Semester	T.E. Semester V – Computer Engineering
Subject	Data Warehousing and Mining
Subject Professor In-charge	Prof. Kavita Shirsat
Assisting Teachers	Prof. Kavita Shirsat
Laboratory	M-313A

Student Name	Vibodh Bhosure	
Roll Number	20102A0032	
Grade and Subject Teacher's Signature		

Experiment Number	03	
Experiment Title	To implement ID3 algorithm on a dataset and find entropy for each attribute	
Resources / Apparatus Required	Hardware: Computer system	Software: Python
Description	In simple words, a decision tree is a structure that contains nodes (rectangular boxes) and edges(arrows) and is built from a dataset (table of columns representing features/attributes and rows corresponds to records). Each node is either used to decide (known as decision node) or represent an outcome (known as leaf node). The initial node is called the root node (colored in blue), the final nodes are called the leaf nodes (colored in green) and the rest of the nodes are called intermediate or internal nodes. The root and intermediate nodes represent the decisions while the leaf nodes represent the outcomes. ID3 stands for Iterative Dichotomiser 3 and is named such because the algorithm iteratively (repeatedly) dichotomizes(divides) features into two or more groups at each step. Invented by Ross Quinlan, ID3 uses a top-down greedy approach to build a decision tree. In simple words, the top-down approach means that we start building the tree from the top and the greedy approach means that at each iteration we select the best feature now to create a node. Most generally ID3 is only used for classification problems with nominal features only.	

Program	`# -*- coding: utf-8 -*-`
	`"""ID3.ipynb`
	
	`Automatically generated by Colaboratory.`	

```
Original file is located at

https://colab.research.google.com/drive/1J3Hpy2rS8LQ
WjUK7f42DE4NZNOWpMdLL
"""

from google.colab import files
uploaded = files.upload()

import pandas as pd
import numpy as np
import math
data = pd.read_csv('PlayTennis.csv')

def highlight(cell_value):
    color_1 = 'background-color: pink;'
    color_2 = 'background-color: lightgreen;'

    if cell_value == 'no':
        return color_1
    elif cell_value == 'yes':
        return color_2

data.style.applymap(highlight)\
    .set_properties(subset=data.columns, **{'width':
'100px'})\
    .set_table_styles([{'selector': 'th', 'props':
[('background-color', 'lightgray'), ('border', '1px
solid gray'),

('font-weight', 'bold')]},
     {'selector': 'tr:hover', 'props':
[('background-color', 'white'), ('border', '1.5px
solid black')]}])

def find_entropy(data):
    entropy = 0
    for i in range(data.nunique()):
        x = data.value_counts()[i]/data.shape[0]
        entropy += (- x * math.log(x,2))
    return round(entropy,3)

def information_gain(data, data_):
    info = 0
    for i in range(data_.nunique()):
        df = data[data_ == data_.unique()[i]]
        w_avg = df.shape[0]/data.shape[0]
        entropy = find_entropy(df.play)
        x = w_avg * entropy
        info += x
    ig = find_entropy(data.play) - info
    return round(ig, 3)
```

```
def entropy_and_infogain(datax, feature):
    for i in range(data[feature].nunique()):
        df =
datax[datax[feature]==data[feature].unique()[i]]
        if df.shape[0] < 1:
            continue

        display(df[[feature,
'play']].style.applymap(highlight)\
                .set_properties(subset=[feature,
'play'], **{'width': '80px'})\
                .set_table_styles([{'selector':
'th', 'props': [('background-color', 'lightgray'),

('border', '1px solid gray'),

('font-weight', 'bold')]},
                                   {'selector':
'td', 'props': [('border', '1px solid gray')]},
                                   {'selector':
'tr:hover', 'props': [('background-color', 'white'),

('border', '1.5px solid black')]}]))

        print(f'Entropy of {feature} -
{data[feature].unique()[i]} =
{find_entropy(df.play)}')
    print(f'Information Gain for {feature} =
{information_gain(datax, datax[feature])}')

"""**Info(D) for complete Dataset**

"""

print(f'Entropy of the entire dataset:
{find_entropy(data.play)}')

"""**Outlook**"""

entropy_and_infogain(data, 'outlook')

"""**Temp**"""

entropy_and_infogain(data, 'temp')

"""**Humidity**"""

entropy_and_infogain(data, 'humidity')

"""**Windy**"""

entropy_and_infogain(data, 'windy')

"""### **Outlook - Sunny**"""
```

```
sunny = data[data['outlook'] == 'sunny']
sunny.style.applymap(highlight)\
    .set_properties(subset=data.columns, **{'width':
'100px'})\
    .set_table_styles([{'selector': 'th', 'props':
[('background-color', 'lightgray'), ('border', '1px
solid gray'),

('font-weight', 'bold')]},
     {'selector': 'tr:hover', 'props':
[('background-color', 'white'), ('border', '1.5px
solid black')]}])

print(f'Entropy of the Sunny dataset:
{find_entropy(sunny.play)}')

"""**Calculating the Information gain for each
attribute**

---



**Temp**
"""

entropy_and_infogain(sunny, 'temp')

"""**Humidity**"""

entropy_and_infogain(sunny, 'humidity')

"""**Windy**"""

entropy_and_infogain(sunny, 'windy')

"""### **Outlook - Rainy**"""

rainy = data[data['outlook'] == 'rainy']
rainy.style.applymap(highlight)\
    .set_properties(subset=data.columns, **{'width':
'100px'})\
    .set_table_styles([{'selector': 'th', 'props':
[('background-color', 'lightgray'), ('border', '1px
solid gray'),

('font-weight', 'bold')]},
     {'selector': 'tr:hover', 'props':
[('background-color', 'white'), ('border', '1.5px
solid black')]}])

print(f'Entropy of the Rainy dataset:
{find_entropy(rainy.play)}')

"""**Calculating the Information gain for each
```

```
attribute**

**Temp**
"""

entropy_and_infogain(rainy, 'temp')

"""**Humidity**"""

entropy_and_infogain(rainy, 'humidity')

"""**Windy**"""

entropy_and_infogain(rainy, 'windy')
```

