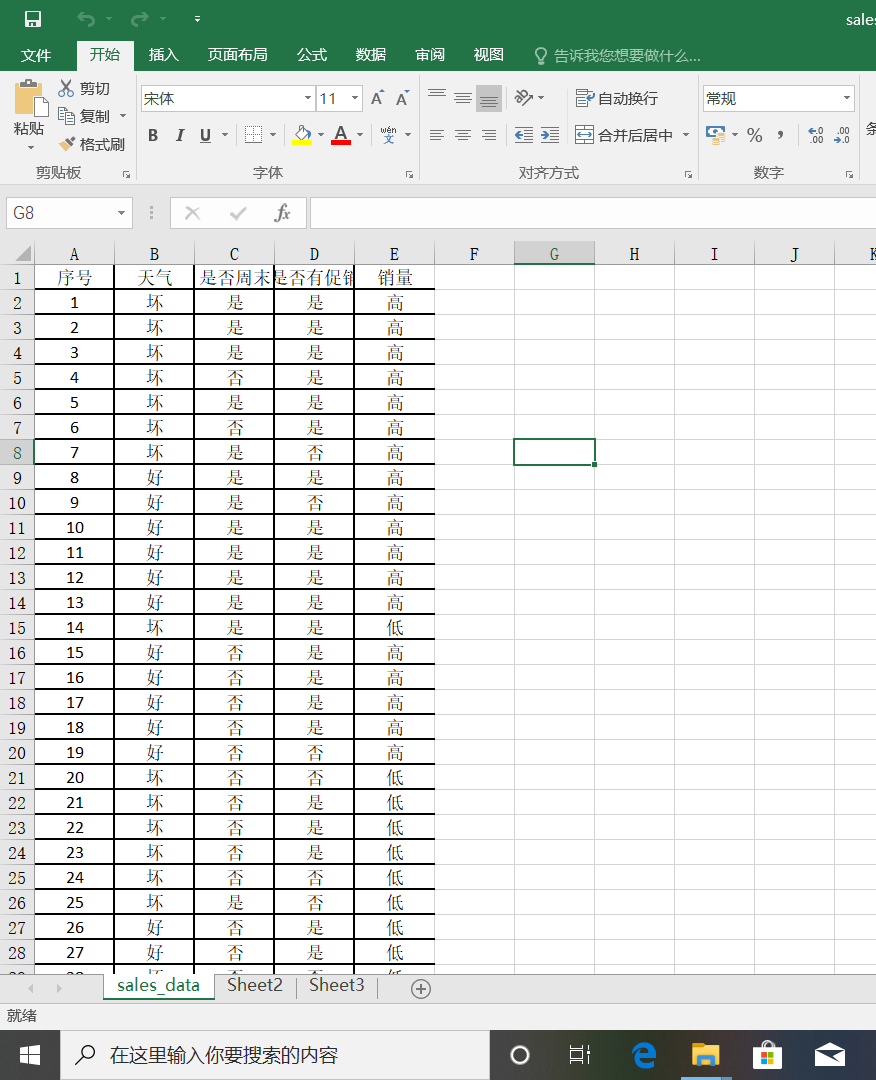
Setdata.xls

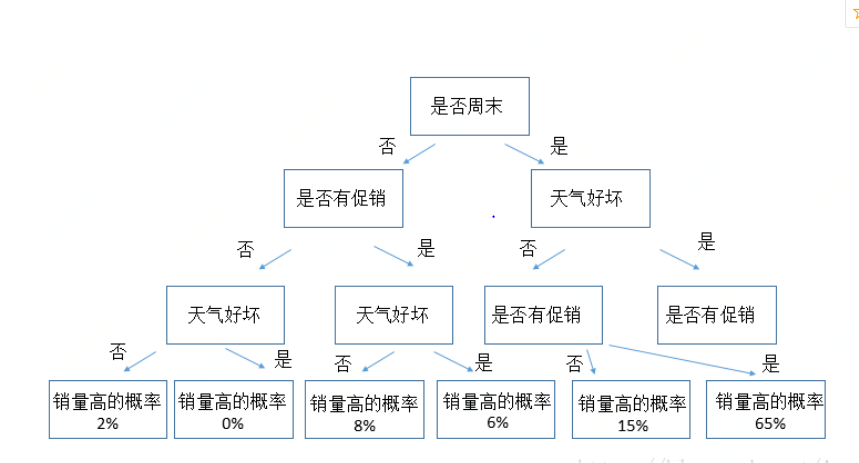


用贝叶斯和决策树处理数据

1. 贝叶斯代码实现

**import** pandas **as** pd  
data=pd.read\_excel(**r'C:\Users\huangjunwen\Desktop\sales\_data.xls'**,index\_col=**'序号'**)  
*#导入数据，将序号列转变成索引*data\_new=data.replace([**'坏'**,**'否'**,**'低'**,**'好'**,**'是'**,**'高'**],[-1,-1,-1,1,1,1])  
*#将数据内容转化成可以计算的数字，-1代表：坏、否、低；1代表：好、是、高***from** sklearn.utils **import** shuffle  
*#这是一个可以将数据重新随机排序的模块*data\_new=shuffle(data\_new)  
*#对数据进行重新随机排列*data\_train=data\_new.iloc[:20,:3]  
*#创建训练集*data\_test=data\_new.iloc[:20,3:]  
*#创建测试的标签集*data\_predict=data\_new.iloc[20:,]  
*#创建最后要预测并检验结果准确的预测集***from** sklearn.naive\_bayes **import** GaussianNB  
*#导入高斯分布的贝叶斯*clf=GaussianNB()  
*#创建高斯分布的贝叶斯模型*clf.fit(data\_train,data\_test)  
*#将训练数据及标签数据放入模型进行训练*data\_predict[**'贝叶斯拟合'**]=clf.predict(data\_predict.iloc[:,:3])  
*#预测结果并赋值到预测集进行模型与实际标签比对*print(len(data\_predict[data\_predict[**'销量'**]==data\_predict[**'贝叶斯拟合'**]])/len(data\_predict))  
*#算出模型准确率*

1. 决策树实现：



