# Code:

**import** csv  
**import** numpy **as** np  
**with** open(**"Restaurent.csv"**) **as** f:  
 dataset=csv.reader(f)  
 restuarent=list(dataset)  
print(**"Restaurent dataset \n"**,restuarent[0])  
  
**def** calculate\_entropy\_Info\_Class2(feature,index):  
 yy=0  
 yn=0  
 ny=0  
 nn=0  
 **for** i **in** restuarent[1:]:  
 feature=i[index]  
 willwait=i[10]  
 **if** feature==**"yes" and** willwait==**"yes"**:  
 yy+=1  
 **elif** feature==**"yes" and** willwait==**"no"**:  
 yn+=1  
 **elif** feature==**"no" and** willwait==**"yes"**:  
 ny+=1  
 **elif** feature==**"no" and** willwait==**"no"**:  
 nn+=1  
 **return** [(yy,yn),(ny,nn)]  
**def** calculate\_entropy\_Info\_Class3(feature,index):  
 Sy=0  
 Sn=0  
 Fy=0  
 Fn=0  
 Ny=0  
 Nn=0  
 feature2=[**"some"**,**"full"**,**"none"**]  
 feature3=[**"$$$"**,**"$$"**,**"$"**]  
 **if** feature==**"Pat"**:  
 feature1=feature2  
 **elif** feature==**"Price"**:  
 feature1=feature3  
 **for** i **in** restuarent[1:]:  
 feature=i[index]  
 willwait=i[10]  
 **if** feature==feature1[0] **and** willwait==**"yes"**:  
 Sy+=1  
 **elif** feature==feature1[0] **and** willwait==**"no"**:  
 Sn+=1  
 **elif** feature==feature1[1] **and** willwait==**"yes"**:  
 Fy+=1  
 **elif** feature==feature1[1] **and** willwait==**"no"**:  
 Fn+=1  
 **elif** feature==feature1[2] **and** willwait==**"yes"**:  
 Ny+=1  
 **elif** feature==feature1[2] **and** willwait==**"no"**:  
 Nn+=1  
 **return** [(Sy,Sn),(Fy,Fn),(Ny,Nn)]  
**def** calculate\_entropy\_Info\_Class4(feature, index):  
 Sy = 0  
 Sn = 0  
 Fy = 0  
 Fn = 0  
 Ny = 0  
 Nn = 0  
 Iy = 0  
 In = 0  
 feature2 = [**"french"**, **"thai"**, **"burger"**, **"italian"**]  
 feature3 = [**"0-10"**, **"10-30"**, **"30-60"**, **">60"**]  
 **if** feature == **"Type"**:  
 feature1 = feature2  
 **elif** feature == **"Est"**:  
 feature1 = feature3  
 **for** i **in** restuarent[1:]:  
 feature = i[index]  
 willwait = i[10]  
 **if** feature == feature1[0] **and** willwait == **"yes"**:  
 Sy += 1  
 **elif** feature == feature1[0] **and** willwait == **"no"**:  
 Sn += 1  
 **elif** feature == feature1[1] **and** willwait == **"yes"**:  
 Fy += 1  
 **elif** feature == feature1[1] **and** willwait == **"no"**:  
 Fn += 1  
 **elif** feature == feature1[2] **and** willwait == **"yes"**:  
 Ny += 1  
 **elif** feature == feature1[2] **and** willwait == **"no"**:  
 Nn += 1  
 **elif** feature == feature1[3] **and** willwait == **"yes"**:  
 Iy += 1  
 **elif** feature == feature1[3] **and** willwait == **"no"**:  
 In += 1  
 **return** [(Sy, Sn), (Fy, Fn), (Ny, Nn), (Iy, In)]  
*# class 2 calculation of entropy*Alt\_Info=calculate\_entropy\_Info\_Class2(**"Alt"**,0)  
Bar\_Info=calculate\_entropy\_Info\_Class2(**"Bar"**,1)  
Fri\_Info=calculate\_entropy\_Info\_Class2(**"Fri"**,2)  
Hun\_Info=calculate\_entropy\_Info\_Class2(**"Hun"**,3)  
Rain\_Info=calculate\_entropy\_Info\_Class2(**"rain"**,6)  
Res\_Info=calculate\_entropy\_Info\_Class2(**"Res"**,7)  
*#class 3 calculation of entropy*Pat\_Info=calculate\_entropy\_Info\_Class3(**"Pat"**,4)  
Price\_Info=calculate\_entropy\_Info\_Class3(**"Price"**,5)  
*#class 4 calculation of entropy*Type\_Info=calculate\_entropy\_Info\_Class4(**"Type"**,8)  
Est\_Info=calculate\_entropy\_Info\_Class4(**"Est"**,9)  
print(Alt\_Info,Bar\_Info,Hun\_Info,Rain\_Info,Res\_Info)  
print(Type\_Info,Est\_Info)  
print(Pat\_Info,Price\_Info)  
*#step2 calculate entropy of classes***def** calculateEntropyInstances(Info):  
 entropy=[]  
 **for** I **in** Info:  
 total=I[0]+I[1]  
 **if** I[0] != 0:  
 phase1= (I[0]/total) \*np.log(I[0]/total)  
 **else**:  
 phase1=0  
 **if** I[1] != 0:  
 phase2= (I[1]/total)\*np.log(I[1]/total)   
 **else**:  
 phase2=0  
 total= -phase1-phase2  
 entropy.append(total)  
 **return** entropy  
  
