Entropy

Entropy-it is a measure of the uncertainty of some system and information. It allows us to make precise statements and perform computations towards things that we don't sure how they will turn out.Here's formula:

$$\hat{H} = -\sum_{i=1} p_i log_2 p_i$$

Conditonal Entropy

Let's say we have 2 random variables X and Y. So we can say that conditonal entropy of X is quantitative measure of uncertainty when we know random value of random variable Y.

Here's formula: $H(\hat{X}|Y) = -\sum p(X, Y)logp(Y, X)$

Mutual Information

mutual information it's calculated between two variables(X and Y) and measures the reduction in uncertainty for one variable given a known value of the other variable. The mutual information that X gives about Y equals the mutual information that Y gives about X.

Here's formula: $H(\hat{X}; Y) = -\sum_{x,y} p(X, Y)log(p(Y, X)/p(Y))p_i$

Inforamtion Gain

information gain measures the level of clogs in a group of examples, in another worlds it's measures the reduction in entropy. It's acomplish by splitting data according to a given value of a random variable

Here's formula: Information Gain =entropy(parent) - [weightes_average]*entropy(chilfren)

In [4]:

```
#entropy calculation
dataSet=[0,0,0,1,1,1,1,1]

import math
def calcEntropy(data):
    entropy_val=0
    processed=list()
    for d in data:
        if d not in processed:
            processed.append(d)
            p_i=data.count(d)/len(data)
            entropy_val+=p_i*math.log2(p_i)
    return entropy_val*(-1)

print(f"Entropy value of {dataSet} is: {calcEntropy(dataSet)}")
```

```
Entropy value of [0, 0, 0, 1, 1, 1, 1, 1] is: 0.954434002924965
```

In [15]:

```
#Conditional entropy calculation
import math
dataSet_X=[1,1,1,0,0]
dataSet_Y=[0,1,0,1,0]

def calcEntropy(data):
    entropy_val=0
    processed=list()
    for d in data:
        if d not in processed:
            processed.append(d)
            p_i=data.count(d)/len(data)
            entropy_val+=p_i*math.log2(p_i)
    return entropy_val*(-1)

def JentropyCalc(Y, X):
    temp=list()
    for x in range(len(Y)):
        yVal=[Y[x],X[x]]
        temp.append(yVal)
    return calcEntropy(temp)

def calcCondEntropy(Y, X):
    return JentropyCalc(Y, X)- calcEntropy(X)

print(f"Conditional entropy value of {dataSet_X} and {dataSet_Y} is: {calcCondEntropy(dataS
```

```
Conditional entropy value of [1, 1, 1, 0, 0] and [0, 1, 0, 1, 0] is: 0.95097
75004326937
```

In [16]:

```
#Mutual Information calculation
import math
dataSet_X=[1,1,1,0,0]
dataSet_Y=[0,1,0,1,0]

def calcEntropy(data):
    entropy_val=0
    processed=list()
    for d in data:
        if d not in processed:
            processed.append(d)
            p_i=data.count(d)/len(data)
            entropy_val+=p_i*math.log2(p_i)
    return entropy_val*(-1)

def MutInformation(data1,data2):
    return calcEntropy(data1) - calcCondEntropy(data1,data2)
print(f"Mutual Information value of {dataSet_X} and {dataSet_Y} is: {MutInformation(dataSet
```

```
Mutual Information value of [1, 1, 1, 0, 0] and [0, 1, 0, 1, 0] is: 0.019973
09402197489
```

Python's libraries for different types entropy calculation

In python we have Pyitlib library-is an MIT-licensed library of information-theoretic methods for data analysis and machine learning, implemented in Python and NumPy. This library implements 19 measures on discrete random variables such as: entropy, Joint and Conditional Entropy, mutual information and others.For the begining you need to install this lubrary on you computer and after that to do this imports:

1)import numpy as np

2)from pyitlib import discrete_random_variable as drv

All information you can find here: [pyitlib 0.2.2 (https://pypi.org/project/pyitlib/)](https://pypi.org/project/pyitlib/)