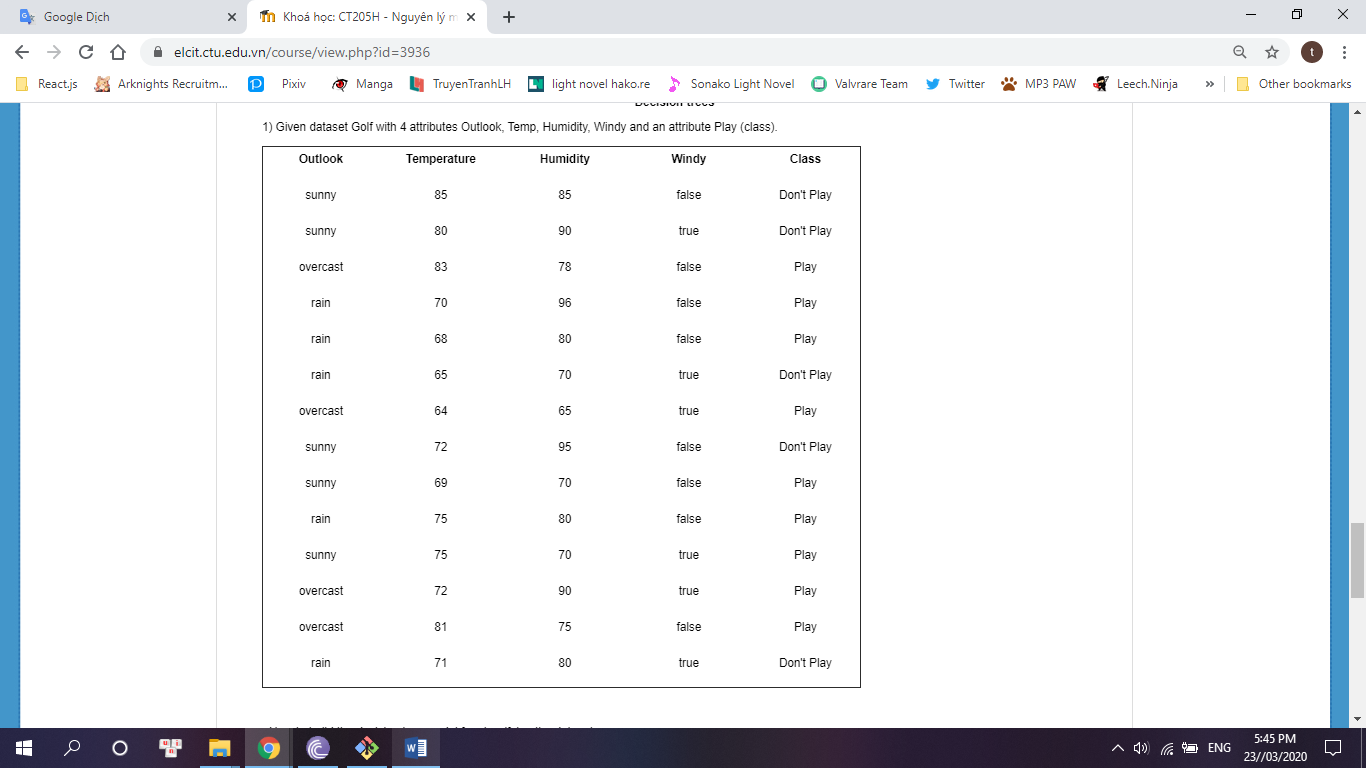
Decision Tree

Student name: Tran Chien Thanh

Student code: B1605365

1. **Given dataset Golf with 4 attributes Outlook, Temp, Humidity, Windy and an attribute Play (class).**



**- How to build the decision tree model for classifying the dataset**

**OutLook**

|  |  |  |  |
| --- | --- | --- | --- |
| OutLook | Play | Don’t Play | Total |
| Sunny | 2 | 3 | 5 |
| Overcast | 4 | 0 | 4 |
| Rain | 3 | 2 | 5 |

* Sunny:

Info([2,3]) = entropy(2/5, 3/5) = -2/5log(2/5) – 3/5log(3/5) = 0.971 bits

* Overcast:

Info([4,0]) = entropy(4/4, 0/4) = -4/4log(4/4) – 0/4log(0/4) = 0 bits

* Rain:

Info([3,2]) = entropy(3/5, 2/5) = -3/5log(3/5) – 2/5log(2/5) = 0.971 bits

* OutLook information:

Infor( [2,3], [4,0] [3,2] ) = (5/14) \* 0.971 + (4/14) \* 0 + (5/14) \* 0.971 = 0.693 bits

=> Gain(“OutLook”) = infor([9,5]) - Infor( [2,3], [4,0] [3,2] ) = 0.94 – 0.693 = 0.247 bits

**Windy**

|  |  |  |  |
| --- | --- | --- | --- |
| Windy | Play | Don’t Play | Total |
| True | 3 | 3 | 6 |
| False | 6 | 2 | 8 |

* True

Info([3,3]) = entropy(3/6, 3/6) = -3/6log(3/6) – 3/6log(3/6) = 1 bits

* False

Info([6,2]) = entropy(6/8, 2/8) = -6/8log(6/8) – 2/8log(2/8) = 0.811 bits

* Windy information

Info( [3,3], [6,2] ) = (6/14) \* 1 + (8/14) \* 0.811 = 0.892 bits

=> Gain(“Windy”) = infor(9/5) - Infor( [3,3], [6,2] ) = 0.94 – 0.892 = 0.048 bits

**Temperature**

* Temp <= 64: play/1 , don’t play/0
* Temp > 64: play/8 , don’t play/5

Info( [1,0], [8, 5] ) = 1/14info( [1,0] ) + 13/14info( [8,5] ) = 1/14 \* 0 + 13/14 \* 0.961 = 0.892

Gain(Temp = 64) = infor(9/5) - Info( [1,0], [8, 5] ) = 0.94 – 0.892

I will do the same calculate for the rest, the result is displayed below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Temperature | <= | | > | | Gain |
| Yes | No | Yes | No |
| 64 | 1 | 0 | 8 | 5 | 0.048 |
| 65 | 1 | 1 | 8 | 4 | 0.010 |
| 68 | 2 | 1 | 7 | 4 | 0.000 |
| 69 | 3 | 1 | 6 | 4 | 0.015 |
| 70 | 4 | 1 | 5 | 4 | 0.045 |
| 71 | 4 | 2 | 5 | 3 | 0.001 |
| 72 | 5 | 3 | 4 | 2 | 0.001 |
| 75 | 7 | 3 | 2 | 2 | 0.025 |
| 80 | 7 | 4 | 2 | 1 | 0.000 |
| 81 | 8 | 4 | 1 | 1 | 0.010 |
| 83 | 9 | 4 | 0 | 1 | 0.114 |

With temperature = 83, Gain value is max

**Humidity**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Humidity | <= | | > | | Gain |
| Yes | No | Yes | No |
| 65 | 1 | 0 | 8 | 5 | 0.048 |
| 70 | 3 | 1 | 6 | 4 | 0.015 |
| 75 | 4 | 1 | 5 | 4 | 0.045 |
| 78 | 5 | 1 | 4 | 4 | 0.090 |
| 80 | 7 | 2 | 2 | 3 | 0.102 |
| 85 | 7 | 3 | 2 | 2 | 0.025 |
| 90 | 8 | 4 | 1 | 1 | 0.010 |
| 95 | 8 | 5 | 1 | 0 | 0.048 |

With humidity = 80, Gain value is max

After all the caculation above, I have the table:

|  |  |
| --- | --- |
|  | Gain |
| Outlook | 0.247 |
| Windy | 0.048 |
| Temperature | 0.114 |
| Humidity | 0.102 |

Because Outlook gain is the biggest, the root node will be “Outlook”

