差矩阵的特征值的大小。

#### 1.1.1.4 PCA基本步骤

1、将原始数据按行排列组成矩阵X；2、对X进行数据标准化，使其均值变为零；3、求X的协方差矩阵C；4、将特征向量按特征值由大到小排列，取前k个按行组成矩阵P；5、通过计算Y = PX，得到降维后数据Y；6、用下式计算每个特征根的贡献率Vi;Vi=xi/(x1+x2+........)；7、根据特征根及其特征向量解释主成分物理意义。

#### 1.1.1.5 PCA算法描述

|  |
| --- |
| **Algorithm 1 PCA算法** |
| **输入: 样本集;**  **低维空间维数**  **过程：**   1. **对所有样本进行中心化：;** 2. **计算样本的协方差矩阵;** 3. **对协方差矩阵做特征值分解；** 4. **取最大的个特征值所对应的特征向量**   **输出：投影矩阵** | |

PCA仅需要保留W与样本的均值向量即可通过简单的向量减法和矩阵-向量乘法将新样本投影制低维空间中。

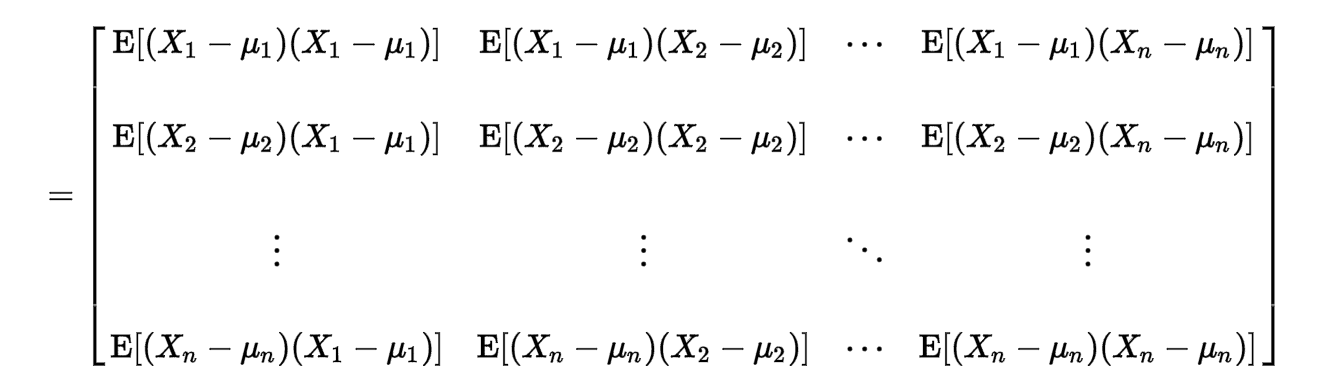
### 1.1.2 协方差矩阵

在统计学与概率论中，协方差矩阵的每个元素是各个向量元素之间的协方差。是从标量随机变量到高维度随机向量的自然推广。

假设{\displaystyle X}X是以n{\displaystyle n}NN个随机变数（其中的每个随机变数是也是一个向量，当然是一个列向量）组成的行向量：

并且{\displaystyle \mu \_{i}}是其第 *i* 个列向量{\displaystyle X\_{i}}中所有元素的平均值，即 {\displaystyle \mu \_{i}=\mathrm {E} (X\_{i})}。协方差矩阵的第i,j项（第i,j项是一个协方差）被定义为如下形式：

而协方差矩阵定义为：

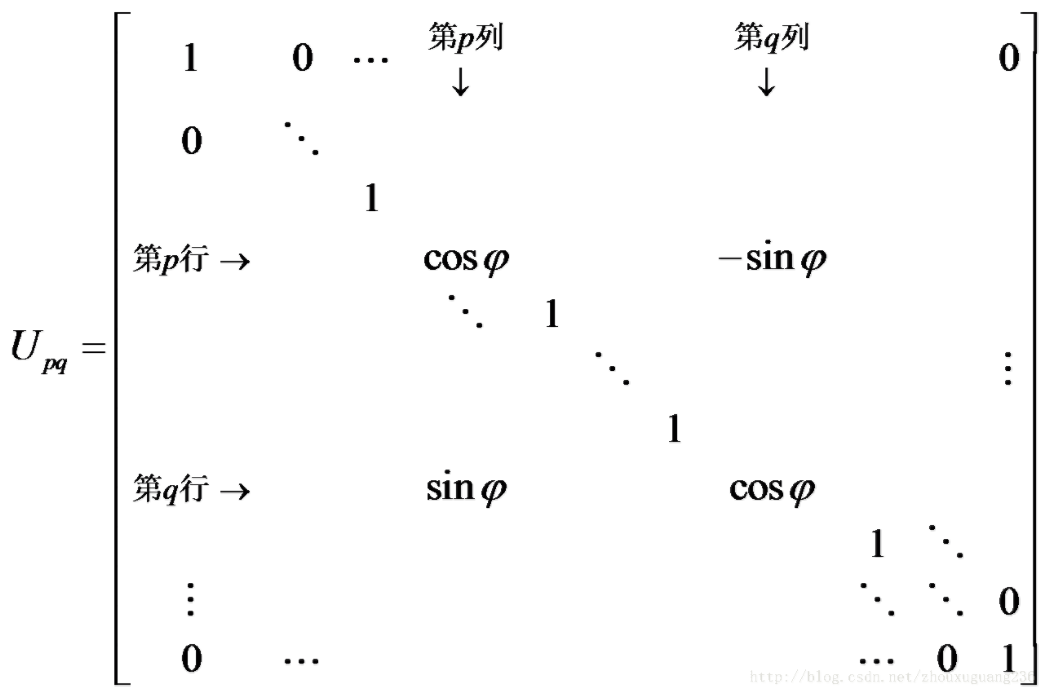


矩阵中的第(i,j)个元素是与的协方差。

### 1.1.3 雅克比算法求特征值和特征向量

算法参考网址：http://blog.csdn.net/zhouxuguang236/article/details/40212143

Jacobi 方法用平面旋转对矩阵A做相似变换，化A 为对角阵，进而求出特征值与特征向量。既然用到了旋转，这里就介绍一下旋转矩阵。对于 p ≠ q,下面定义的 n 阶矩阵Upq 是平面旋转矩阵。



对A做下面的变换：**A1= Upq.T\*A\*Upq**

这里需要求下Upq中的角度的弧度值。由下面的公式求解：

归纳可以得到雅可比迭代法求解矩阵特征值和特征向量的具体步骤如下：

(1) 初始化特征向量为对角阵V，即主对角线的元素都是1.其他元素为0。

(2) 在A的非主对角线元素中，找到绝对值最大元素 。

(3) 用式(3.14)计算**tan2j**，求 **cosj**, **sinj** 及矩阵**Upq** .

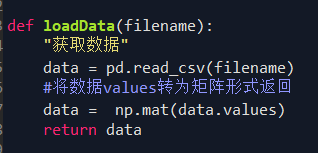
(4) 用公式(1)-(4)求A1；用当前特征向量矩阵V乘以矩阵Upq得到当前的特征向量V。

(5) 若当前迭代前的矩阵A的非主对角线元素中最大值小于给定的阈值e时，停止计算；否则, 令A = A1 , 重复执行(2) ~ (5)。 停止计算时，得到特征值 li≈(A1) ij ，i,j= 1,2,…,n.以及特征向量V。

## PCA降维算法实现

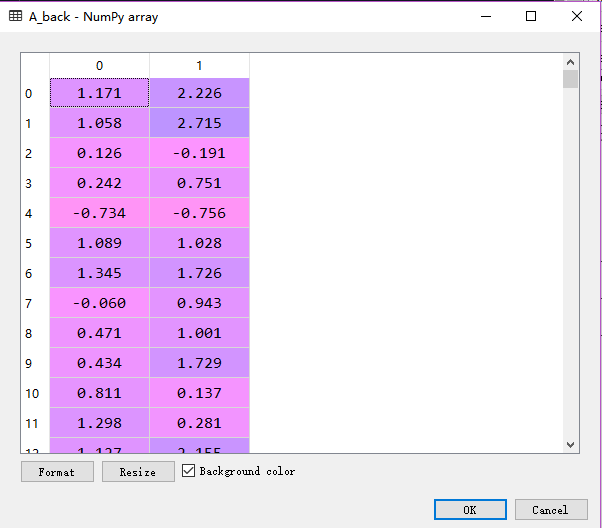
### 1.2.1 传入数据集

1、构造函数：



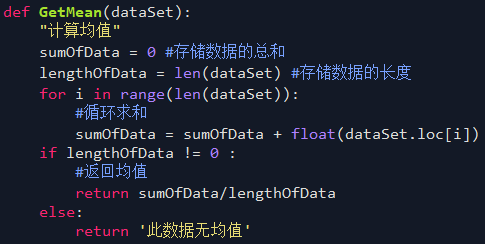
2、过程：用pandas读取csv文件，将values值以矩阵形式返回。

3、传入结果：



### 1.2.2 计算均值

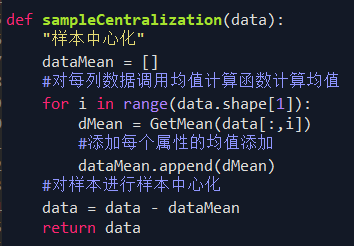
1、构造计算均值的函数：



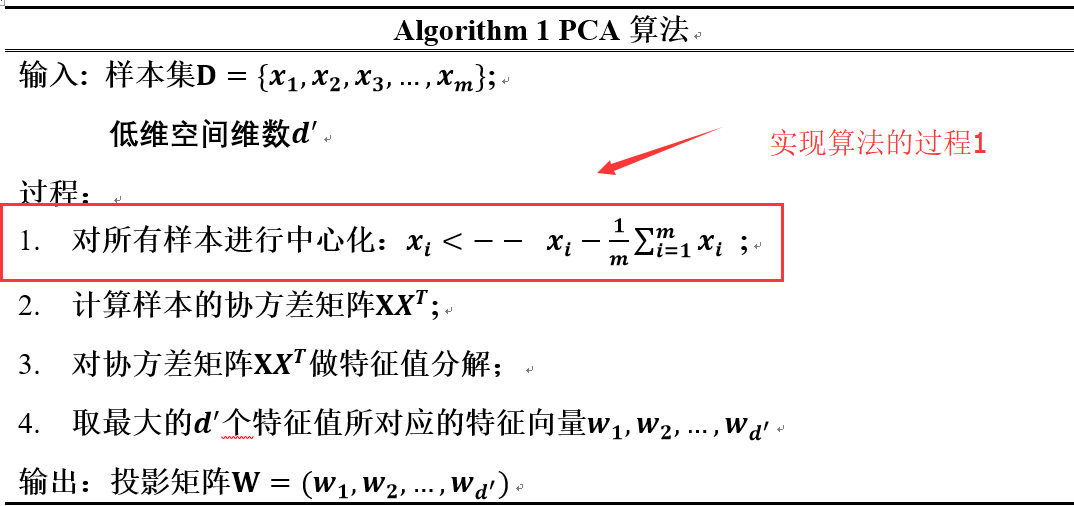
2、步骤为：（1）传入要计算列的参数；（2）设置变量存储该列数据的总和，以及设置变量存储数据的长度；（3）长度调用len()函数；（4）数据总和循环整个长度（也可以设置循环当有值存在时，长度加1，同时累加数据的和）；（5）数据总和除以数据长度得到均值（需满足长度不为0）。

### 1.2.3 样本中心化

1、构造函数：

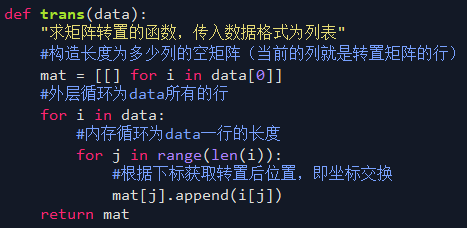


2、过程：（1）首先传入data数据；（2）循环数据所有列，对每列求均值；（3）然后对所有样本数据进行中心化，即减去对应的均值。实现算法的过程1：



### 1.2.4 矩阵的转置

1、构造计算矩阵转置的函数：

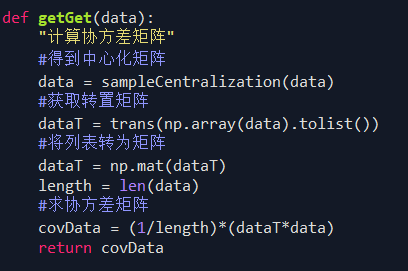


2、步骤为：（1）首先建立一个原矩阵长度为多少列的矩阵，因为转置后新矩阵的行数就是原矩阵的列数；（2）循环所有行，内层循环为每一行的长度，新矩阵每个元素的转置结果为坐标交换的原矩阵的值。

