**Python决策树算法分析天气、周末和促销活动对销量的影响**

**任务目标**

1.掌握决策树的计算原理

2.掌握利用决策树分析商品销量的影响因素

**相关知识**

1. 决策树的基本认识

决策树是一种依托决策而建立起来的一种树。在机器学习中，决策树是一种预测模型，代表的是一种对象属性与对象值之间的一种映射关系，每一个节点代表某个对象，树中的每一个分叉路径代表某个可能的属性值，而每一个叶子节点则对应从根节点到该叶子节点所经历的路径所表示的对象的值。决策树仅有单一输出，如果有多个输出，可以分别建立独立的决策树以处理不同的输出。

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。

决策树最重要的是决策树的构造。所谓决策树的构造就是进行属性选择度量确定各个特征属性之间的拓扑结构。构造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。分裂属性分为三种不同的情况：

1、属性是离散值且不要求生成二叉决策树。此时用属性的每一个划分作为一个分支。

2、属性是离散值且要求生成二叉决策树。此时使用属性划分的一个子集进行测试，按照“属于此子集”和“不属于此子集”分成两个分支。

3、属性是连续值。此时确定一个值作为分裂点split\_point，按照>split\_point和<=split\_point生成两个分支。

信息论中有熵（entropy）的概念，表示状态的混乱程度，熵越大越混乱。熵的变化可以看做是信息增益，决策树ID3算法的核心思想是以信息增益度量属性选择，选择分裂后信息增益最大的属性进行分裂。

设D为用（输出）类别对训练元组进行的划分，则D的熵表示为：

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/01-1.png)

其中pi表示第i个类别在整个训练元组中出现的概率，一般来说会用这个类别的样本数量占总量的占比来作为概率的估计；熵的实际意义表示是D中元组的类标号所需要的平均信息量。

如果将训练元组D按属性A进行划分，则A对D划分的期望信息为：

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/01-2.png)

于是，信息增益就是两者的差值：

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/01-3.png)

ID3决策树算法就用到上面的信息增益，在每次分裂的时候贪心选择信息增益最大的属性，作为本次分裂属性。每次分裂就会使得树长高一层。这样逐步生产下去，就一定可以构建一颗决策树。

**2. ID3算法介绍**

ID3算法是决策树的一种，它是基于奥卡姆剃刀原理的，即尽量用较少的东西做更多的事。ID3算法，即Iterative Dichotomiser 3，迭代二叉树3代，是Ross Quinlan发明的一种决策树算法，这个算法的基础就是上面提到的奥卡姆剃刀原理，越是小型的决策树越优于大的决策树，尽管如此，也不总是生成最小的树型结构，而是一个启发式算法。

在信息论中，期望信息越小，那么信息增益就越大，从而纯度就越高。ID3算法的核心思想就是以信息增益来度量属性的选择，选择分裂后信息增益最大的属性进行分裂。该算法采用自顶向下的贪婪搜索遍历可能的决策空间。

**3. 信息熵与信息增益**

在信息增益中，重要性的衡量标准就是看特征能够为分类系统带来多少信息，带来的信息越多，该特征越重要。在认识信息增益之前，先来看看信息熵的定义。熵这个概念最早起源于物理学，在物理学中是用来度量一个热力学系统的无序程度，而在信息学里面，熵是对不确定性的度量。在1948年，香农引入了信息熵，将其定义为离散随机事件出现的概率，一个系统越是有序，信息熵就越低，反之一个系统越是混乱，它的信息熵就越高。所以信息熵可以被认为是系统有序化程度的一个度量。

假如一个随机变量

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/01-4.png)

的取值为

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/01-5.png)

，每一种取到的概率分别是

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/01-6.png)

，那么

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/01-4.png)

的熵定义为

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/01-7.png)

意思是一个变量的变化情况可能越多，那么它携带的信息量就越大。

对于分类系统来说，类别

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/02-1.png)

是变量，它的取值是

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/02-2.png)

，而每一个类别出现的概率分别是：

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/02-3.png)

而这里的

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/02-4.png)

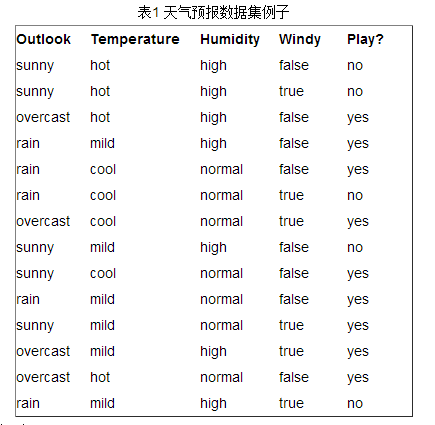
就是类别的总数，此时分类系统的熵就可以表示为：

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/02-5.png)