Output

	outlook	play
0	sunny	no
1	sunny	no
7	sunny	no
8	sunny	yes
10	sunny	yes

Entropy of outlook - sunny = 0.971

	outlook	play
2	overcast	yes
6	overcast	yes
11	overcast	yes
12	overcast	yes

Entropy of outlook - overcast = 0.0

	outlook	play
3	rainy	yes
4	rainy	yes
5	rainy	no
9	rainy	yes
13	rainy	no

Entropy of outlook - rainy = 0.971
Information Gain for outlook = 0.246

	humidity	play
0	high	no
1	high	no
2	high	yes
3	high	yes
7	high	no
11	high	yes
13	high	no

Entropy of humidity - high = 0.985

	humidity	play
4	normal	yes
5	normal	no
6	normal	yes
8	normal	yes
9	normal	yes
10	normal	yes
12	normal	yes

Entropy of humidity - normal = 0.592
Information Gain for humidity = 0.151

	windy	play
0	False	no
2	False	yes
3	False	yes
4	False	yes
7	False	no
8	False	yes
9	False	yes
12	False	yes

Entropy of windy - False = 0.811

	windy	play
1	True	no
5	True	no
6	True	yes
10	True	yes
11	True	yes
13	True	no

Entropy of windy - True = 1.0
Information Gain for windy = 0.048

	outlook	temp	humidity	windy	play
0	sunny	hot	high	False	no
1	sunny	hot	high	True	no
7	sunny	mild	high	False	no
8	sunny	cool	normal	False	yes
10	sunny	mild	normal	True	yes

	temp	play
0	hot	no
1	hot	no

Entropy of temp - hot = 0.0

	temp	play
7	mild	no
10	mild	yes

Entropy of temp - mild = 1.0

	temp	play
8	cool	yes

Entropy of temp - cool = 0.0
Information Gain for temp = 0.571

	humidity	play
0	high	no
1	high	no
7	high	no

```
Entropy of humidity - high = 0.0
```

	humidity	play
8	normal	yes
10	normal	yes

```
Entropy of humidity - normal = 0.0
Information Gain for humidity = 0.971
```

	windy	play
0	False	no
7	False	no
8	False	yes

```
Entropy of windy - False = 0.918
```

	windy	play
1	True	no
10	True	yes

```
Entropy of windy - True = 1.0
Information Gain for windy = 0.02
```

	outlook	temp	humidity	windy	play
3	rainy	mild	high	False	yes
4	rainy	cool	normal	False	yes
5	rainy	cool	normal	True	no
9	rainy	mild	normal	False	yes
13	rainy	mild	high	True	no

	temp	play
3	mild	yes
9	mild	yes
13	mild	no

```
Entropy of temp - mild = 0.918
```

	temp	play
4	cool	yes
5	cool	no

```
Entropy of temp - cool = 1.0
Information Gain for temp = 0.02
```

	humidity	play
3	high	yes
13	high	no

```
Entropy of humidity - high = 1.0
```

	humidity	play
4	normal	yes
5	normal	no
9	normal	yes

```
Entropy of humidity - normal = 0.918
Information Gain for humidity = 0.02
```

	windy	play
3	False	yes
4	False	yes
9	False	yes

```
Entropy of windy - False = 0.0
```

	windy	play
5	True	no
13	True	no

```
Entropy of windy - True = 0.0
Information Gain for windy = 0.971
```

Conclusion:	A random dataset was taken from Kaggle repository and ID3 algorithm was implemented on the dataset. It is identified that Windy attribute has highest Information Gain hence, it is the decision node.