3、这里求矩阵的转置是为求数据的协方差矩阵做准备。

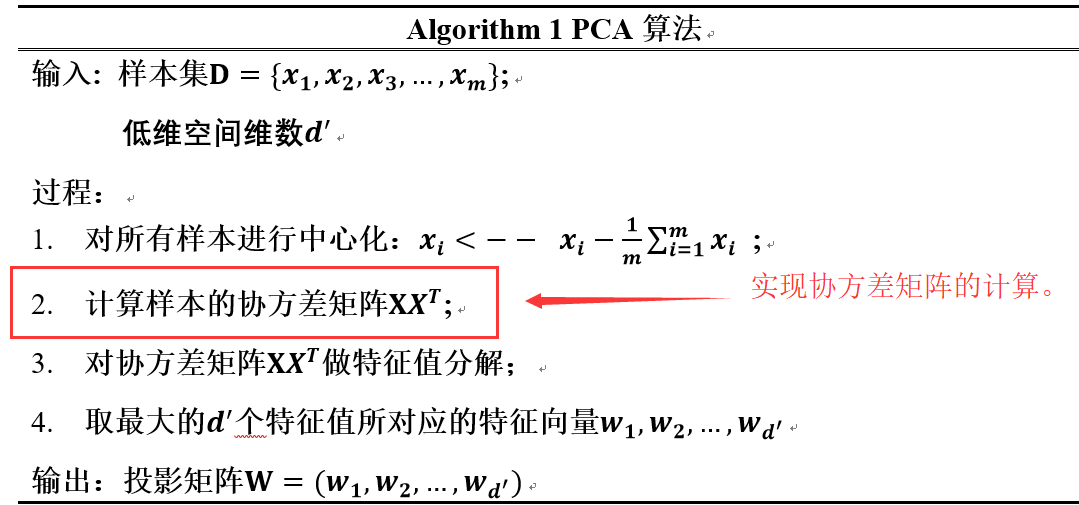
### 1.2.5 计算协方差矩阵

1、构造函数



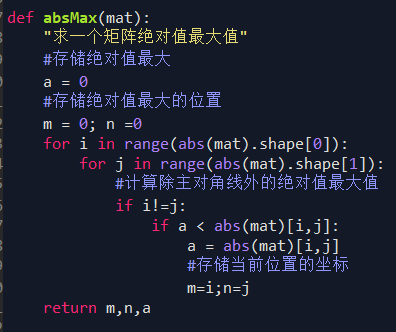
2、过程：（1）首先调用中心化数据的函数将数据进行中心化；（2）将矩阵调用转置函数进行转置；（3）根据公式二者乘积为协方差矩阵。