以上就是信息熵的定义，接下来介绍信息增益。

信息增益是针对一个一个特征而言的，就是看一个特征t，系统有它和没有它时的信息量各是多少，两者的差值就是这个特征给系统带来的信息量，即信息增益。

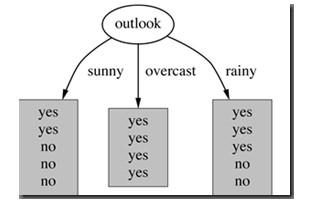
接下来以天气预报的例子来说明。下面是描述天气数据表，学习目标是play或者not play。

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/01.png)

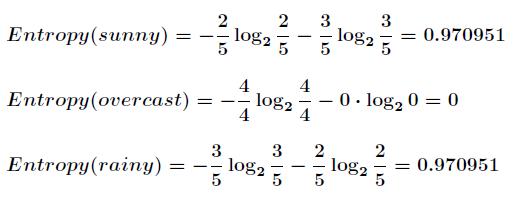
可以看出，一共14个样例，包括9个正例和5个负例。那么当前信息的熵计算如下：

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/02.png)

在决策树分类问题中，信息增益就是决策树在进行属性选择划分前和划分后信息的差值。假设利用属性Outlook来分类，那么如下图：

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/03.jpg)

划分后，数据被分为三部分了，那么各个分支的信息熵计算如下

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/03.png)

那么划分后的信息熵为：

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/04.png)

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/05.png)

代表在特征属性

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/05-1.png)

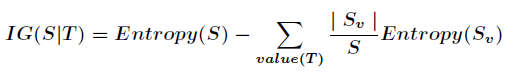
的条件下样本的条件熵。那么最终得到特征属性

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/05-1.png)

带来的信息增益为：

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/06.png)

信息增益的计算公式如下：

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/07.png)

其中

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/07-1.png)

为全部样本集合，

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/07-2.png)

是属性

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/05-1.png)

所有取值的集合，

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/07-3.png)

是

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/05-1.png)

的其中一个属性值，

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/07-4.png)

是

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/07-1.png)

中属性

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/05-1.png)

的值为

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/07-3.png)

的样例集合，

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/07-5.png)

为

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/07-4.png)

中所含样例数。

在决策树的每一个非叶子结点划分之前，先计算每一个属性所带来的信息增益，选择最大信息增益的属性来划分，因为信息增益越大，区分样本的能力就越强，越具有代表性，很显然这是一种自顶向下的贪心策略。

**系统环境**

Linux Ubuntu 16.04

Python3.6

sklearn0.19.0

**任务内容**

T餐饮企业作为大型连锁企业，生产的产品种类比较多，另外涉及的分店所在的位置也不同，数目比较多。对于企业的高层来讲，了解周末和非周末销售量是否有很大的区别，以及天气、促销活动这些因素是否能够影响门店的销售量等信息至关重要。因此，为了让决策者准确了解和销量有关的一系列影响因素，采用算法构建决策树模型，来分析天气、是否周末和是否有促销活动对销量的影响。

**任务步骤**

1.首先在Linux上新建/data/python13目录，并切换到该目录下。

[view plain](https://www.ipieuvre.com/e/182/311/9539) [copy](https://www.ipieuvre.com/e/182/311/9539)

1. sudo mkdir -p /data/python13/

[view plain](https://www.ipieuvre.com/e/182/311/9539) [copy](https://www.ipieuvre.com/e/182/311/9539)

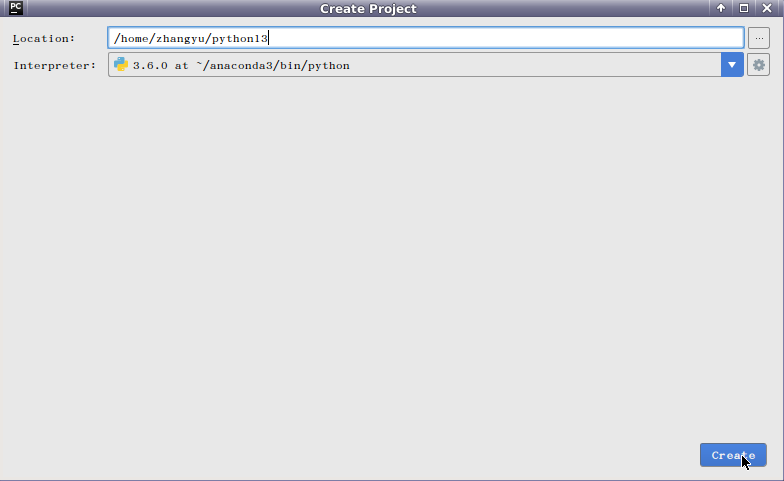
1. cd /data/python13/

2.使用wge命令，从网址 http://192.168.1.100:60000/allfiles/python13/目录下，将实验所需数据下载到linux本地/data/python13目录下。