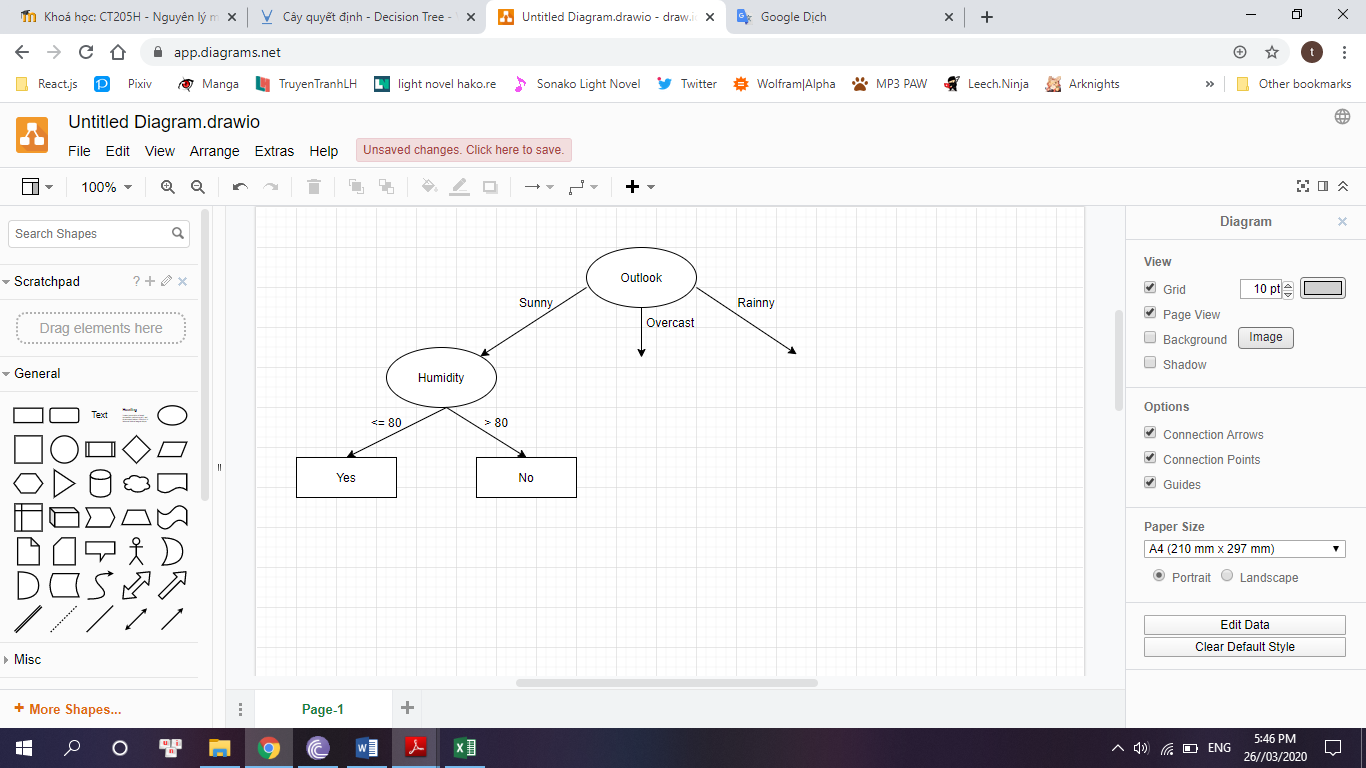
Next, summary all possible case with oulook:

- With outlook = ‘sunny’

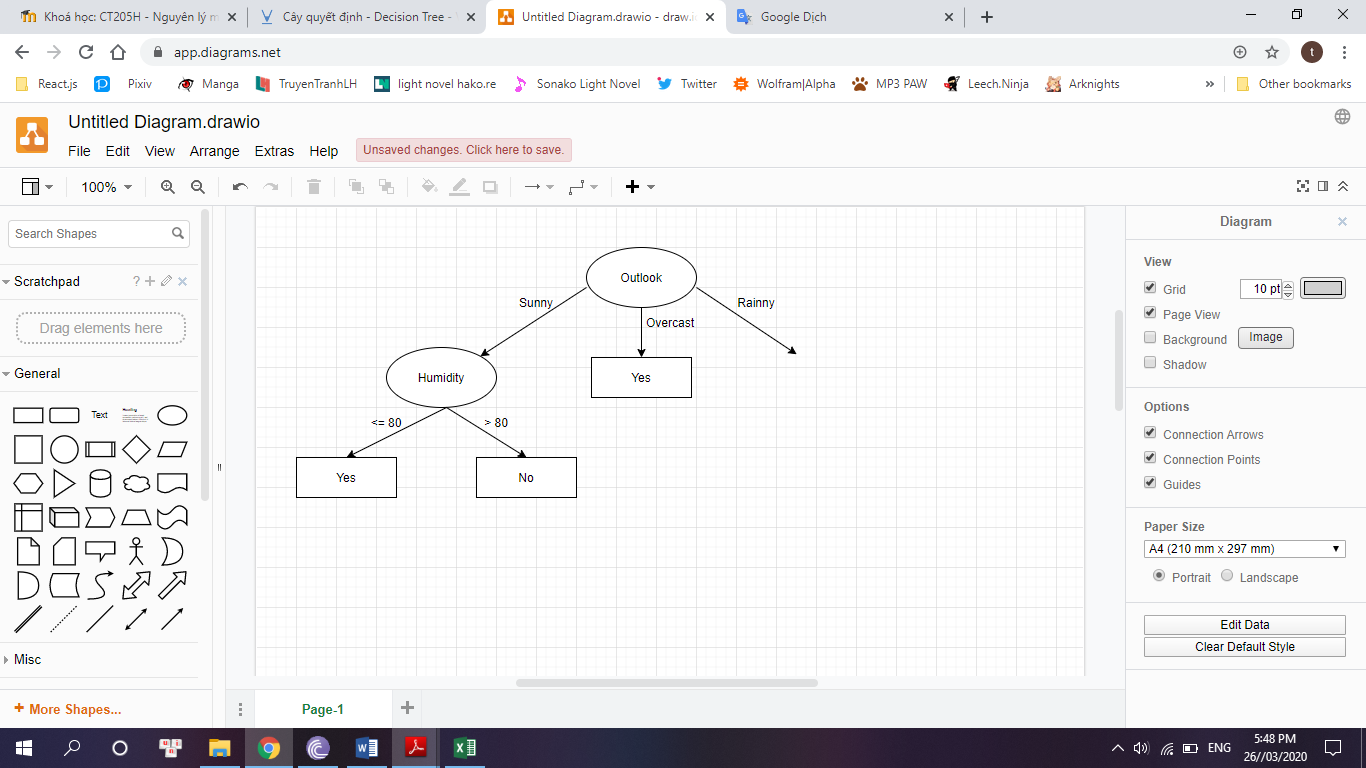
+ If Humidity <= 80, then class = ‘Yes’

