Data Mining
Home work 08
Machine Learning Start...

Aqeel Labash
Lecturer: Jaak Vilo

29 March 2016

First Question

The quality of classifier as I understood from the page is when we can classify accurately depending on that classifier.By that I mean to have a certain point where it completely separate data into two groups.
Another meaning for quality of classifier might be if the classifier really represent a real case or just something happened with high probability in training dataset.If it's just high probability then depending on that classifier will just make our model worst at predicting with real data or unseen data.

Second Question

For this question I build the decision tree and put all results on it by hand and here is the drawn :

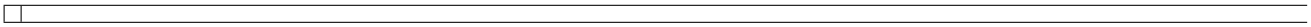

Figure 1: Draw show the decision tree

After that I just used some code to draw the table here is the code :

```
# coding: utf-8
import csv
with open('data.csv') as f:
spam = csv.DictReader(f)
trainset = list(spam)
trainset  = sorted(trainset, key=lambda k: k['Play'])
len(trainset)
class v:
def __init__(self):
self.lst={}

def AddItem(self,values):

if len(values)<=0:
return
print values[0]
if values[0] in self.lst.keys():
self.lst[values[0]].AddItem(values[1:])
else:
self.lst[values[0]]= v()
self.lst[values[0]].AddItem(values[1:])
def printeverything(self,level=0):
#print
for itm in self.lst.keys():
thespace = '----'*level
print(thespace+itm)
self.lst[itm].printeverything(level=level+1)
class Core:
def __init__(self,name=None,occurence=0):
self.name = name
self.occurence =occurence
self.sons={}
def AddItem(self,items):
if len(items)<=0:
```

```
return
if items[0] in self.sons.keys():
self.sons[items[0]].occurence+=1
self.sons[items[0]].AddItem(items[1:])
else:
self.sons[items[0]] = Core(name=items[0],occurence=1)
self.sons[items[0]].AddItem(items[1:])
def printeverything(self,level=0):
#print
thespace = '----'*level
print(thespace+str(self.name)+':'+str(self.occurence))
for itm in self.sons.keys():
self.sons[itm].printeverything(level=level+1)
root = Core()
for i in trainset:
root.AddItem((i['Outlook'],i['Temp'],i['Humidity'],i['Windy'],i['Play']))
root.printeverything()
```

Which result to the following table :

```
None:0
-----Rainy:5
----------Mild:3
----------High:2
----------FALSE:1
----------Yes:1
----------TRUE:1
----------No:1
----------Normal:1
----------FALSE:1
----------Yes:1
----------Cool:2
----------Normal:2
----------FALSE:1
----------Yes:1
----------TRUE:1
----------No:1
-----Overcast:5
----------Hot:2
----------High:1
----------FALSE:1
----------Yes:1
----------Normal:1
----------FALSE:1
----------Yes:1
----------Mild:1
----------High:1
----------TRUE:1
----------Yes:1
----------Cool:2
----------High:1
----------FALSE:1
----------No:1
----------Normal:1
----------TRUE:1
----------Yes:1
-----Sunny:5
----------Hot:2
----------High:2
----------TRUE:1
----------No:1
----------FALSE:1
----------No:1
----------Mild:2
----------High:1
----------FALSE:1
----------No:1
----------Normal:1
----------TRUE:1
----------Yes:1
----------Cool:1
----------Normal:1
----------FALSE:1
----------Yes:1
```

What I did is just put all the possible decisions

Third Question

Fourth Question

Fifth Question

Sixth Question