**一、机器学习伊始** 机器学习主要是基于目前互联网发展产生的大量数据而进行的价值分析、挖掘，在互联网方面，以零散的用户数据，对用户行为习惯进行分析，从而分析总结出用户的喜好和行为预测。 在目前互联网公司大多都存在机器学习的身影，比如： 今日头条的用户会发现在推荐栏里出现的推荐的内容和用户平时浏览的自媒体内容息息相关，用户在平时浏览中点击了哪些内容，会产生相关的反馈信息：用户点击了被推荐的哪类内容，没有点击被推荐的哪类内容。根据用户习惯点击的内容对内容进行个体化标签化分类，从而根据机器学习算法推导此用户的浏览内容偏向，根据分类偏向分析结果进而在新的内容被打上内容分类标签被上传至头条后给被标签化的用户推送相应的内容信息，从而使得用户的点击浏览率增高，相信大家也体验过在淘宝上搜索某件商品后，进入头条也会被推荐相应的商品，这也是目前互联网信息共享手段之一，用最近的热播剧<<猎头>>里的理解就是：这是一个信息资源共享的时代，多个服务方获得用户的信息，可以在多方面提供给用户更好的体验 滴滴出行是国内目前一站式的打车平台，随着平台和订单的扩大化，乘客和司机之间产生的问题也不计其数，相信大家使用滴滴出行时候也会出现司机取消订单/乘客取消订单/司机绕路等情况的发生，这也是机器学习可以解决的问题，收集司乘的数据，对相应的数据进行建模分析，在司乘出现矛盾时及时对相应收集的GPS、司乘对App的操作来进行相应的责任判定，使得大量的订单出现的复杂问题由人工化处理转向智能化处理 百度搜索引擎下推荐的新闻也与机器学习有很大的关系 阿里在双十一传出AI“鲁班”的8000张/s的海报制作，三大技术原理：风格学习、行动器、评估网络，对大量设计素材的设计数据进行结构化标注，在经过一系列的神经网络学习，输出空间+视觉的设计框架[出自阿里技术公众号] 由此可见，机器学习对目前互联网及生活的应用之广泛，很多时候我们一直都在这个数据圈内，既然在这个数据圈内，就需要更加了解这个数据圈 在大二到大三时，目前大四，之前一直在做Java方面的学习开发，但是对机器学习领域还是小白一枚，若有错误或者不适当地方，还请见谅

1. 根据周志华教授的<<机器学习>>--决策树部分列出的点

决策树是一类常见的机器学习方法，以二分类任务为例，我们希望从给定训练数据集学得一个模型用以对新示例进行分类，这个把样本分类的任务可看作对”当前样本属于正类吗？“这个问题的”决策“或”判决“过程 决策时牵扯到属性的划分： 信息熵：主要指信息的混乱程度 信息增益：Gain(D,a)为以某一属性对数据集进行划分后，整体数据集混乱程度的降低量，整个决策树的构建过程就是在不断减少信息混乱程度的过程 增益率：根据信息增益的公式了解到，信息增益在划分属性时候对可取值数目较多的属性有所偏好(由于不太方便，这里就不列出公式了)，使用Gain(D,a)/IV(a)对信息增益带来的偏好度进行减弱，属性a的可能取值数目越多，IV(a)的取值通常会越大，但是一般不以增益率做直接的划分原则值，C4.5算法使用信息增益+增益率一起来决定划分的属性：先从候选划分属性中找出信息增益高于平均水平的属性，再从中选择增益率最高的，尽力达到对属性划分过程中的平衡 基尼指数：基尼指数Gini(D)反映了从数据集D中随机抽取两个样本，其类别标记不一致的概率，因此Gini(D)值越小，则数据集D的纯度越高，是CART决策树的划分属性方法 根据以上的比较经典的划分属性方式，我们可以在练习（有限少量）时先进行一下计算，当然觉得手工计算麻烦可以写个程序对数据进行计算，这个可以省很多事的(此时庆幸自己还是会写点程序的^\_^) 决策树的剪枝处理： 过拟合：训练数据训练出的模型将训练数据“太当回事”了，也就是说，训练出的模型将训练数据集自身的一些特点当作当作所有数据的特点了，比如：假设我们小时候对白马王子的认知：骑白马的一定是王子，实际上还有唐僧不是吗？过拟合，差不多是这个理儿

欠拟合：一般我们进行监督学习的时候会将数据集以留出法，n次k折交叉验证法、自助法等划分为训练集和测试集，如果把训练集太当回事，那是“过拟合”，如果把训练数据都不当回事（就是对训练数据的一般性质都没学习好），那就是“欠拟合”了，很明显，“欠拟合”，一般我们需要解决大多是“过拟合”问题，对于决策树来说，“过拟合”可以采用“剪枝”的方法进行降低

