

## COMP9414: 人工智能教程6：机器学习

1. 为以下一组例子构建一个决策树。

天	展望	温度	湿度	风	玩转网球
D1	阳光	热销	高	弱点	没有
D2	阳光	热销	高	强大的	没有
D3	阴天	热销	高	弱点	是
D4	雨	温和的	高	弱点	是
D5	雨	凉爽	正常	弱点	是
D6	雨	酷	正常	强大的	没有
D7	阴天	酷	正常	强大的	是
D8	阳光	温和的	高	弱点	没有
D9	阳光	凉爽	正常	弱点	是
D10	雨	温和的	正常	弱点	是
D11	阳光	温和的	正常	强大的	是
D12	阴天	温和的	高	强大的	是
D13	阴天	热销	正常	弱点	是
D14	雨	温和的	高	强大的	没有

赋予实例 {D15, Sunny, Hot, High, Weak} 什么类别？

2. 考虑对同一组实例采用Naive Bayes分类器。现在分配给实例 {D15, Sunny, Hot, High, Weak} 的是什么类别？
3. 编程。试试NLTK的Naive Bayes分类器。在这里，你可以加载一组文档，将它们转换成特征，然后训练和测试分类器。下面的例子是关于moview的评论。类别是 "neg" 和 "pos"，文档特征是 "真" 或 "假"，取决于文档中是否包含某个词。

```
from nltk import FreqDist, NaiveBayesClassifier
from nltk.corpus import movie_reviews
from nltk.classify import accuracy
import random

documents = [(list(movie_reviews.words(fileid)),
                 category) for category in movie_reviews.
               categories()
               for fileid in movie_reviews.fileids(category)]
random.shuffle(documents) # 这一行是对文件顺序的洗牌
all_words = FreqDist(w.lower() for w in movie_reviews.words())
word_features = list(all_words)[:2000]

def document_features(document):
    document_words = set(document)
    features = {}
    for word in word_features:
        features['contains({})'.format(word)] = (word in document_words)
    return features

featuresets = [(document_features(d), c) for (d,c) in documents]
train_set, test_set = featuresets[100:], featuresets[:100]
# Split data
classifier = NaiveBayesClassifier.train(train_set)
print(accuracy(classifier, test_set))
```