3、这里就实现协方差矩阵的计算，即算法步骤2：



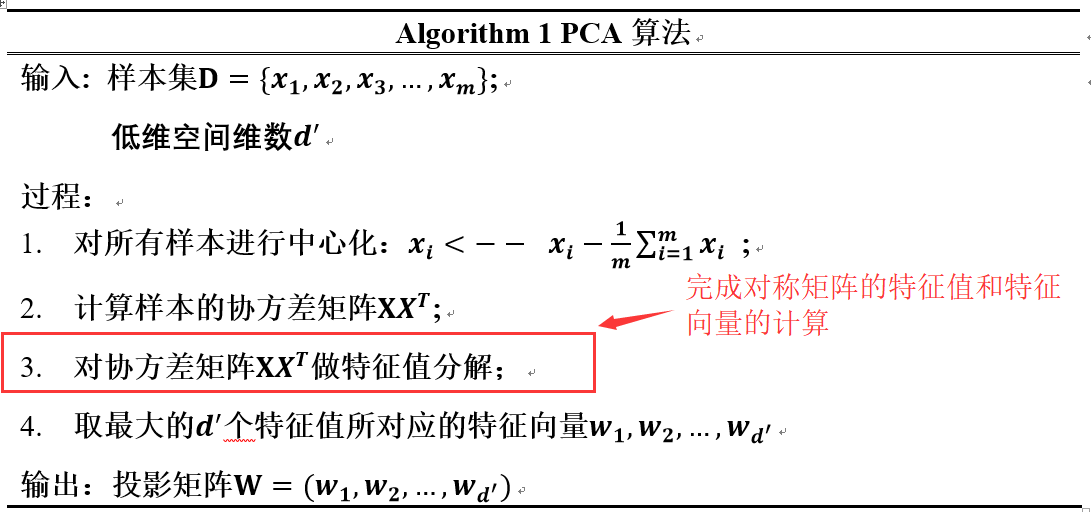
### 1.2.6 计算矩阵除主对角线的绝对值最大值

1、构造函数：



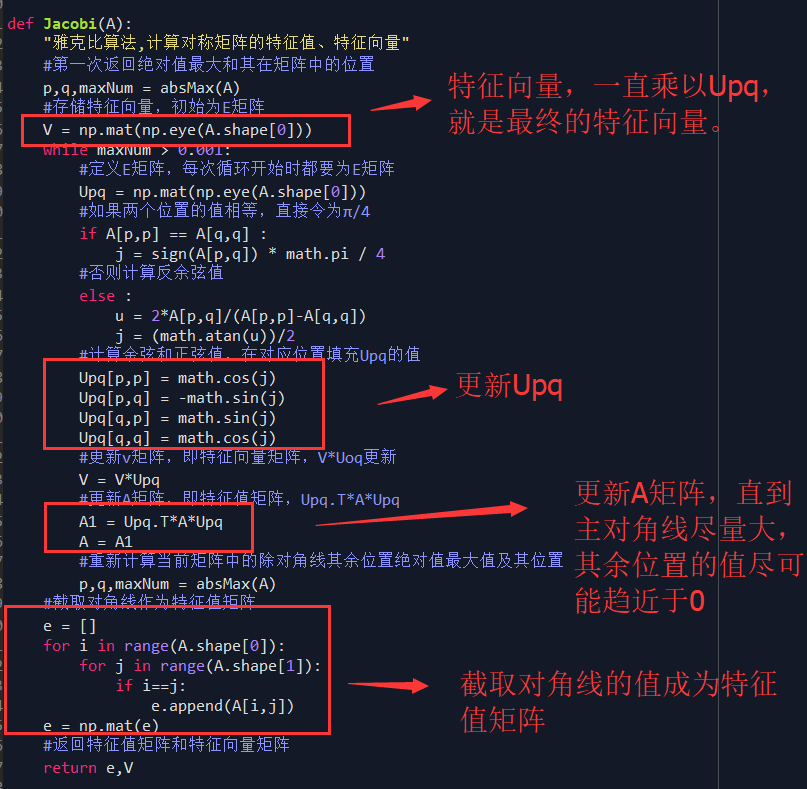
2、步骤：（1）首先传入需要计算的矩阵，设置存储值的变量和该值的坐标位置的变量；（2）遍历整个矩阵，找非对角线上绝对值的最大值，并保存当前位置坐标；（3）返回最大值和最大值的坐标。

3、完成算法过程3的计算：



### 1.2.7 雅克比算法

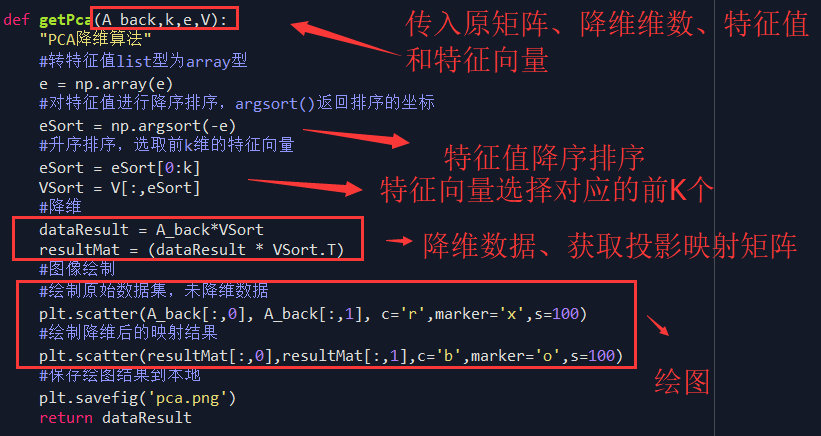
1、构造函数：



2、步骤：（1）第一次调用求非主对角线求绝对值最大值的函数，获得第一次的最大值的位置和绝对值大小，进入循环；（2）根据返回的值求得角度的弧度值，根据对应坐标更新Upq矩阵，然后Upq.T\*A\*Upq就是更新后的A矩阵，设定阈值：最大值小于一个极小值时，跳出循环。而最终的A的主对角线就是特征值，至于特征向量就是E矩阵一直乘以迭代过程中的Upq；（3）截止迭代后，返回特征值和特征向量。

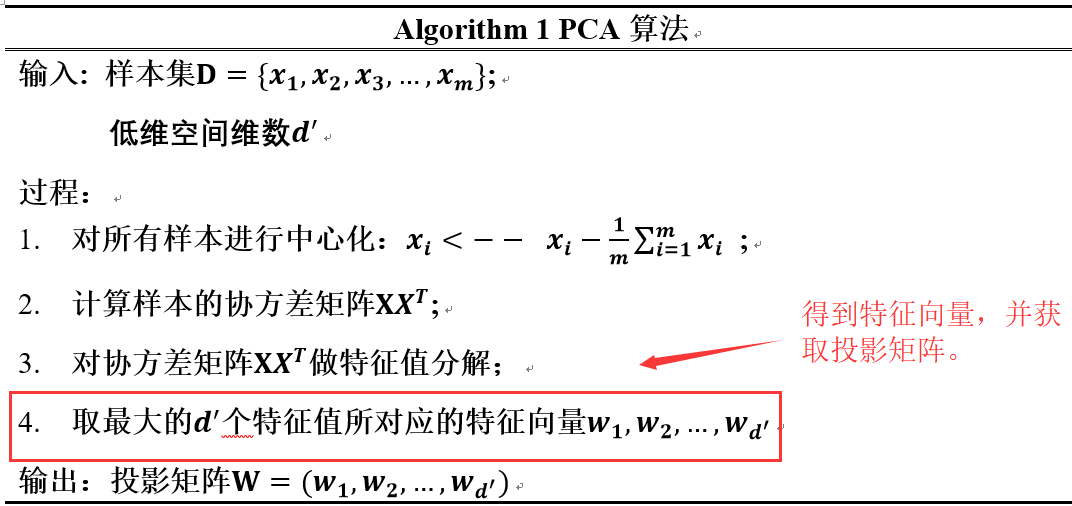
### 1.2.8 PCA算法

1、构造函数：



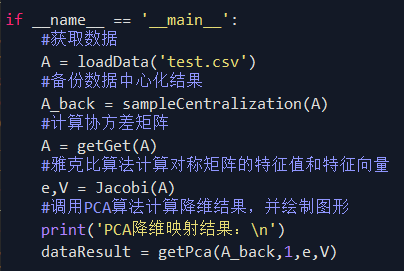
2、步骤：（1）传入原矩阵、特征维数、特征值、特征向量，根据前面写的函数计算这些传入的参数；（2）对特征值大小进行排序，根据排序结果选择对应的前K个特征向量；(3)根据截取的特征向量计算降维后的特征数据值和投影映射矩阵；（4）根据结果绘制出图像，降维前后数据分别用红色小×和蓝色小圆来表示。

3、完成算法步骤4：



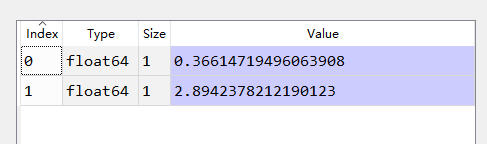
### 1.2.9 PCA算法演示

1、传入数据文件的位置，依次调用函数：获取数据、中心化数据样本、计算协方差矩阵、雅克比算法、PCA算法：

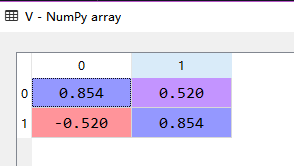


2、得到结果：

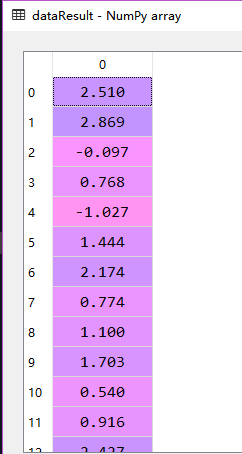
（1）特征值结果：



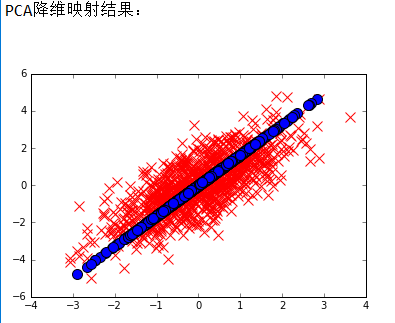
（2）特征向量结果：



（3）特征降维结果：

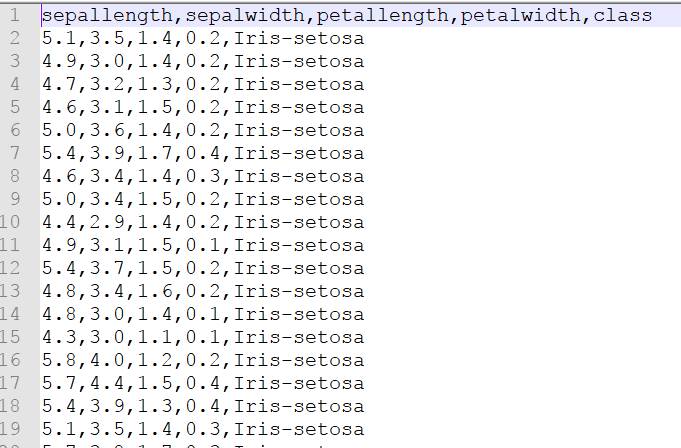


（4）pca降维可视化结果：



### 1.2.10 PCA降维效果评价

1、传入数据集为weka中的iris.csv数据集（本身为arff格式，改为csv格式传入使用），格式为：



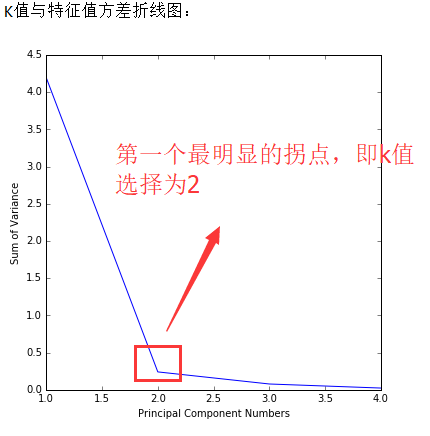
2、代码整体结构为改变，对pca函数做了修改，以及增加了评估函数：

**pca函数：**

（1）函数构造：



（2）绘图结果：



原理和K-means算法中的肘方法确定K值一致，在第一个最明显的拐点处选择K值，显然，这里应该选择降维维数为2。

**评估函数：**

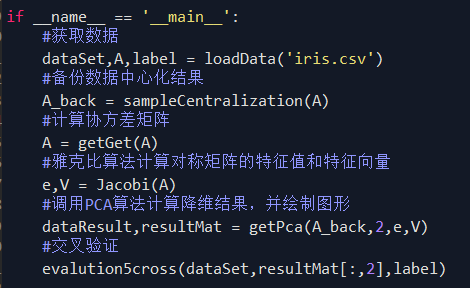
（1）函数构造：

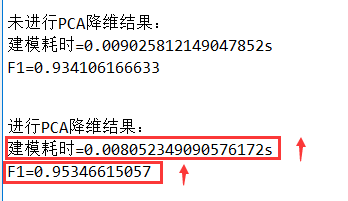


（2）过程：导入sklearn的交叉验证函数，以及贝叶斯分类器；数据传入原始未降维数据，以及降2维后的数据，以及数据样本的标签。分别两个数据集调用贝叶斯分类器，然后用交叉验证计算评均的F1值，查看F1值的提升情况；并导入time包，分别在两组建模前后获取系统当前时间，二者做差为建模时间。

也尝试过随机森林模型，但是随机性太强了，不能很好的说明问题，最后选择了贝叶斯分类器。

（3）调用函数，得到结果：





根据结果可以看出，时间性能以及F1值均有提升，降维效果明显。

# 基于J48算法的特征选择算法实现

## 2.1 J48算法介绍

J48是C4.5决策树学习算法，决策树是一类常用的机器学习方法，基于树的结构来进行决策，是面临决策问题是自然的一种处理机制。决策树的学习目的是为了产生一棵泛化能力强，即处理未见示例能力强的决策树，其生成是一个递归的过程。

决策树最重要的划分选择，J48是利用增益率来进行选择最优属性划分的：

**“信息熵”**是度量样本集合纯度最常用的指标，假定样本集合D中第k类样本所占比例为(k=1,2,…,|y|)，则D的信息熵定义为：

的值越小，则D的纯度越高。根据信息熵公式计算出某个属性a的样本的信息熵，再考虑不同分支结点所包含样本数不同，给其赋予权重，样本数越多的分支结点的影响越大，于是计算出“信息增益”：

信息增益准则对可取数目较多的属性有所偏好，为减少这种偏好带来的不利影响，C4.5使用“增益率”来选择最优划分属性，而不直接使用信息增益。

**“增益率”**的定义为：

,其中

因此，实现J48算法首先需要计算信息熵，然后通过计算属性的增益率来划分最优属性，也需要写划分数据集函数来根据最优属性来进行数据集的划分。

## 2.2 特征选择算法介绍

### 2.2.1子集搜索与评价

给定属性集，我们将属性称为“特征”，“相关属性”：对当前学习任务有用的属性，“无关属性”：对当前学习任务没什么用的属性。从给定的特征集合中选择相关特征子集的过程，称为“特征选择”。