Alt\_E=calculateEntropyInstances(Alt\_Info)  
Bar\_E=calculateEntropyInstances(Bar\_Info)  
Hun\_E=calculateEntropyInstances(Hun\_Info)  
Rain\_E=calculateEntropyInstances(Rain\_Info)  
Res\_E=calculateEntropyInstances(Res\_Info)  
Pat\_E=calculateEntropyInstances(Pat\_Info)  
Price\_E=calculateEntropyInstances(Price\_Info)  
Type\_E=calculateEntropyInstances(Type\_Info)  
Est\_E=calculateEntropyInstances(Est\_Info)  
print(**"Class wise Entropy Values:\n"**,Alt\_E,Bar\_E,Hun\_E,Rain\_E,Res\_E,Pat\_E,Price\_E,Type\_E,Est\_E)  
  
*#step3 calculate entropy of features***def** calculateEntropyFeature(Info,Alt\_E):  
 entropy=0  
 **def** Total(Info):  
 total=0  
 **for** i **in** Info:  
 total+=i[0]+i[1]  
 **return** total  
 **for** i **in** Info:  
 sum=i[0]+i[1]  
 ind=Info.index(i)  
 v1=sum/Total(Info)  
 v2=Alt\_E[ind]  
 value=v1\*v2  
 print(value)  
 entropy += value  
 **return** entropy  
   
Alt\_TE=calculateEntropyFeature(Alt\_Info,Alt\_E)  
Bar\_TE=calculateEntropyFeature(Bar\_Info,Bar\_E)  
Hun\_TE=calculateEntropyFeature(Hun\_Info,Hun\_E)  
Rain\_TE=calculateEntropyFeature(Rain\_Info,Rain\_E)  
Res\_TE=calculateEntropyFeature(Res\_Info,Res\_E)  
Pat\_TE=calculateEntropyFeature(Pat\_Info,Pat\_E)  
Price\_TE=calculateEntropyFeature(Price\_Info,Price\_E)  
Type\_TE=calculateEntropyFeature(Type\_Info,Type\_E)  
Est\_TE=calculateEntropyFeature(Est\_Info,Est\_E)  
print(**"Colum wise Entropy Values:\n"**,Alt\_TE,Bar\_TE,Hun\_TE,Rain\_TE,Res\_TE,Pat\_TE,Price\_TE,Type\_TE,Est\_TE)  
*#step 4 Computing Info gain***def** InfoGain(Alt\_TE):  
 Info\_gain=1-Alt\_TE  
 **return** Info\_gain  
  
Alt\_IG=InfoGain(Alt\_TE)  
Bar\_IG=InfoGain(Bar\_TE)  
Hun\_IG=InfoGain(Hun\_TE)  
Rain\_IG=InfoGain(Rain\_TE)  
Res\_IG=InfoGain(Res\_TE)  
Pat\_IG=InfoGain(Pat\_TE)  
Price\_IG=InfoGain(Price\_TE)  
Type\_IG=InfoGain(Type\_TE)  
Est\_IG=InfoGain(Est\_TE)  
print(**"Info Gain Values:\n"**,Alt\_IG,Bar\_IG,Hun\_IG,Rain\_IG,Res\_IG,Pat\_IG,Price\_IG,Type\_IG,Est\_IG)

# Output:

Colum wise Entropy Values:

0.6931471805599453 0.6931471805599453 0.5574916031489784 0.6787845889957987 0.6787845889957987 0.3182570841474064 0.5574916031489784 0.6931471805599453 0.5493061443340548

Info Gain Values:

Alt: 0.3068528194400547

Bar: 0.3068528194400547

Hun: 0.4425083968510216

rain: 0.32121541100420126

Res: 0.32121541100420126

Pat: 0.6817429158525936

Price: 0.4425083968510216

Type: 0.3068528194400547

Est: 0.4506938556659452

# Graph:

