

1. โจทย์ปัญหาการเล่นกอล์ฟ

จงเขียนโปรแกรมสร้าง Decision Tree ที่ใช้ในการตัดสินใจปัญหาการเล่นกอล์ฟ?

<i>Outlook</i>	<i>Temperature</i>	<i>Humidity</i>	<i>Windy</i>	<i>Play (positive) / Don't Play (negative)</i>
sunny	85	85	false	Don't Play
sunny	80	90	true	Don't Play
overcast	83	78	false	Play
rain	70	96	false	Play
rain	68	80	false	Play
rain	65	70	true	Don't Play
overcast	64	65	true	Play
sunny	72	95	false	Don't Play
sunny	69	70	false	Play
rain	75	80	false	Play
sunny	75	70	true	Play
overcast	72	90	true	Play
overcast	81	75	false	Play
rain	71	80	true	Don't Play

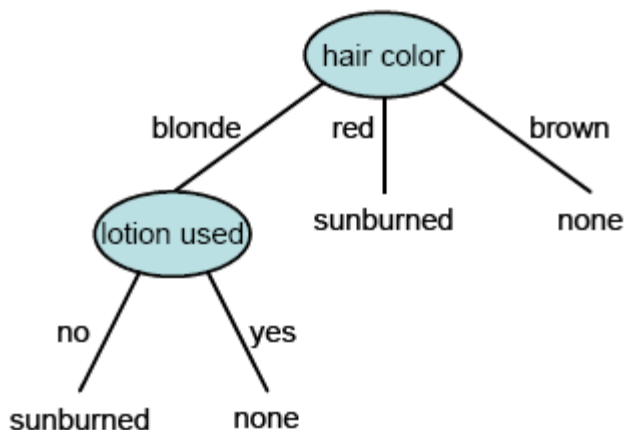
2.ทฤษฎี Decision Tree นำมาใช้แก้ปัญหาการเล่นกอล์ฟ

Decision Tree

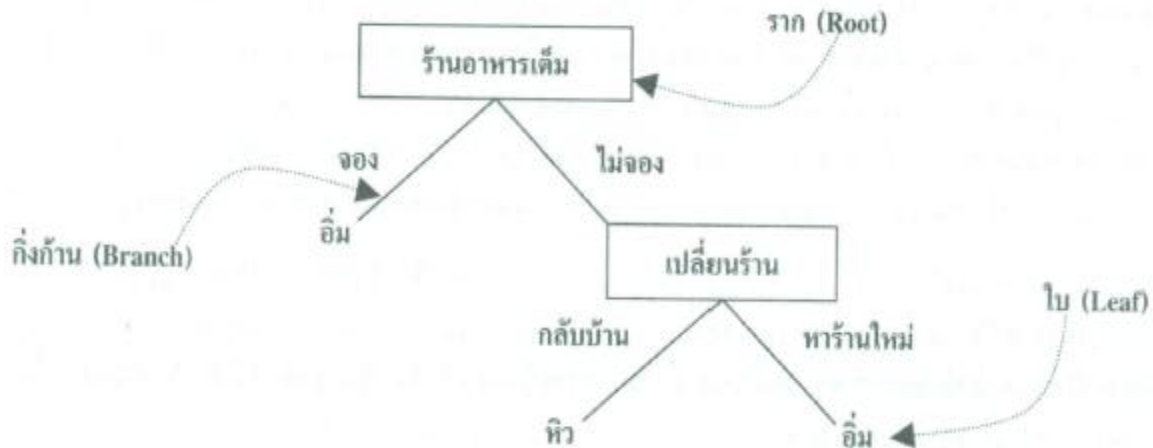
- Decision Tree เป็นเทคนิคการเรียนรู้แบบมีผู้สอน ที่มีลักษณะเป็นแบบจำลองทางคณิตศาสตร์ที่ใช้ทำนายหรือจำแนกประเภทของวัตถุหรือเหตุการณ์โดยพิจารณาจากลักษณะของวัตถุหรือเหตุการณ์
- Decision Tree เป็นอัลกอริทึมการเรียนรู้ที่มีลักษณะไม่ค่อยซับซ้อนมาก โดยจะมีการแตกแขนงจาก โหนดราก (Root) สู่มิ (Leaf) และมีกิ่งก้าน (Branch) แตกออกไปตามเงื่อนไขหรือข้อมูลที่ได้คาดคะเนไว้ว่าจะเกิดขึ้น เพื่อให้ทราบถึงผลลัพธ์ของแต่ละเหตุการณ์
- เช่น คุณหิวข้าวไหม คำตอบสามารถเป็นไปได้ทั้ง หิว และไม่หิว เป็นต้น