剪枝：

预剪枝：将划分属性后和划分属性前进行数据集验证，如果划分某条属性后反而降低了整个验证集精度，或者对整体的验证集精度没有提升，那么我们就不展开对此属性的划分，这样减少了训练的资源成本同时也不影响验证集精度，但是没有展开的属性可能对后续的训练会有较大的提升，在此处就把它扼杀在摇篮里，比较可能导致“欠拟合”的风险

后剪枝：当我们将整棵决策树训练出来以后，再对整棵决策树进行剪枝处理，从底至上，当我们将某两类的领衔节点分类结果直接替换为其叶子节点的分类占比较高的节点类别，再进行验证精度测试，如果验证精度提高，选择剪去此条属性划分，并将分类占比高的类别直接替换掉其领衔节点，这样通常后剪枝处理比预剪枝处理使分支保留更多，欠拟合风险小，但是因为其是先训练好决策树再进行的对非叶子节点的剪枝操作，会对资源成本有较高的消耗

连续值离散化处理：

通常以决策树而进行的分类操作一般是以离散属性(有限个结果取值)进行的分类操作，但是实际操作中我们常常面临的是需要对连续属性进行划分属性操作，这给决策分类思想带来很大的麻烦，需要用到的技术就是“连续属性离散化”，其中最简单的策略是二分法对连续属性进行处理，也是C4.5决策树算法采用的机制：我们认为在连续属性的相邻值[a,a+1)之中取任意值对其划分结果相同，那么我们将数据集进行排序后，取连续两个属性的中间值作为候选划分点集合，然后对此集合进行离散式的属性划分，便可以达到连续属性离散化后对离散属性的划分方式进行划分

缺失值处理：

一般学习时不太会遇到缺失值的情况，但是在实际应用过程中经常会出现某样本上缺失属性值的情况，比如：数据传输失败、网络原因、设备原因等都会造成属性上数据值的却是

对于属性值的缺失会产生两个比较显著的问题：

1>对于划分属性时，此属性的却是对划分原则的影响，无论使用信息增益、信息率、基尼指数对属性进行划分，缺失的属性值对其影响

2>当确定好以某属性对样本集进行划分训练时，遇见欲划分样本缺失此项属性值时该怎么解决

对于问题1:在缺失值不多的情况下，我们可以选择将数据集D中此项属性值未缺失的样本过滤出来为D`，个根据D`上的数据对属性进行划分，同时也需要对每个样本x赋予一定的权重

对于问题2：当对每个样本划分好权重后，欲划分样本此时可以根据此属性在不同类别的占比情况表示为r，让此样本同时划分入所有的子节点，同时将样本本身权值\*r进行重新复制，直观来看，就是让同一个样本以不同概率划入到不同的子节点去，以此解决问题2

目前针对<<机器学习>>的书上<决策树>章节个人认为的重点总结完毕，有误的地方欢迎指出

纸上得来终觉浅，绝知此事要躬行

必然要选择一门语言进行尝试了

在机器学习领域，目前比较流行的语言有R语言、Python、Java、C/C++、Lisp，因为大三下半学期进行了课程python部分的学习所以决定要从python入手

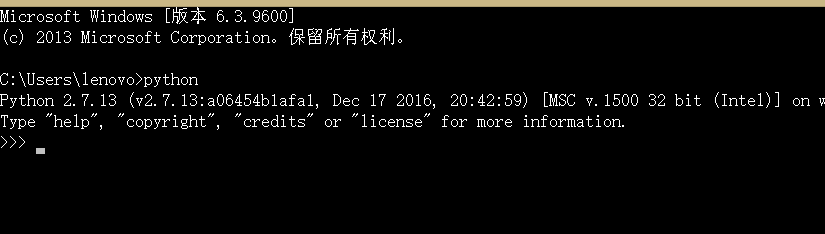
三．Python及Sklearn配置

1.关于Python的安装和学习网上有很多教程，在此不再赘述，菜鸟教程：[http://www.runoob.com/python/python-install.html](http://www.runoob.com/python/python-install.html有关于python的安装和基础的语法和使用教程还是很不错的)

[有关于python的安装和基础的语法和使用教程还是很不错的](http://www.runoob.com/python/python-install.html有关于python的安装和基础的语法和使用教程还是很不错的)，看教程时发现有一点Python的垃圾回收机制是用“引用计数器”方式来进行判断的，我们都知道这样很容易出现“循环引用指向”导致对象不得被释放的情况，为此python引入了“循环引用垃圾回收器”，这点个人觉得很新奇，因为以前做Java时候，学习JVM的时候在<<深入理解Java虚拟机>>这本书中知道Java为避免循环引用问题使用GCRoot来进行可达性分析对对象是否有用进行判定，Python却使用的引用计数器+循环垃圾回收器来进行判定

还是感觉挺新奇的，作为小白有点大惊小怪吧，下来研究一下

安装好python后当你再dos命令窗口输入：python时出现以下提示代表安装成功



2.IDE的选择

建议下载PyCharm这种Python IDE，用过IntelliJ IDEA的童鞋大多都比较喜欢IntelliJ IDEA，同由JetBrains打造的，所以风格和IntelliJ IDEA基本相似，操作方式雷同，个人感觉很亲切

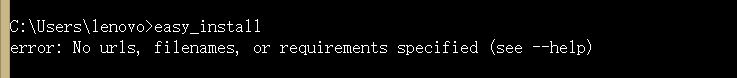
1. 安装setuptools

setuptools是 Python Enterprise Application Kit（PEAK）的一个副项目，它 是一组Python的 distutilsde工具的增强工具（适用于 Python 2.3.5 以上的版本，64 位平台则适用于 Python 2.4 以上的版本），可以让程序员更方便的创建和发布 Python 包，特别是那些对其它包具有依赖性的状况。

当需要安装第三方python包时，可能会使用easy\_install命令。easy\_install是由PEAK(Python Enterprise Application Kit)开发的setuptools包里带的一个命令，所以使用easy\_install实际上是在调用setuptools来完成安装模块的工作。---来自百度百科

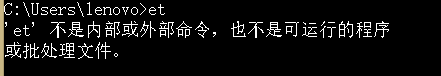
附上安装教程：<http://blog.csdn.net/liuci3234/article/details/48750785>

当在dos命令框输入：easy\_install时出现



代表安装成功，出现此error是因为后面需要跟参数来执行命令

如果出现此类情况：



请检查环境变量将：盘符\python\Scripts添加到环境变量里即可

1. 安装pip

python pip是一个安装和管理 Python 包的工具 ,是easy\_install的替代品。

可以使用easy\_install安装pip：easy\_install pip进行安装

本小白遇见easy\_install安装numpy、scipy和sklearn时出现异常，为了避免麻烦就选择了pip进行安装，顺利安装成功

安装命令：pip install param

是否安装这个可以自行根据情况决定

1. 安装numpy

NumPy系统是Python的一种开源的数值计算扩展。这种工具可用来存储和处理大型矩阵，比Python自身的嵌套列表（nested list structure)结构要高效的多（该结构也可以用来表示矩阵（matrix））。据说NumPy将Python相当于变成一种免费的更强大的MatLab系统。

pip install numpy

1. 安装scipy

SciPy是一款方便、易于使用、专为科学和工程设计的Python工具包.它包括统计,优化,整合,线性代数模块,傅里叶变换,信号和图像处理,常微分方程求解器等等.