**import** pandas **as** pd  
**from** sklearn **import** tree *#导入决策树模块***import** graphviz *#导入可视化模块  
  
#导入数据*data=pd.read\_excel(**r'C:\Users\huangjunwen\Desktop\sales\_data.xls'**)  
  
*#对数据进行清理*data=data.replace([[**'好'**,**'是'**,**'高'**]],1)*#将数据集中的好，是，高替换成数字1*data=data.replace([[**'坏'**,**'否'**,**'低'**]],-1)*#将数据及中的坏，否，低替换成数字-1*x=data.iloc[:,1:4]*#选择训练集*y=data.iloc[:,4]*#选择对应目标标签  
  
#创建决策树模型并设置参数*dtc=tree.DecisionTreeClassifier(criterion=**'entropy'**)*#创建基于信息熵的模型*dtc.fit(x,y)*#训练模型  
  
#模型输出*dot\_data = \ *#设置输出的参数* tree.export\_graphviz(  
 dtc, *#（决策树模型）* out\_file = **None**,  
 feature\_names = [**'天气'**,**'周末'**,**'促销'**], *#模型中对应标签名称* filled = **True**,  
 impurity = **False**,  
 rounded = **True** )  
dot\_data=dot\_data.replace(**'helvetica'**,**'"Microsoft Yahei"'**)  
*#因为标签是中文所以需要将参数设置成支持微软雅黑的格式*graph=graphviz.Source(dot\_data)*#选择要可视化的dot数据*graph.render(**r'C:\Users\huangjunwen\Desktop\tree'**)*#将可视化结果输出至指定位置*

具体分析：

贝叶斯算法：

正向概率：n白m黑 伸手摸一个 黑球概率 正向概率 黑球个数/总个数

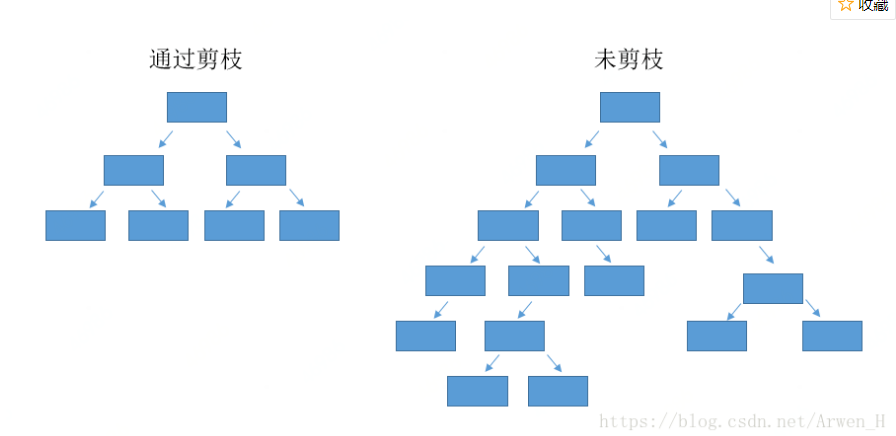
逆向概率：摸多个球判断袋子里的黑白球个数

优点：古典数学理论 有稳定的分类效率 对小规模数据处理的比较好 能处理多分类任务 适合增量式训练 对缺失数据不敏感算法比较简单

缺点：误差较大 预测效果不佳 存在错误率

决策树算法：将所有数据落到叶子结点，分类回归，

决策树剪枝：因为决策树过拟合风险很大，



剪枝策略：预剪枝和后剪枝