特征选择是一个重要的数据预处理过程，在现实机器学习任务中，获得数据之后通常进行特征选择，然后再训练学习器。原因：1、解决维数灾难问题；2、留下关键因素，数据真相更加明确。但需要保证特征选择过程必须确保不丢失重要特征，否则后续学习过程会因为重要信息缺失而无法获得好的性能。

两个关键的环节：

1、“子集搜索”：给定特征集合，将每个特征看作一个候选子集，对候选单特征子集进行评价。直到第k+1轮时，最优的候选(k+1)特征子集不如上一轮的选定集。这称为“前向搜索”，还可以使用“后向搜索”和“双向搜索”。

2、“子集评价”：对属性子集A，假定根据其取值将数据集D分成了V个子集，每个子集中的样本在A上取值相同，于是计算到属性子集A的信息增益：

其中信息熵定义为：。信息增益越大，意味着特征子集A包含的有助于分类信息越多。

将特征子集搜索机制与子集评价机制相结合即可得到特征选择方法，大致分为三类：过滤式(filter)、包裹式(wrapper)、嵌入式(embedding)。

### 2.2.2过滤式选择

先对数据集进行特征选择，然后训练学习器，特征选择过程与后续学习器无关。先用特征选择过程对初始特征进行“过滤”，再用过滤后的特征来训练模型。

Relif方法是著名的过滤式特征选择方法，设计了“相关统计量”来度量特征的重要性。该统计量是一个向量，其每个分量分别对应一个初始特征，而特征的重要性由子集中每个特征所对应的相关统计量分量之和来决定。可以指定阈值，也可以指定预选取的的特征个数K。Relief的关键是确定相关统计量，通常使用猜中近邻和猜错近邻的结合。对基于不同样本得到的估计结果进行平均，就得到各属性的相关统计量的分量，分量值越大，则对应属性的分类能力就越强。时间开销随采样次数以及原始特征数线性增长，因此是一种运行效率很高的过滤式特征选择算法。

### 2.2.3包裹式选择

过滤式特征选择不考虑后续学习器的不同，包裹式特征选择直接把最终要使用的学习器的性能作为特征子集的评价标准。即为给定学习器选择最有利于其性能、“量身定做”的特征子集。

包裹式特征选择方法直接针对给定学习器进行优化，从最终性能来说，比过滤式特征选择更好，但在特征选择过程中需多次训练学习器，因此包裹式特征选择的计算开销通常比过滤式特征选择大很多。

LVW是典型的包裹式特征选择方法，在拉斯维加斯方法框架下使用随机策略进行子集搜索，以最终分类器的误差为特征子集评价准则。由于特征子集搜索采用了随机策略，而每次特征子集评价都需训练学习器，计算开销很大，因此算法需要设置停止条件控制参数T。