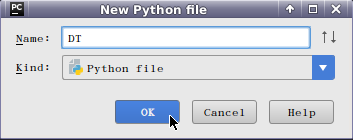
[view plain](https://www.ipieuvre.com/e/182/311/9539) [copy](https://www.ipieuvre.com/e/182/311/9539)

1. sudo wget http://192.168.1.100:60000/allfiles/python13/sales\_data.txt

3.新建Python项目，名为python13

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/08.png)

4.新建python file 文件,名为：DT

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/09.png)

5.导入数据所需的外包

[view plain](https://www.ipieuvre.com/e/182/311/9539) [copy](https://www.ipieuvre.com/e/182/311/9539)

1. **import** pandas as pd
2. from sklearn.tree **import** DecisionTreeClassifier as DTC
3. from sklearn.tree **import** export\_graphviz

6.导入数据

[view plain](https://www.ipieuvre.com/e/182/311/9539) [copy](https://www.ipieuvre.com/e/182/311/9539)

1. filename='/data/python13/sales\_data.txt'
2. data=pd.read\_csv(filename,index\_col='序号')

7.数据预处理

[view plain](https://www.ipieuvre.com/e/182/311/9539) [copy](https://www.ipieuvre.com/e/182/311/9539)

1. data[data=='好']=1
2. data[data=='是']=1
3. data[data=='高']=1
4. data[data!=1]=-1

8.特征提取

[view plain](https://www.ipieuvre.com/e/182/311/9539) [copy](https://www.ipieuvre.com/e/182/311/9539)

1. x=data.iloc[:,:3].as\_matrix().astype(**int**)
2. y=data.iloc[:,3].as\_matrix().astype(**int**)

9.建立决策树模型

[view plain](https://www.ipieuvre.com/e/182/311/9539) [copy](https://www.ipieuvre.com/e/182/311/9539)

1. dtc=DTC(criterion="gini").fit(x,y)

10.模型可视化

[view plain](https://www.ipieuvre.com/e/182/311/9539) [copy](https://www.ipieuvre.com/e/182/311/9539)

1. **with** open( 'tree.dot','w') as f:
2. f=export\_graphviz(dtc,feature\_names=data.columns,out\_file=f)

完整代码

[view plain](https://www.ipieuvre.com/e/182/311/9539) [copy](https://www.ipieuvre.com/e/182/311/9539)

1. #-\*- coding: utf-8 -\*-
2. **import** pandas as pd
3. from sklearn.tree **import** DecisionTreeClassifier as DTC
5. filename='/data/python13/sales\_data.txt'
6. data=pd.read\_csv(filename,index\_col='序号')
7. print(data.columns)
8. data[data=='好']=1
9. data[data=='是']=1
10. data[data=='高']=1
11. data[data!=1]=-1
13. x=data.iloc[:,:3].as\_matrix().astype(**int**)
14. y=data.iloc[:,3].as\_matrix().astype(**int**)
16. dtc=DTC(criterion="gini").fit(x,y)
18. from sklearn.tree **import** export\_graphviz
20. **with** open( 'tree.dot','w') as f:
21. f=export\_graphviz(dtc,feature\_names=data.iloc[:,:3].columns,out\_file=f)

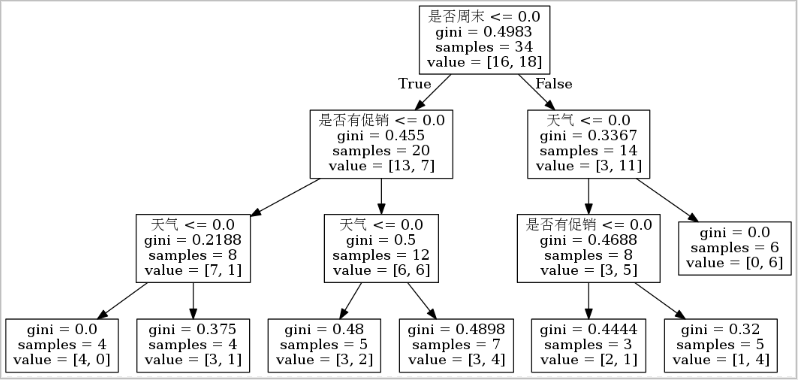
11.运行结果：

可以在当前目录下，看到一个名为tree.dot的文件，切换到当前项目所在目录下：~/python13文件下，以如下方式编译，将其转化为可视化文件tree.png

[view plain](https://www.ipieuvre.com/e/182/311/9539) [copy](https://www.ipieuvre.com/e/182/311/9539)

1. cd ~/python13
2. dot -Tpng tree.dot -o tree.png

12.在pycharm中打开tree.png文件。

[](https://www.ipieuvre.com/doc/exper/287e56ba-91ad-11e9-beeb-00215ec892f4/img/10.png)