**Program4: Decision based on Gini index**

*# Algorithm : gini*

**import** pandas **as** pd  
**import** numpy **as** np  
**import** math  
  
maindf = pd.read\_csv(**'data.csv'**)  
mainAttributes = maindf.columns[0:-1]  
last = maindf.columns[-1]  
  
**def** giniByV(a,b):  
 gini = 1 - (a/(a+b))\*\*2 - (b/(a+b))\*\*2  
 **return** gini.\_\_round\_\_(4)  
  
**def** giniBydf(df1,df2=pd.DataFrame()):  
 uniqueC = df1[last].value\_counts().to\_numpy()  
 a = uniqueC[0]  
 b = 0  
 **if** len(uniqueC)>1: b = uniqueC[1]  
 totaldf1 = a+b  
 **if** df2.empty == **True**:  
 **return** giniByV(a,b)  
 **else**:  
 uniqueC1 = df2.iloc[:, -1].value\_counts().to\_numpy()  
 a1 = uniqueC1[0]  
 b1 = 0  
 **if** len(uniqueC1) > 1: b1 = uniqueC1[1]  
 totaldf2 = a1+b1  
 total = totaldf1+totaldf2  
 ginid1 = giniByV(a,b)  
 ginid2 = giniByV(a1,b1)  
 **return** ((totaldf1/total)\*ginid1 + (totaldf2/total)\*ginid2).\_\_round\_\_(4)  
  
**def** findDs(attribute,df):  
 attrUnique = list(df[attribute].unique())*# ex: gives youth,adult,senior* subset = {}  
 d = {}  
 *#to create subsets of having unique attr more than 2* **if** len(attrUnique)>2:  
 **for** attr **in** attrUnique:  
 index = attrUnique.index(attr)  
 subset[attr] = attrUnique[0:index] + attrUnique[index+1:]  
 *# print(subset)  
 #to find d for perticular main attribute* **for** key,values **in** subset.items():  
 keydf = df.loc[(df[attribute] == key), [attribute, last]]  
 valuesdf = pd.DataFrame()  
 **if** len(values) > 1:  
 **for** value **in** values:  
 valuesdf = valuesdf.append(df.loc[(df[attribute] == value), [attribute, last]])*#it will return new obj so assigning is required  
 # print(keydf,"\n",valuesdf)* d[key] = giniBydf(keydf,valuesdf)  
 **else**:  
 df1 = df.loc[(df[attribute] == attrUnique[0]), [attribute, last]]  
 df2 = df.loc[(df[attribute] == attrUnique[1]), [attribute, last]]  
 d[attrUnique[0]] = giniBydf(df1,df2)  
 **return** d  
  
**def** findMaxReduction(attributes,df):  
 maxReduction = {**'key'**:**None**,**'value'**:-100}  
 **for** attribute **in** attributes:  
 valueDic = findDs(attribute,df)  
 minimum = min(valueDic.items())  
 reduction = giniBydf(df) - minimum[1]  
 **if** reduction > maxReduction[**'value'**]:  
 maxReduction[**'value'**] = reduction.\_\_round\_\_(4)  
 *# print(type(maxReduction))* maxReduction[**'key'**] = attribute  
 **return** maxReduction  
  
  
*# root node*print(**'Root Node -'**)  
root = findMaxReduction(mainAttributes,maindf)  
print(root)  
  
*#children at level 1*print(**'Level 1 -'**)  
level\_1 = {}  
level\_1\_nodes = maindf[root[**'key'**]].unique()  
newAttrs = mainAttributes.to\_list()  
newAttrs.remove(root[**'key'**])  
newAttrs.append(last)  
**for** node **in** level\_1\_nodes:  
 newDf = maindf.loc[(maindf[root[**'key'**]] == node), newAttrs]  
 *# print(newDf)* gini = giniBydf(newDf)  
 **if** gini == 0:  
 level\_1[node] = 0  
 **continue** attrs = newAttrs[:-1]  
 ans = findMaxReduction(attrs,newDf)  
 level\_1[node] = ans[**'key'**]  
print(level\_1)

Dataset :

***age***,***income***,***student***,***credit***,***buy  
youth***,***high***,***no***,***fair***,***no  
youth***,***high***,***no***,***excellent***,***no  
adult***,***high***,***no***,***fair***,***yes  
senior***,***medium***,***no***,***fair***,***yes  
senior***,***low***,***yes***,***fair***,***yes  
senior***,***low***,***yes***,***excellent***,***no  
adult***,***low***,***yes***,***excellent***,***yes  
youth***,***medium***,***no***,***fair***,***no  
youth***,***low***,***yes***,***fair***,***yes  
senior***,***medium***,***yes***,***fair***,***yes  
youth***,***medium***,***yes***,***excellent***,***yes  
adult***,***medium***,***no***,***excellent***,***yes  
adult***,***high***,***yes***,***fair***,***yes  
senior***,***medium***,***no***,***excellent***,***no***

Output:

