## Experiment no. 4

**Name: Sonali Dattatray Kaingade**

**PRN: 21620002**

**Title:** Find info gain of an attribute from given data.

**Code:**

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <map>
#include <cmath>
using namespace std;

// Function to calculate entropy
double calculateEntropy(int positive, int negative)
{
   double total = positive + negative;
   double entropy = 0.0;

   if (total > 0)
   {
      double positiveProbability = positive / total;
      double negativeProbability = negative / total;

      if (positiveProbability > 0)
```

```
        {
            entropy -= positiveProbability * log2(positiveProbability);
        }

        if (negativeProbability > 0)
        {
            entropy -= negativeProbability * log2(negativeProbability);
        }
    }

    return entropy;
}

// Function to compute information gain
double computeInformationGain(map<string, int> &parentCounts, map<string, map<string, int>>
&childCounts)
{
    double positiveParent = parentCounts["Yes"];
    double negativeParent = parentCounts["No"];
    double totalParent = positiveParent + negativeParent;

    double parentEntropy = calculateEntropy(positiveParent, negativeParent);
    cout << "Parent Entropy: " << parentEntropy << "\n";

    double childEntropy = 0;
```

```
    for (auto it = childCounts.begin(); it != childCounts.end(); ++it)
    {
        string childName = it->first;
        double positiveChild = it->second["Yes"];
        double negativeChild = it->second["No"];
        double totalChild = positiveChild + negativeChild;

        double childEntropyPart = calculateEntropy(positiveChild, negativeChild);

        childEntropy += (totalChild / totalParent) * childEntropyPart;
    }

    cout << "Weighted Child Entropy: " << childEntropy << "\n";

    double informationGain = parentEntropy - childEntropy;
    cout << "Information Gain: " << informationGain << "\n";

    return informationGain;
}

int main()
{
    ifstream file("info-gain.csv");

    string line, day, level, routine, playGame, value;
```

```
map<string, int> parentCounts;
map<string, map<string, int>> childCounts;

if (!file.is_open())
{
   cerr << "Error opening input file." << endl;
   return -1;
}

int i = 0;
string childName;
int choice;

while (getline(file, line))
{
   stringstream str(line);
   getline(str, day, ',');
   getline(str, level, ',');
   getline(str, routine, ',');
   getline(str, playGame, ',');
   getline(str, value, ',');

   if (i == 0)
   {
      i++;
```

```
    cout << "Enter Child Column Number: ";
    cin >> choice;
    continue;
}

switch (choice)
{
case 1:
    childName = day;
    break;

case 2:
    childName = level;
    break;

case 3:
    childName = routine;
    break;

case 4:
    childName = value;
    break;

default:
    childName = routine;
```

```
            break;
        }

        parentCounts[playGame]++;
        childCounts[childName][playGame]++;
    }

    double informationGain = computeInformationGain(parentCounts, childCounts);

    cout << "Overall Information Gain: " << informationGain << "\n";

    return 0;
}
```

**Input file:**

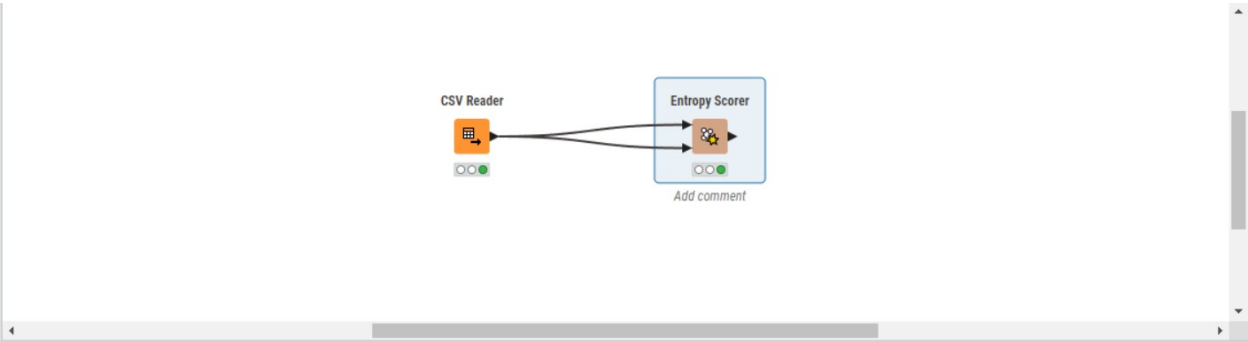
| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Day | Level | Routine | PlayGame | Value | |
| 2 | 1 | High | InDoor | No | FALSE | |
| 3 | 2 | High | OutDoor | No | FALSE | |
| 4 | 3 | High | InDoor | Yes | TRUE | |
| 5 | 4 | High | InDoor | Yes | TRUE | |
| 6 | 5 | Normal | InDoor | Yes | TRUE | |
| 7 | 6 | Normal | OutDoor | No | FALSE | |
| 8 | 7 | Normal | OutDoor | Yes | TRUE | |
| 9 | 8 | High | InDoor | No | FALSE | |
| 10 | 9 | Normal | InDoor | Yes | TRUE | |
| 11 | 10 | Normal | InDoor | Yes | TRUE | |
| 12 | 11 | Normal | OutDoor | Yes | TRUE | |
| 13 | 12 | High | OutDoor | Yes | TRUE | |
| 14 | 13 | Normal | InDoor | Yes | TRUE | |
| 15 | 14 | High | OutDoor | No | FALSE | |
| 16 | | | | | | |
| 17 | | | | | | |

**Output:**

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\second year engg IT\7th sem\dm lab\13th experiment> cd ..
PS E:\second year engg IT\7th sem\dm lab> cd '.\4th experiment\'
PS E:\second year engg IT\7th sem\dm lab\4th experiment> g++ infogain.cpp -o infog
PS E:\second year engg IT\7th sem\dm lab\4th experiment> ./infog
Enter Child Column Number: 2
Parent Entropy: 0.940286
Weighted Child Entropy: 0.78845
Information Gain: 0.151836
Overall Information Gain: 0.151836
PS E:\second year engg IT\7th sem\dm lab\4th experiment>
```

**knime:**




1: Quality Table | Flow Variables

Rows: 4 | Columns: 4 — Table | Statistics

| # | Row... | Size (Number (integer)) | Entropy (Number (double)) | Normalized Entropy (Number (double)) | Quality (Number (double)) |
|---|---|---|---|---|---|
| 1 | Over... | 4 | 0 | 0 | ? |
| 2 | Sunny | 5 | 0.971 | 0.971 | ? |
| 3 | Rain | 5 | 0.971 | 0.971 | ? |
| 4 | Overall | 14 | 0.694 | 0.694 | 0.306 |