NAME : AVIREDDY NVSRK ROHAN

REG.no:19BCE1180

```
import pandas as pd
import·numpy·as·np
import·math
```

```
data=pd.read_csv("/content/Buy_Computer.csv")
data
```

	id	age	income	student	credit_rating	Buy_Computer
0	1	youth	high	no	fair	no
1	2	youth	high	no	excellent	no
2	3	middle_age	high	no	fair	yes
3	4	senior	medium	no	fair	yes
4	5	senior	low	yes	fair	yes
5	6	senior	low	yes	excellent	no
6	7	middle_age	low	yes	excellent	yes
7	8	youth	medium	no	fair	no
8	9	youth	low	yes	fair	yes
9	10	senior	medium	yes	fair	yes
10	11	youth	medium	yes	excellent	yes
11	12	middle_age	medium	no	excellent	yes
12	13	middle_age	high	yes	fair	yes
13	14	senior	medium	no	excellent	no

```
for i in data.columns:
  print(data[i].unique())
```

```
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14]
['youth' 'middle_age' 'senior']
['high' 'medium' 'low']
['no' 'yes']
['fair' 'excellent']
['no' 'yes']
```

FUNCTION INFORMATION

1.GAIN : returns the difference between label entropy and residual information for the respective class value in an attribute

2.CONDITIONAL_PROB :returns the conditional probability value for the respective class value in an attribute relative to the class value in the target label

3.CLASS_PROB :returns the class probability value of a class value within an attribute

4.RESIDUAL_INFO :it is the sum of product of class probability with its weighted entropy

5.LABEL_ENTROPY :gives the weighted entropy just for the target label

6.ENTROPY : gives the weighted entropy value

```
def class_prob(class_val,atr):
    p=data[atr]==class_val
    #print("CP "+atr+" "+class_val)
    return p.sum()/data.shape[0]
def residual_info(atr,label):
    sum=0
    #print("RI "+atr+" "+label)
    for i in data[atr].unique():
        sum=sum+class_prob(i,atr)*entropy(atr,i,label)
    return sum

def gain(atr,label):
    #print("gain "+atr+" "+label)

    return label_entropy(label)-residual_info(atr,label)

def conditional_prob(col,class_val,label):
      temp=[]
      for j in data[label].unique():
          c=(data[col]==class_val) & (data[label]==j)
          temp.append(c.sum()/(data[col]==class_val).sum())
      return temp
def label_entropy(label):
    sum=0
    #print("LE "+label)
    for i in data[label].unique():
        p=class_prob(i,label)
        sum=sum+(p*np.log2(p))
    return -1*sum
def entropy(col,class_val,label):
    #print("ENT "+col,class_val,label)
    p=conditional_prob(col,class_val,label)
    sum=0
    for i in p:
      if i!=0:
         sum=sum+i*np.log2(i)
    return -1*sum
```

```
gain_data=[]
for i in data.columns[1:5]:
  gain_data.append([i,gain(i,data.columns[5])])
gain_data=pd.DataFrame(gain_data)
gain_data.columns=["Attribute","Information Gain"]
```

information gain values

```
gain_data
```

	Attribute	Information Gain
0	age	0.246750
1	income	0.029223
2	student	0.151836
3	credit_rating	0.048127

```
label_entropy("Buy_Computer")
```

```
0.9402859586706309
```

```
residual_info("age","Buy_Computer")
```

```
0.6935361388961918
```

```
residual_info("income","Buy_Computer")
```

```
0.9110633930116763
```

```
residual_info("student","Buy_Computer")
```

```
    0.7884504573082896
```

```
residual_info("credit_rating","Buy_Computer")
```

```
    0.8921589282623617
```

✓ 0s completed at 12:06