### 2.2.4嵌入式选择

在过滤式和包裹式特征选择方法中，特征选择过程与学习器训练过程有明显的分别。而嵌入式特征选择过程与学习器训练过程融为一体，即学习器训练过程中自动地进行了特征选择。

给定数据集D，考虑最简单的线性回归模型，以平方误差为损失函数，则优化目标为：

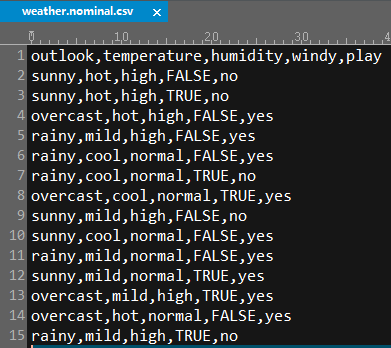
当将范数替换为，令P=1，即采用范数，则有LASSO：

二者均可降低过拟合风险，但是范数更易获得稀疏解，即求得的w会有更少的非零分量。

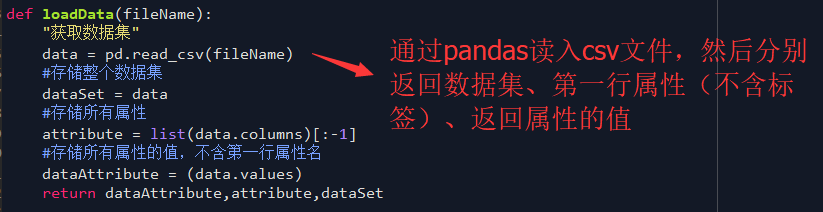
## J48特征选择算法实现

### 2.3.1 读入数据

1、选择weka安装目录下data文件夹中的weather.nominal.arff数据集，转为weather.nominal.csv格式待使用：



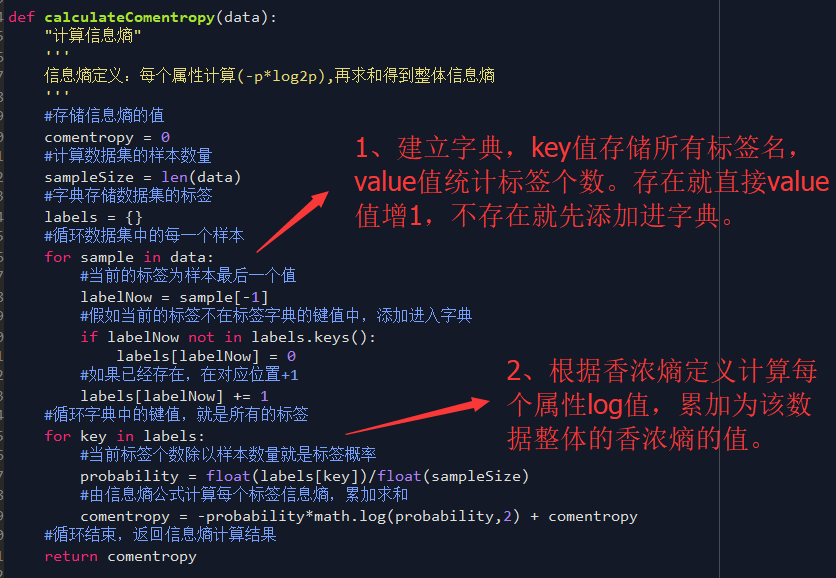
2、 函数读入.csv格式的文件：



### 2.3.2 计算香浓熵

1、数据集传入weather数据集。

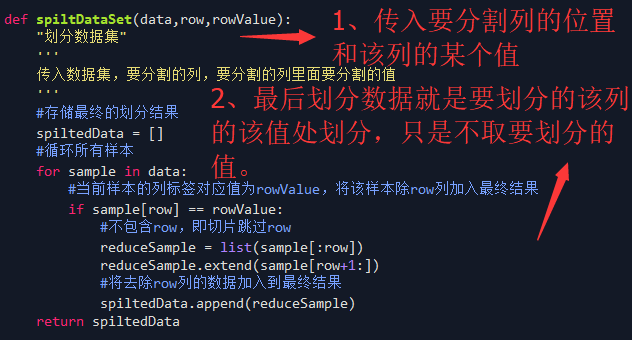
2、构造计算香浓熵函数：



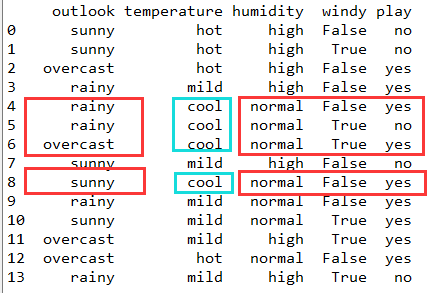
### 2.3.3 划分数据集

1、在后面属性信息增益率计算需要先写出划分数据集的函数，传入weather数据集。

2、构造函数：



3、调用函数，例如传入数据集，按数据集的**第1列的cool值**划分，**预期划分结果就是红色框定的部分**：



### 2.3.4 计算信息增益率

1、传入weather数据集

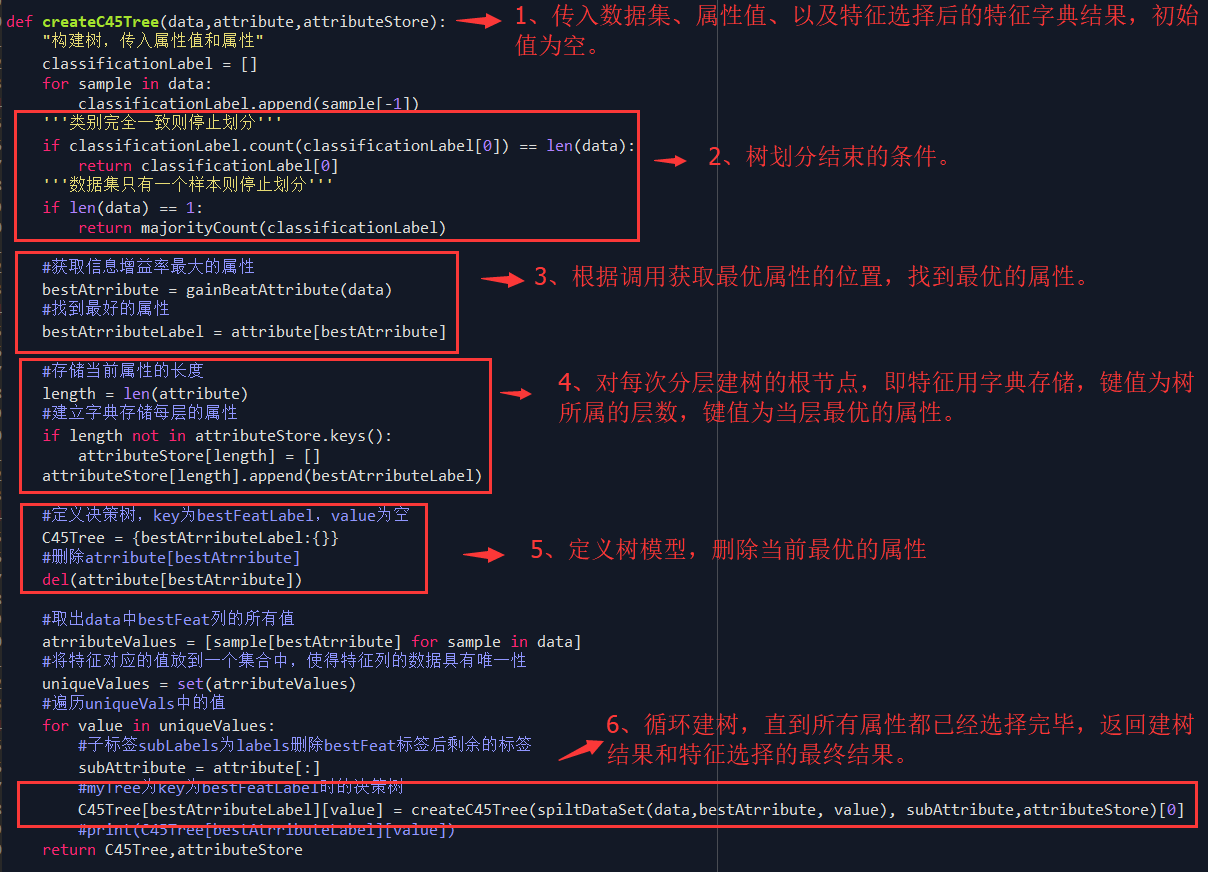
2、构建计算信息增益率的函数：



### 2.3.5 建立特征选择树模型

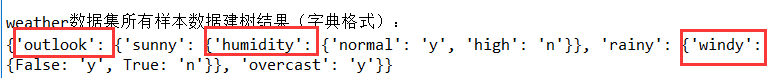
1、传入weather数据集

2、构建建树函数：



相当于对当前最优属性列中，按每个值进行遍历，建树的字典分别等于当前属性值递归调用建树的子树。

3、调用函数，并将建立的树模型按字典的形式打印出来：

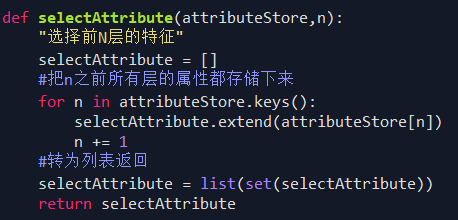


而对于建树结果中，第四层（顶层）的属性：outlook，以及第三层的属性：humidity和windy分别存储在字典中，为后期做特征选择评价做准备。



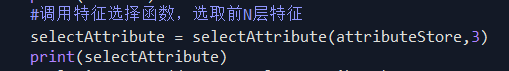
### 2.3.6 对树模型选择前N层的属性

1、函数构造：

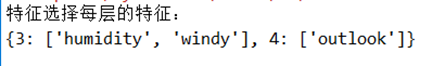


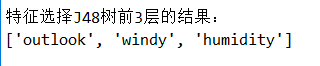
2、过程：传入之前建树时存储下来的每层对应的属性的字典，以及需要保存的层数；如果层数在字典的键值中，就将对应的键值加入进列表中；n值不断加1；重复此过程，直到层数达到最大值，即到达顶层；返回列表，即存储属性的列表。

3、调用函数返回结果：



这里将会将3层以上的特征结果，转为列表返回：





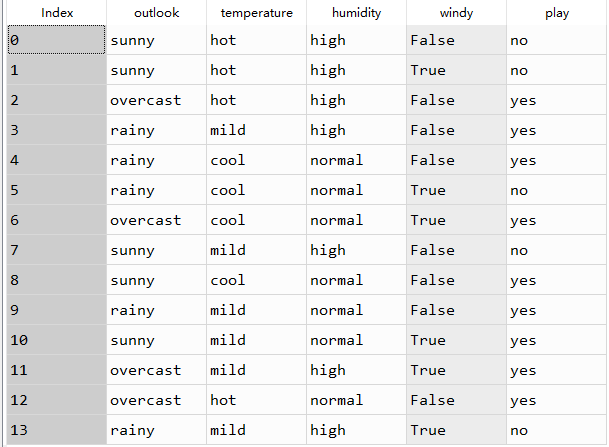
### 2.3.7 对特征选择的评估

1、构造函数：

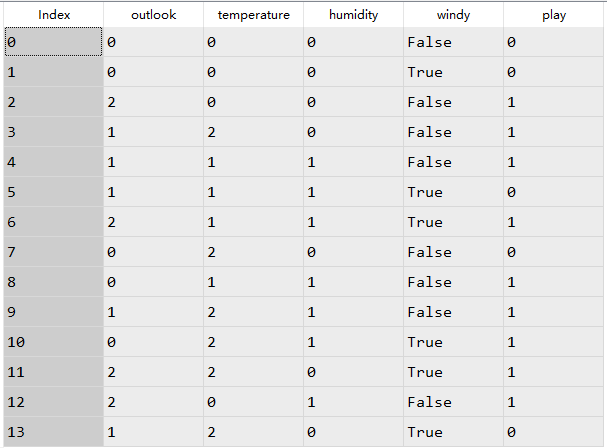


2、过程：（1）传入原始数据集，以及特征选择后的属性列表；（2）由于sklearn交叉验证只能对数值型属性值进行计算，这里先对天气数据集做了处理：

原始数据：

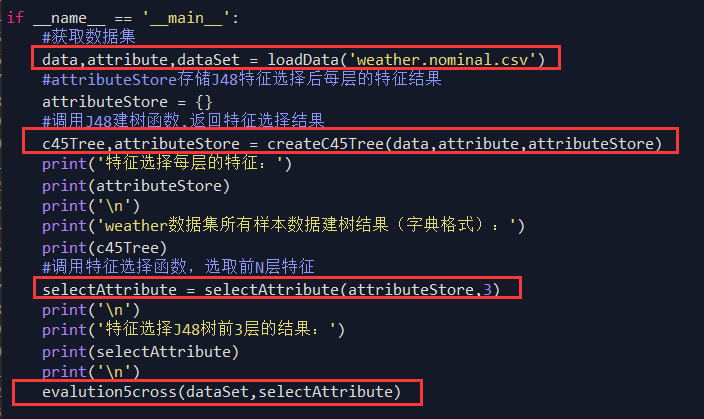


数值化数据：

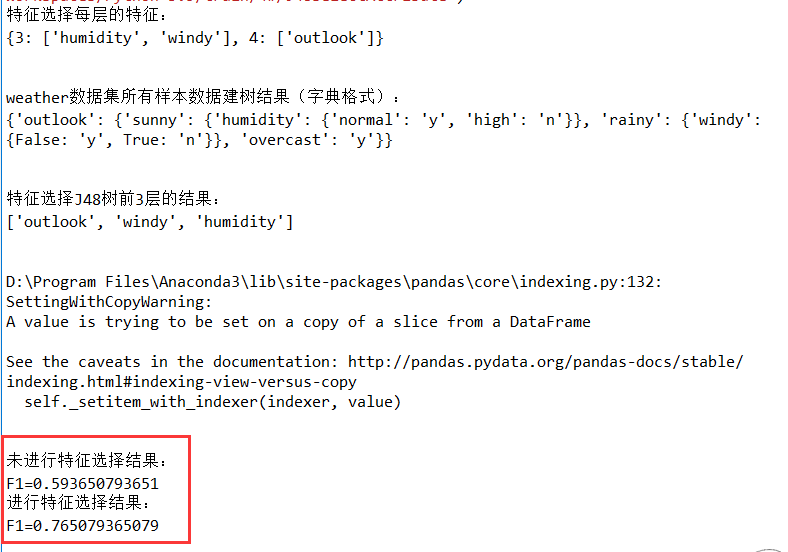


（3）由于数据样本太少，这里调用**随机森林（或者gbdt，随机性太大）**观察特征选择的效果并不好，于是选择调用了sklearn里面的贝叶斯分类器；（4）最后调用交叉验证，返回F1的均值作为性能评价。

3、调用函数，最终结果：



返回结果：



由结果可以看出，F1值从0.593增加到0.765，表明基于J48的特征选择是有效可行的，并且性能评估良好。

# 心得体会

本次作业实现了PCA和基于J48的特征选择实现，首先对两个算法有了更深刻的理解。

在实现PCA的时候，计算协方差实现很容易，但是实现特征值和特征向量很复杂，最终利用一个博客分享的雅克比算法近似求对称阵的特征值和特征向量，实现了整个算法，整个过程调bug最痛苦。对降维映射结果进行了可视化，也调用分类器进行对比降维前后的时间和F1值，对于评测函数调用了sklearn库中的交叉验证函数。

实现基于J48的特征选择相对来说容易很说，因为算法是使用的之前的，只是实现了存储N层之前的特征，最后调用了sklearn中的分类器和交叉验证来评估了特征选择的效果。