โครงสร้าง Decision Tree

- Decision Tree จะทำการจัดกลุ่มชุดข้อมูลประกอบด้วย
 - โหนดหรือบัพ (node) เป็นส่วนที่แสดงคุณสมบัติหรือตัวแปร (attribute) ของชุดข้อมูล
 - กิ่ง (link) ที่ต่อกับโหนด เป็นส่วนที่แสดงค่าของคุณสมบัติหรือตัวแปร(attribute)
 - โหนดที่ปลายสุดเรียกว่าโหนดใบ (leaf node) หรือเรียกย่อๆว่าใบ (leaf) เป็นส่วนที่แสดงประเภทของวัตถุหรือเหตุการณ์
- ตัวอย่างต้นไม้ตัดสินใจแสดงดังรูป



- ตัวอย่างแบบจำลองของ Decision Tree ที่แสดงถึงเหตุการณ์ร้านอาหารเต็มแล้วจะตัดสินใจอย่างไร แสดงดังรูป



- จากรูปเมื่อเกิดเหตุการณ์ร้านอาหารเต็ม เราจะจองหรือไม่ ถ้าตัดสินใจว่าไม่จอง เหตุการณ์ที่เกิดขึ้นก็จะแตกต่างกันไป เช่น เดินทางกลับบ้าน หรือเดินทางไปหาร้านอาหารใหม่ เป็นต้น โดยจะแยกแยะสิ่งที่เราสนใจได้ 3 อย่าง คือ จำนวนลูกค้า ระยะเวลาในการรอ และการจองโต๊ะ

ตัวอย่าง Decision Tree

- การสร้าง Decision Tree ทำได้โดยสร้างโหนดทีละโหนดเพื่อตรวจสอบคุณสมบัติของตัวอย่าง แล้วแยกตัวอย่างลงตามค่าของกิ่ง ทำจนกระทั่งตัวอย่างในใบแต่ละใบอยู่ในประเภทเดียวกันทั้งหมด
- ปัญหาการฟุ้งแฉด
 - เราไปเที่ยวที่ชายทะเลและพบว่าคนที่ไปฟุ้งแฉดตามชายทะเล บางคนก็จะมีผิวเปลี่ยนเป็นสีแทน แต่บางคนต้องได้รับความทรมานจากผิวไหม้ เราต้องการหาว่าอะไรคือปัจจัยที่ทำให้คนที่ไปฟุ้งแฉดตามชายทะเลแล้วผิวไหม้หรือไม่ไหม้ โดยข้อมูลที่สังเกตได้ประกอบด้วย ความแตกต่างของสีผิมน้ำหนัก ส่วนสูงของผู้ที่ไปฟุ้งแฉด และการใช้โลชั่น ซึ่งบางคนก็ใช้โลชั่น บางคนก็ไม่ใช้ ดังรูป

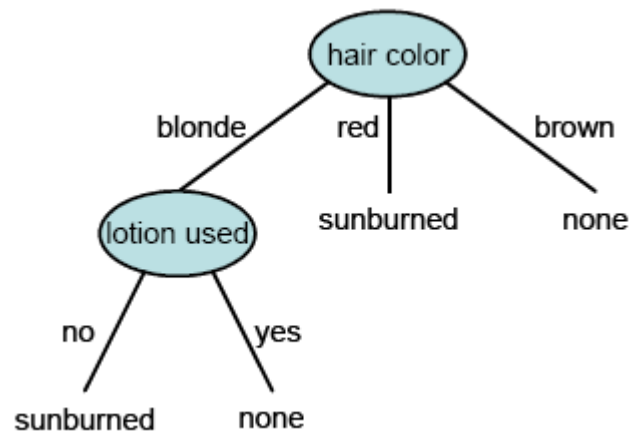
ประเภท
↓

คุณสมบัติ →

Name	Hair	Height	Weight	Lotion	Result
Sarah	blonde	average	light	no	sunburned
Dana	blonde	tall	average	yes	none
Alex	brown	short	average	yes	none
Annie	blonde	short	average	no	sunburned
Emily	red	average	heavy	no	sunburned
Pete	brown	tall	heavy	no	none
John	brown	average	heavy	no	none
Katie	blonde	short	light	Yes	none

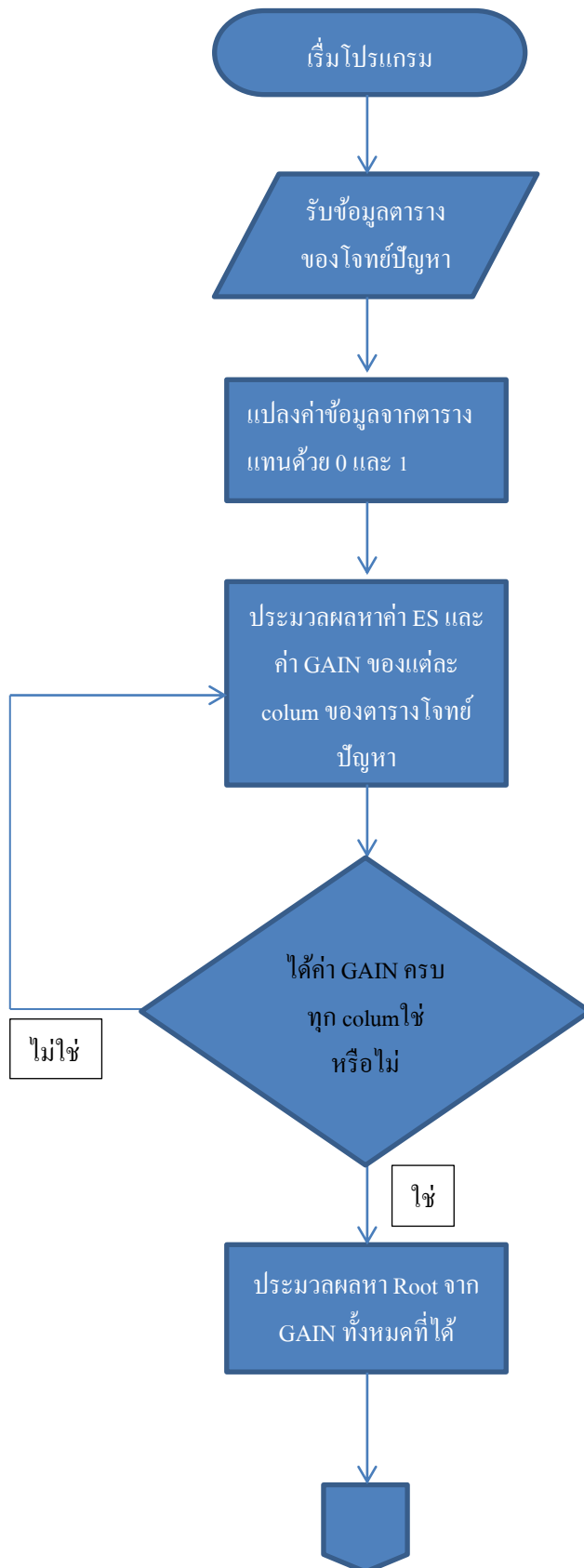
ค่า {

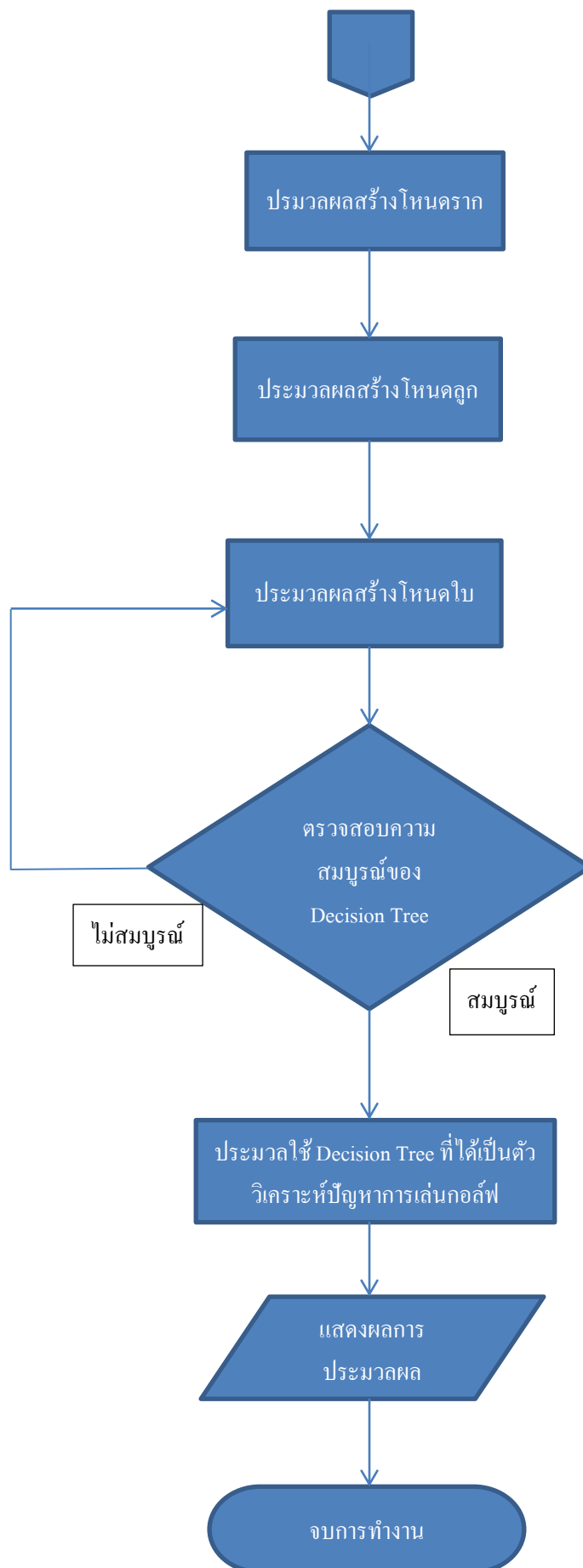
- การเรียนรู้ต้นไม้ตัดสินใจจะทำการวางนัยทั่วไปของข้อมูลโดยสร้างเป็นโมเดลอยู่ในรูปต้นไม้ตัดสินใจ



- สมมติว่าเราเลือกคุณสมบัติ hair color เป็นโหนดแรกหรือโหนดรากของต้นไม้ เราจะแยกตัวอย่างลงตามกิ่งของโหนด hair color ตัวอย่างใดที่มีค่าของ hair color เป็น blonde ก็แยกลงตามกิ่งซ้าย ถ้าเป็น red ก็แยกลงตามกิ่งกลาง และถ้าเป็น brown ก็แยกลงตามกิ่งขวา
 - เครื่องหมาย + และ - แสดงประเภท sunburned และ none ตามลำดับดังรูป

3.โฟลว์ชาร์ตกระบวนการทำงานของโปรแกรม





4. โค้ดโปรแกรมแก้ปัญหาการเล่นกอล์ฟ

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Collections;

namespace Golf_Decision_Tree
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        //variable
        DataTable data = new DataTable();
        string[] outlook;
        string[] temperature;
        string[] humidity;
        string[] windy;
        string[] play;
        int row;

        private void Form1_Load(object sender, EventArgs e)
        {
            data.Columns.Add("Outlook");
            data.Columns.Add("Temperature");
            data.Columns.Add("Humidity");
            data.Columns.Add("Windy");
            data.Columns.Add("Play");

            data.Rows.Add("sunny", "85", "85", "fale", "Don't Play");
            data.Rows.Add("sunny", "80", "90", "true", "Don't Play");
            data.Rows.Add("overcast", "83", "78", "fale", "Play");
            data.Rows.Add("rain", "70", "96", "fale", "Play");
            data.Rows.Add("rain", "68", "80", "fale", "Play");
        }
    }
}
```

```
data.Rows.Add("rain", "65", "70", "true", "Don't Play");
data.Rows.Add("overcast", "64", "65", "true", "Play");
data.Rows.Add("sunny", "72", "95", "fale", "Don't Play");
data.Rows.Add("sunny", "69", "70", "fale", "Play");
data.Rows.Add("rain", "75", "80", "fale", "Play");
data.Rows.Add("sunny", "75", "70", "true", "Play");
data.Rows.Add("overcast", "72", "90", "fale", "Play");
data.Rows.Add("overcast", "81", "75", "fale", "Play");
data.Rows.Add("rain", "71", "80", "true", "Don't Play");
```

```
dataGridView1.DataSource = data;
```

```
groupBox4.Enabled = false;
```

```
panel1.Visible = false;
```

```
panel2.Visible = false;
```

```
panel3.Visible = false;
```

```
label6.Visible = false;
```

```
label7.Visible = false;
```

```
label8.Visible = false;
```

```
label9.Visible = false;
```

```
label10.Visible = false;
```

```
label11.Visible = false;
```

```
label12.Visible = false;
```

```
textBox4.Visible = false;
```

```
textBox5.Visible = false;
```

```
textBox6.Visible = false;
```

```
textBox7.Visible = false;
```

```
textBox8.Visible = false;
```

```
textBox9.Visible = false;
```

```
textBox10.Visible = false;
```

```
lineShape1.Visible = false;
```

```
lineShape2.Visible = false;
```

```
lineShape3.Visible = false;
```

```
lineShape4.Visible = false;
```

```
lineShape5.Visible = false;
```

```
lineShape6.Visible = false;
```

```
lineShape7.Visible = false;
```

```
}
```



```
//build tree
private void button2_Click(object sender, EventArgs e)
{
    panel1.Visible = true;
    panel3.Visible = true;

    row = dataGridView1.Rows.Count;

    outlook = new string[row];
    temperature = new string[row];
    humidity = new string[row];
    windy = new string[row];
    play = new string[row];

    for (int i = 0; i < row; i++)
    {
        outlook[i] = Convert.ToString(dataGridView1.Rows[i].Cells[0].Value);
        temperature[i] = Convert.ToString(dataGridView1.Rows[i].Cells[1].Value);
        humidity[i] = Convert.ToString(dataGridView1.Rows[i].Cells[2].Value);
        windy[i] = Convert.ToString(dataGridView1.Rows[i].Cells[3].Value);
        play[i] = Convert.ToString(dataGridView1.Rows[i].Cells[4].Value);
    }

    textBox2.Text = "Calculate root";
    timer1.Enabled = true;
}

//check
private void button1_Click(object sender, EventArgs e)
{
    if (Outlook.SelectedIndex == 0)
    {
        if (int.Parse(Humidity.Text.ToString()) <= 70)
        {
            ans.Text = "Play";
            ans.BackColor = Color.Green;
        }
        else if (int.Parse(Humidity.Text.ToString()) > 70)
        {
            ans.Text = "Don't Play";
            ans.BackColor = Color.Red;
        }
    }
}
```

```
else if (Outlook.SelectedIndex == 1)
{
    ans.Text = "Play";
    ans.BackColor = Color.Green;
}
else if (Outlook.SelectedIndex == 2)
{
    if (Windy.SelectedIndex == 1)
    {
        ans.Text = "Play";
        ans.BackColor = Color.Green;
    }
    else if (Windy.SelectedIndex == 0)
    {
        ans.Text = "Don't Play";
        ans.BackColor = Color.Red;
    }
}
}

private void timer1_Tick(object sender, EventArgs e)
{
    timer1.Enabled=false;

    double a1, b1, c1, d1, e1;
    a1 = entropy();
    b1 = outlook_gain();
    c1 = temperature_gain();
    d1 = humidity_gain();
    e1 = windy_gain();

    textBox2.Text += "\r\n outlook_gain  = " + b1;
    textBox2.Text += "\r\n temperature_gain = " + c1;
    textBox2.Text += "\r\n humidity_gain  = " + d1;
    textBox2.Text += "\r\n windy_gain    = " + e1;

    if (b1 > c1 && b1 > d1 && b1 > e1)
    {
        textBox2.Text += "\r\nRoot = outlook";
        textBox3.Text = "Outlook";
    }
    else if (c1 > b1 && c1 > d1 && c1 > e1)
```

```

{
    textBox2.Text += "\r\nRoot = temperature";
    textBox3.Text = "Temperature";
}
else if(d1 > b1 && d1 > c1 && d > e1)
{
    textBox2.Text += "\r\nRoot = humidity";
    textBox3.Text = "Humidity";
}
else if (e1 > b1 && e1 > c1 && e1 > d1)
{
    textBox2.Text += "\r\nRoot = windy";
    textBox3.Text = "Windy";
}

label6.Visible = true;
label7.Visible = true;
label8.Visible = true;
textBox4.Visible = true;
textBox5.Visible = true;
textBox6.Visible = true;
lineShape1.Visible = true;
lineShape2.Visible = true;
lineShape7.Visible = true;

textBox6.Text = "Play";
textBox2.Text += "\r\nCalculate subroot";

timer2.Enabled = true;
}

private void timer2_Tick(object sender, EventArgs e)
{
    timer2.Enabled = false;

    double a2, b2, c2, d2;
    a2 = entropy2();
    b2 = temperature_gain2();
    c2 = humidity_gain2();
    d2 = windy_gain2();

    textBox2.Text += "\r\n temperature_gain = " + b2;

```

```

textBox2.Text += "\r\n humidity_gain  = " + c2;
textBox2.Text += "\r\n windy_gain    = " + d2;

if (b2 > c2 && b2 > d2 )
{
    textBox2.Text += "\r\nRoot = temperature";
    textBox4.Text = "Temperature";
}
else if (c2 > d2 && c2 > b2)
{
    textBox2.Text += "\r\nRoot = humidity";
    textBox4.Text = "Humidity";
}
else if (d2 > b2 && d2 > c2)
{
    textBox2.Text += "\r\nRoot = windy";
    textBox4.Text = "Windy";
}

label9.Visible = true;
label10.Visible = true;
textBox7.Visible = true;
textBox8.Visible = true;
lineShape3.Visible = true;
lineShape4.Visible = true;

textBox7.Text = "Play";
textBox8.Text = "Don't play";
textBox2.Text += "\r\nCalculate subroot";

timer3.Enabled = true;
}

private void timer3_Tick(object sender, EventArgs e)
{
    timer3.Enabled = false;

    double a3, b3, c3, d3, e3;
    a3 = entropy3();
    b3 = temperature_gain3();
    c3 = windy_gain3();

```

```

textBox2.Text += "\r\n temperature_gain = " + b3;
textBox2.Text += "\r\n windy_gain    = " + c3;

```

```

if (b3 > c3)
{
    textBox2.Text += "\r\nRoot = temperature";
    textBox5.Text = "temperature";
}
else if (c3 > b3)
{
    textBox2.Text += "\r\nRoot = windy";
    textBox5.Text = "Windy";
}

```

```

label11.Visible = true;
label12.Visible = true;
textBox9.Visible = true;
textBox10.Visible = true;
lineShape5.Visible = true;
lineShape6.Visible = true;

```

```

textBox9.Text = "Play";
textBox10.Text = "Don't play";

```

```

panel2.Visible = true;
groupBox4.Enabled = true;
}

```

```
//หาค่าเอนโทรปี
```

```

double e;
double p, d;
double entropy()
{
    for (int i = 0; i < row; i++)
    {
        if (play[i] == "Play")
        {
            p++;
        }
        else if (play[i] == "Don't Play")
        {
            d++;
        }
    }
}

```

```

    }
}
e = ((-p / row) * ((Math.Log10(p / row)) / Math.Log10(2))) + ((-d / row) * ((Math.Log10(d / row)) / Math.Log10(2)));
return e;
}

```

//หาค่าเกินของoutlook

```

double s, o, r, goutlook;
double sp, op, rp, sd, od, rd;
double outlook_gain()
{
    for (int i = 0; i < row; i++)
    {
        if (outlook[i] == "sunny")
        {
            s++;
            if (play[i] == "Play")
            {
                sp++;
            }
            else if (play[i] == "Don't Play")
            {
                sd++;
            }
        }
        else if (outlook[i] == "overcast")
        {
            o++;
            if (play[i] == "Play")
            {
                op++;
            }
            else if (play[i] == "Don't Play")
            {
                od++;
            }
        }
        else if (outlook[i] == "rain")
        {
            r++;
            if (play[i] == "Play")
            {

```

```

        rp++;
    }
    else if (play[i] == "Don't Play")
    {
        rd++;
    }
}
}

goutlook = e - ((s / row * (((-sp / s) * ((Math.Log10(sp / s)) / Math.Log10(2))) + ((-sd / s) * ((Math.Log10(sd / s)) / Math.Log10(2)))))

+
(o / row * (((-op / o) * ((Math.Log10(op / o)) / Math.Log10(2))))) +
(r / row * (((-rp / r) * ((Math.Log10(rp / r)) / Math.Log10(2))) + ((-rd / r) * ((Math.Log10(rd / r)) / Math.Log10(2)))));
return goutlook;
}

//หาค่าเกณฑ์ของtemperature
double t1, t2, gtemperature;
double t1p, t2p, t1d, t2d;
double temperature_gain()
{
    for (int i = 0; i < row; i++)
    {
        //temperature
        if (int.Parse(temperature[i]) <= 70)
        {
            t1++;
            if (play[i] == "Play")
            {
                t1p++;
            }
            else if (play[i] == "Don't Play")
            {
                t1d++;
            }
        }
        else if (int.Parse(temperature[i]) > 70)
        {
            t2++;
            if (play[i] == "Play")
            {
                t2p++;
            }
        }
    }
}

```

```

        else if (play[i] == "Don't Play")
        {
            t2d++;
        }
    }

    gtemperature = e - ((t1 / row * (((-t1p / t1) * ((Math.Log10(t1p / t1)) / Math.Log10(2))) + ((-t1d / t1) * ((Math.Log10(t1d / t1)) /
Math.Log10(2)))))) +
        (t2 / row * (((-t2p / t2) * ((Math.Log10(t2p / t2)) / Math.Log10(2))) + ((-t2d / t2) * ((Math.Log10(t2d / t2)) / Math.Log10(2))))));
    return gtemperature;
}

//หาค่าแกนของhumidity
double h1, h2, ghumidity;
double h1p, h2p, h1d, h2d;
double humidity_gain()
{
    for (int i = 0; i < row; i++)
    {
        //humidity
        if (int.Parse(humidity[i]) <= 70)
        {
            h1++;
            if (play[i] == "Play")
            {
                h1p++;
            }
            else if (play[i] == "Don't Play")
            {
                h1d++;
            }
        }
        else if (int.Parse(humidity[i]) > 70)
        {
            h2++;
            if (play[i] == "Play")
            {
                h2p++;
            }
            else if (play[i] == "Don't Play")
            {
                h2d++;
            }
        }
    }
}

```



```

    }
    }
}

ghumidity = e - ((h1 / row * (((-h1p / t1) * ((Math.Log10(h1p / h1)) / Math.Log10(2))) + ((-h1d / h1) * ((Math.Log10(h1d / h1)) /
Math.Log10(2))))) +
    (h2 / row * (((-h2p / h2) * ((Math.Log10(h2p / h2)) / Math.Log10(2))) + ((-h2d / h2) * ((Math.Log10(h2d / h2)) /
Math.Log10(2))))) );
return ghumidity;
}

//หาค่าเกณฑ์ของwindy
double t, f, gwindy;
double tp, fp, td, fd;
double windy_gain()
{
    for (int i = 0; i < row; i++)
    {
        //windy
        if (windy[i] == "true")
        {
            t++;
            if (play[i] == "Play")
            {
                tp++;
            }
            else if (play[i] == "Don't Play")
            {
                td++;
            }
        }
        else if (windy[i] == "fale")
        {
            f++;
            if (play[i] == "Play")
            {
                fp++;
            }
            else if (play[i] == "Don't Play")
            {
                fd++;
            }
        }
    }
}

```

```

    }
    windy = e - ((t / row * (((-tp / t) * ((Math.Log10(tp / t)) / Math.Log10(2)))) + ((-td / t) * ((Math.Log10(td / t)) / Math.Log10(2)))) +
        (f / row * (((-fp / f) * ((Math.Log10(fp / f)) / Math.Log10(2)))) + ((-fd / f) * ((Math.Log10(fd / f)) / Math.Log10(2)))));
    return windy;
}

```

//หาค่าเอนโทรปี2

```

double e2, c2;
double p2, d2;
double entropy2()
{
    for (int i = 0; i < row; i++)
    {
        if (outlook[i] == "sunny")
        {
            c2++;
            if (play[i] == "Play")
            {
                p2++;
            }
            else if (play[i] == "Don't Play")
            {
                d2++;
            }
        }
    }
    e2 = ((-p2 / c2) * ((Math.Log10(p2 / c2)) / Math.Log10(2))) + ((-d2 / c2) * ((Math.Log10(d2 / c2)) / Math.Log10(2)));
    return e2;
}

```

//หาค่าเอนโทรปีของtemperature2

```

double t12, t22, gtemperature2;
double t1p2, t2p2, t1d2, t2d2;
double temperature_gain2()
{
    for (int i = 0; i < row; i++)
    {
        if (outlook[i] == "sunny")
        {
            //temperature
            if (int.Parse(temperature[i]) <= 70)
            {

```

```

        t12++;
        if (play[i] == "Play")
        {
            t1p2++;
        }
        else if (play[i] == "Don't Play")
        {
            t1d2++;
        }
    }
    else if (int.Parse(temperature[i]) > 70)
    {
        t22++;
        if (play[i] == "Play")
        {
            t2p2++;
        }
        else if (play[i] == "Don't Play")
        {
            t2d2++;
        }
    }
}

gtemperature2 = e2 - ((t12 / c2 * (((-t1p2 / t12) * ((Math.Log10(t1p2 / t12)) / Math.Log10(2)))))) +
    (t22 / c2 * (((-t2p2 / t22) * ((Math.Log10(t2p2 / t22)) / Math.Log10(2))) + ((-t2d2 / t22) * ((Math.Log10(t2d2 / t22)) /
Math.Log10(2))))));
return gtemperature2;
}

//หาค่าเอนของhumidity2
double h12, h22, ghumidity2;
double h1p2, h2p2, h1d2, h2d2;
double humidity_gain2()
{
    for (int i = 0; i < row; i++)
    {
        if (outlook[i] == "sunny")
        {
            //humidity
            if (int.Parse(humidity[i]) <= 70)
            {

```

```

        h12++;
        if (play[i] == "Play")
        {
            h1p2++;
        }
        else if (play[i] == "Don't Play")
        {
            h1d2++;
        }
    }
    else if (int.Parse(humidity[i]) > 70)
    {
        h22++;
        if (play[i] == "Play")
        {
            h2p2++;
        }
        else if (play[i] == "Don't Play")
        {
            h2d2++;
        }
    }
}

ghumidity2 = e2 - ((h12 / c2 * (((-h1p2 / h12) * ((Math.Log10(h1p2 / h12)) / Math.Log10(2)))))) +
    (h22 / c2 * (((-h2d2 / h22) * ((Math.Log10(h2d2 / h22)) / Math.Log10(2))))));

return ghumidity2;
}

```

//หาค่าเกณฑ์ของwindy2

```

double tt2, ff2, gwindy2;
double tp2, fp2, td2, fd2;
double windy_gain2()
{
    for (int i = 0; i < row; i++)
    {
        if (outlook[i] == "sunny")
        {
            //windy
            if (windy[i] == "true")
            {
                tt2++;
            }
        }
    }
}

```

```

        if (play[i] == "Play")
        {
            tp2++;
        }
        else if (play[i] == "Don't Play")
        {
            td2++;
        }
    }
    else if (windy[i] == "fale")
    {
        ff2++;
        if (play[i] == "Play")
        {
            fp2++;
        }
        else if (play[i] == "Don't Play")
        {
            fd2++;
        }
    }
}

gwindy2 = e2 - ((tt2 / c2 * (((-tp2 / tt2) * ((Math.Log10(tp2 / tt2)) / Math.Log10(2)))) + ((-td2 / tt2) * ((Math.Log10(td2 / tt2)) /
Math.Log10(2)))))) +
    ((ff2 / c2 * (((-fp2 / ff2) * ((Math.Log10(fp2 / ff2)) / Math.Log10(2))) + ((-fd2 / ff2) * ((Math.Log10(fd2 / ff2)) /
Math.Log10(2))))));
return gwindy2;
}

//หาค่าเอนโทรปี3
double e3, c3;
double p3, d3;
double entropy3()
{
    for (int i = 0; i < row; i++)
    {
        if (outlook[i] == "rain")
        {
            c3++;
            if (play[i] == "Play")
            {

```

```

        p3++;
    }
    else if (play[i] == "Don't Play")
    {
        d3++;
    }
}
}

e3 = ((-p3 / c3) * ((Math.Log10(p3 / c3)) / Math.Log10(2))) + ((-d3 / c3) * ((Math.Log10(d3 / c3)) / Math.Log10(2)));
return e3;
}

//หาค่าแกนของtemperature3
double t13, t23, gtemperature3;
double t1p3, t2p3, t1d3, t2d3;
double temperature_gain3()
{
    for (int i = 0; i < row; i++)
    {
        if (outlook[i] == "rain")
        {
            //temperature
            if (int.Parse(temperature[i]) <= 70)
            {
                t13++;
                if (play[i] == "Play")
                {
                    t1p3++;
                }
            }
            else if (play[i] == "Don't Play")
            {
                t1d3++;
            }
        }
        else if (int.Parse(temperature[i]) > 70)
        {
            t23++;
            if (play[i] == "Play")
            {
                t2p3++;
            }
            else if (play[i] == "Don't Play")

```

```

        {
            t2d3++;
        }
    }
}

gtemperature3 = e3 - ((t13 / c3 * (((-t1p3 / t13) * ((Math.Log10(t1p3 / t13)) / Math.Log10(2)))) + ((-t1d3 / t13) * ((Math.Log10(t1d3 / t13)) / Math.Log10(2)))))) +
    ((t23 / c3 * (((-t2p3 / t23) * ((Math.Log10(t2p3 / t23)) / Math.Log10(2)))) + ((-t2d3 / t23) * ((Math.Log10(t2d3 / t23)) / Math.Log10(2))))));
return gtemperature3;
}

//หาค่าเกณฑ์ของwindy3
double tt3, ff3, gwindy3;
double tp3, fp3, td3, fd3;
double windy_gain3()
{
    for (int i = 0; i < row; i++)
    {
        if (outlook[i] == "rain")
        {
            //windy
            if (windy[i] == "true")
            {
                tt3++;
                if (play[i] == "Play")
                {
                    tp3++;
                }
                else if (play[i] == "Don't Play")
                {
                    td3++;
                }
            }
            else if (windy[i] == "fale")
            {
                ff3++;
                if (play[i] == "Play")
                {
                    fp3++;
                }
            }
        }
    }
}

```

```
        else if (play[i] == "Don't Play")
        {
            fd3++;
        }
    }
}

gwindy3 = e3 - ((tt3 / c3 * (((-td3 / tt3) * ((Math.Log10(td3 / tt3)) / Math.Log10(2)))))) +
    (ff3 / c3 * (((-fp3 / ff3) * ((Math.Log10(fp3 / ff3)) / Math.Log10(2))))));
return gwindy3;
}
}
```


5.ผลการรันโปรแกรมแก้ปัญหาการเล่นกอล์ฟ

Decision Tree : Play Golf

Control

Build tree

Check

Outlook : sunny Humidity : 85

Temperature : 85 Windy : false Golf : **Don't Play**

DATA

	Outlook	Temperature	Humidity	Windy	Play
►	sunny	85	85	false	Don't Play
	sunny	80	90	true	Don't Play
	overcast	83	78	false	Play
	rain	70	96	false	Play
	rain	68	80	false	Play
	rain	65	70	true	Don't Play
	overcast	64	65	true	Play
	sunny	72	95	false	Don't Play
	sunny	69	70	false	Play
	rain	75	80	false	Play
	sunny	75	70	true	Play
	overcast	72	90	false	Play
	overcast	81	75	false	Play
	rain	71	80	true	Don't Play

Table

```

graph TD
    Outlook[Outlook] -- Sunny --> Humidity[Humidity]
    Outlook -- Overcast --> Play1[Play]
    Outlook -- Rain --> Windy[Windy]
    Humidity -- "Humidity <= 70" --> Play2[Play]
    Humidity -- "Humidity > 70" --> DontPlay1[Don't play]
    Windy -- False --> Play3[Play]
    Windy -- True --> DontPlay2[Don't play]
  
```

Rule

1. IF Sunny AND Humidity <= 70 THEN Play golf
2. IF Sunny AND Humidity > 70 THEN Don't Play golf
3. IF Overcast THEN Play golf
4. IF Rain AND Windy = true THEN Don't Play golf
5. IF Rain AND Windy = false THEN Play golf

Gain

Calculate root

outlook_gain = 0.246749819774439

temperature_gain = 0.0453341720291442

humidity_gain = 0.0327433913266373

windy_gain = 0.10224356360985

Root = outlook

Calculate subroot

temperature_gain = 0.321928094887362

humidity_gain = 0.970950594454669