安装同上

为什么要安装这两个玩意，上网查询后才知道依赖关系

sklearn 依赖于 numpy 和 scipy

scipy 依赖于 numpy + mkl

唉，纯小白真悲哀

1. 下载安装sklearn

sklearn 是一个 Python 的 科学计算库，提供了数种聚类算法可供选择

numpy、scipy 是 Python 的科学运算库，matplotlib 是图形库，用于绘图

所以matplotlib也安装一下吧

注：pip安装sklearn时候建议使用

pip install -U scikit-learn

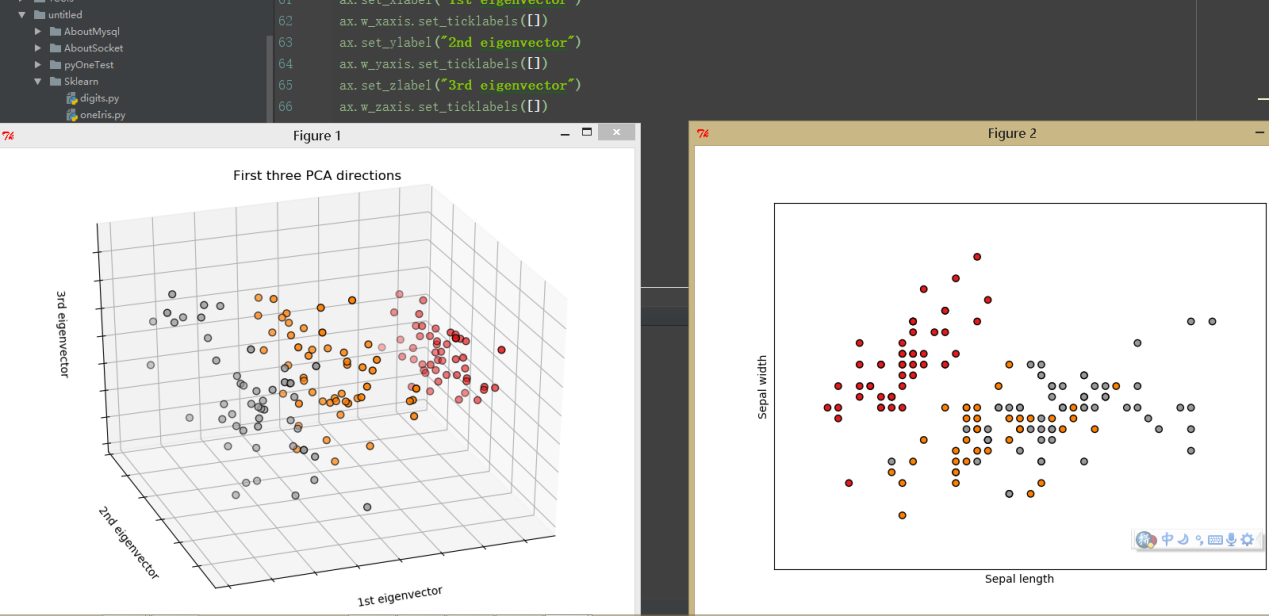
否则可能出问题

那么安装完后我们怎么检测是否安装成功呢？

打开PyCharm

从这个网址 :http://scikit-learn.org/stable/auto\_examples/datasets/plot\_iris\_dataset.html

下载一份源码，放到PyCharm里运行，如果出现下图所示，那么恭喜你至少到目前为止一切都是很顺利的(先不要在意它在写什么)



1. 来一套简单的代码练练手吧！！！

鸢尾花数据集：

鸢尾花数据集是原则20世纪30年代的经典数据集。它是用统计进行分类的鼻祖。

很愉快的这是sklearn自带的玩具式数据测试学习范本:

注：一部分代码参考网上代码

#!/usr/bin/python  
  
# \_\*\_ encoding=utf-8  
# 加载所需要的模块  
**from** \_\_future\_\_ **import** print\_function  
#数据范本存在这里  
**from** sklearn **import** datasets  
**from** sklearn.cross\_validation **import** train\_test\_split  
**from** sklearn.neighbors **import** KNeighborsClassifier  
  
**'''  
一共150组数据  
'''**# 将数据加载，属性存储在x中，类别标签存储在y中  
# 类别标签一共有三种0，1，2；属性每个样本有四种  
iris **=** datasets.load\_iris**()**#print(iris.values())  
iris\_x **=** iris.data  
iris\_y **=** iris.target  
**'''  
print("data\n",iris\_x)  
count = 0  
for list in iris\_x:  
 print(list)  
 count+=1  
print("count:",count)  
print("target\n",iris\_y)  
counttarget = 0  
'''  
  
'''print(counttarget)'''**# 把数据集分为训练集和测试集，其中 test\_size=0.3，即测试集占总数据的 30%：  
X\_train, X\_Test, Y\_train, Y\_Test **=** train\_test\_split**(**iris\_x, iris\_y, test\_size**=**0.3**)  
  
print(**X\_train,**"\n\n")  
print(**X\_Test,**"\n\n")  
print(**Y\_train,**"\n\n")  
print(**Y\_Test,**"\n\n")  
  
'''  
定义模块方式 KNeighborsClassifier()，  
用 fit 来训练 training data，这一步就完成了训练的所有步骤，  
后面的 knn 就已经是训练好的模型，可以直接用来 predict 测试集的数据，  
对比用模型预测的值与真实的值，可以看到大概模拟出了数据，但是有误差，是不会完完全全预测正确的。  
'''**knn **=** KNeighborsClassifier**()**knn.fit**(**X\_train, Y\_train**)**#获得以训练出的模型的测试验证集结果  
**print(**knn.predict**(**X\_Test**))**#输出原结果  
**print(**Y\_Test**)**end **=** knn.predict**(**X\_Test**)**#不要在意这段，这是我得知score()这个玩意前进行的测试，在得知后，呵呵呵呵呵  
All **=** len**(**X\_Test**)**trues**=**0.0  
counts**=** len**(**Y\_Test**)  
for** i **in** range**(**0,counts**):  
 if(**end**[**i**]==**Y\_Test**[**i**]):** trues**+=**1  
**print("验证集精度:"**,trues**/**All**)  
  
print("VVV"**,knn.score**(**X\_Test,Y\_Test**))**

至此告一段落，谢谢看到最后的人听我啰嗦，有兴趣深究的请自行查阅以下网址：

<http://scikit-learn.org/stable/datasets/index.html#datasets>

小白一枚，谬误之处，敬请指出，会在以后的博客中更改，博客也会发一些Java的知识，一些来自自己总结，一些从网上收集，谢谢大家浏览