+ if Humidity > 80, then class = ‘No’

|  |  |  |  |
| --- | --- | --- | --- |
| Temperature | Humidity | Windy | Class |
| 85 | 85 | FALSE | No |
| 80 | 90 | TRUE | No |
| 72 | 95 | FALSE | No |
| 69 | 70 | FALSE | Yes |
| 75 | 70 | TRUE | Yes |



- With outlook = ‘Overcast’ : because overcast infor = 0. This mean, when outlook = overcast, the game of goft will be play

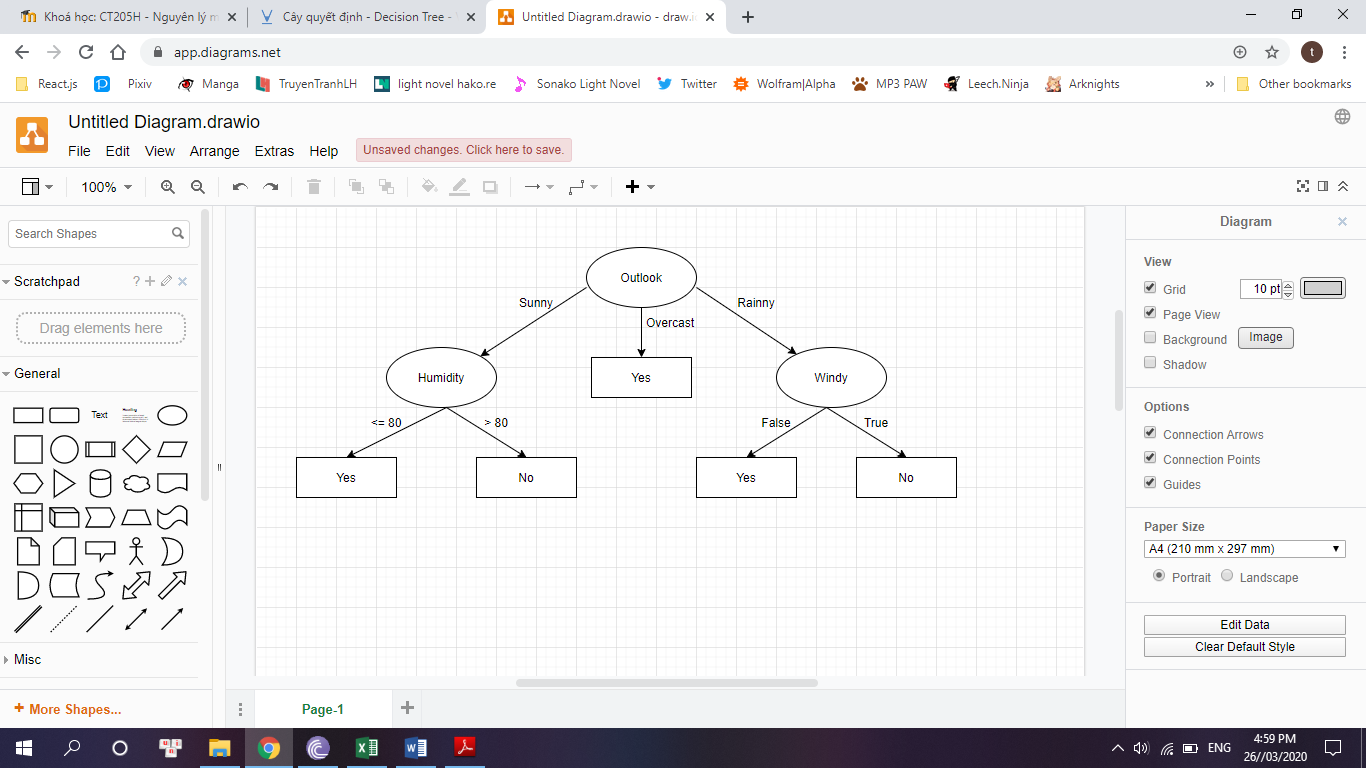


- With outlook = ‘rainny’:

+ if Windy = ‘false’, then class = ‘Yes’

+ If Windy = ‘true’, then class = ‘No’

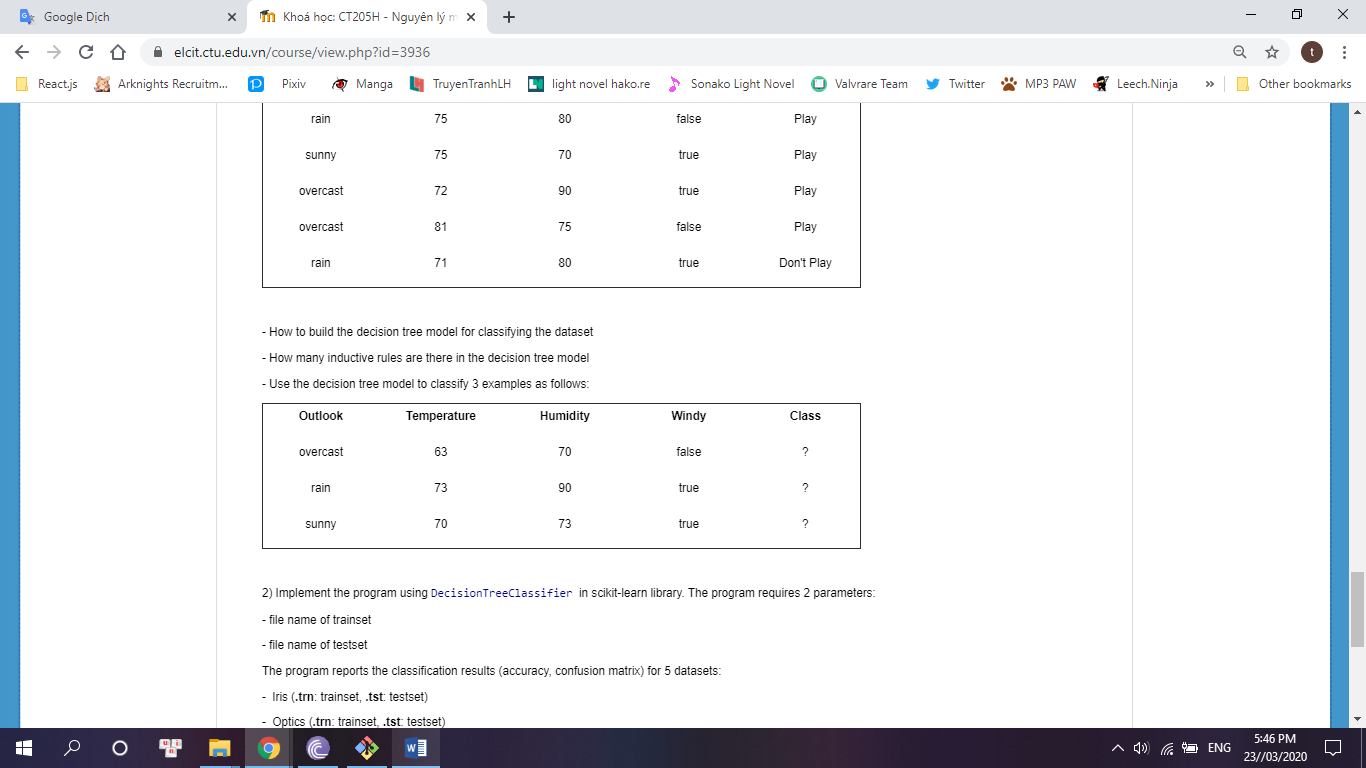
And the completely tree is displayed below



**- How many inductive rules are there in the decision tree model**

1. If outlook = ‘sunny’ and humidity <= 80, then play = yes
2. If outlook = ‘sunny’ and humidity > 80, then play = no
3. If outlook = ‘overcast’, then play = yes
4. If outlook = ‘rainny’ and windy = ‘false’, then play = yes
5. If outlook = ‘rainny’ and windy = ‘true’, then play = no

**- Use the decision tree model to classify 3 examples as follows:**

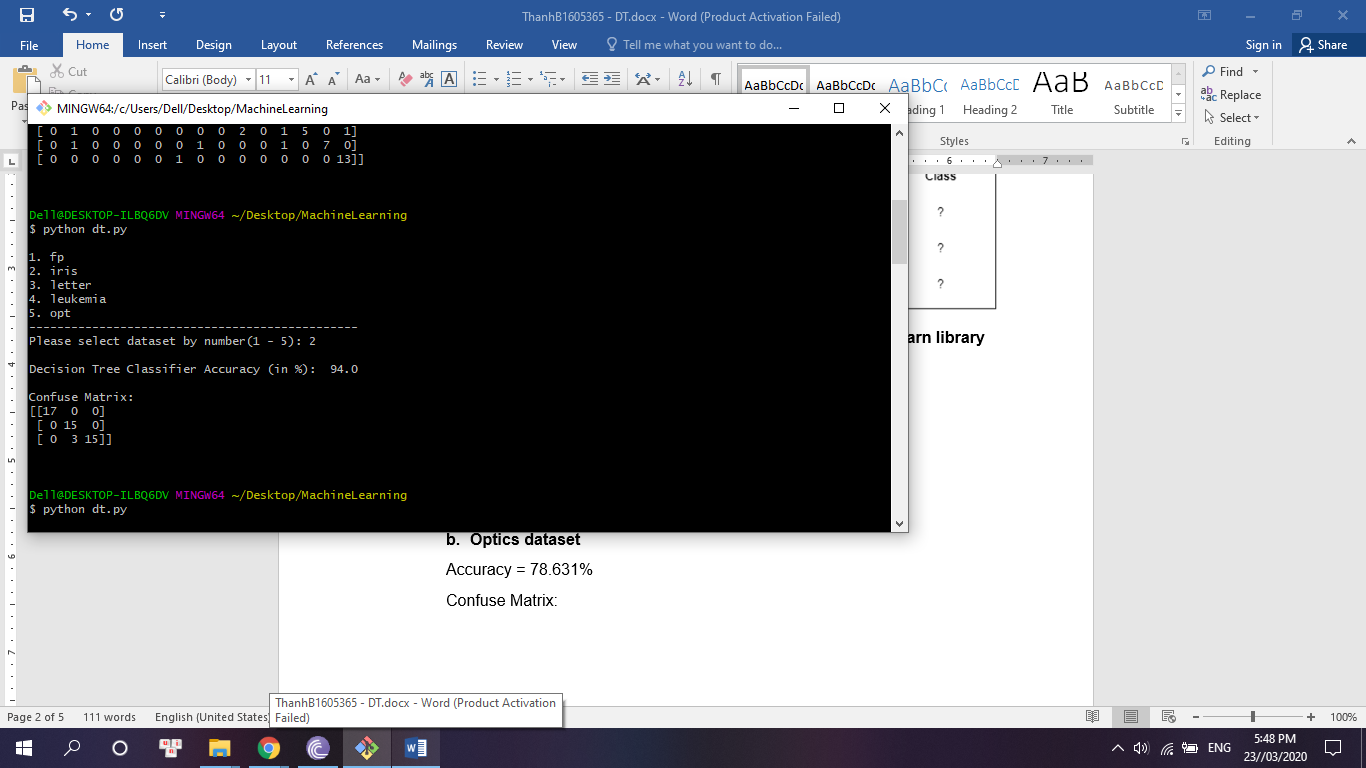


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Outlook | Temperature | Humidity | Windy | Class |
| Overcast | 63 | 70 | FALSE | Yes |
| Rainny | 73 | 90 | TRUE | No |
| Sunny | 70 | 73 | TRUE | Yes |

1. **Implement the program using DecisionTreeClassifier in scikit-learn library**
   1. **Iris dataset**

Accuracy = 94%

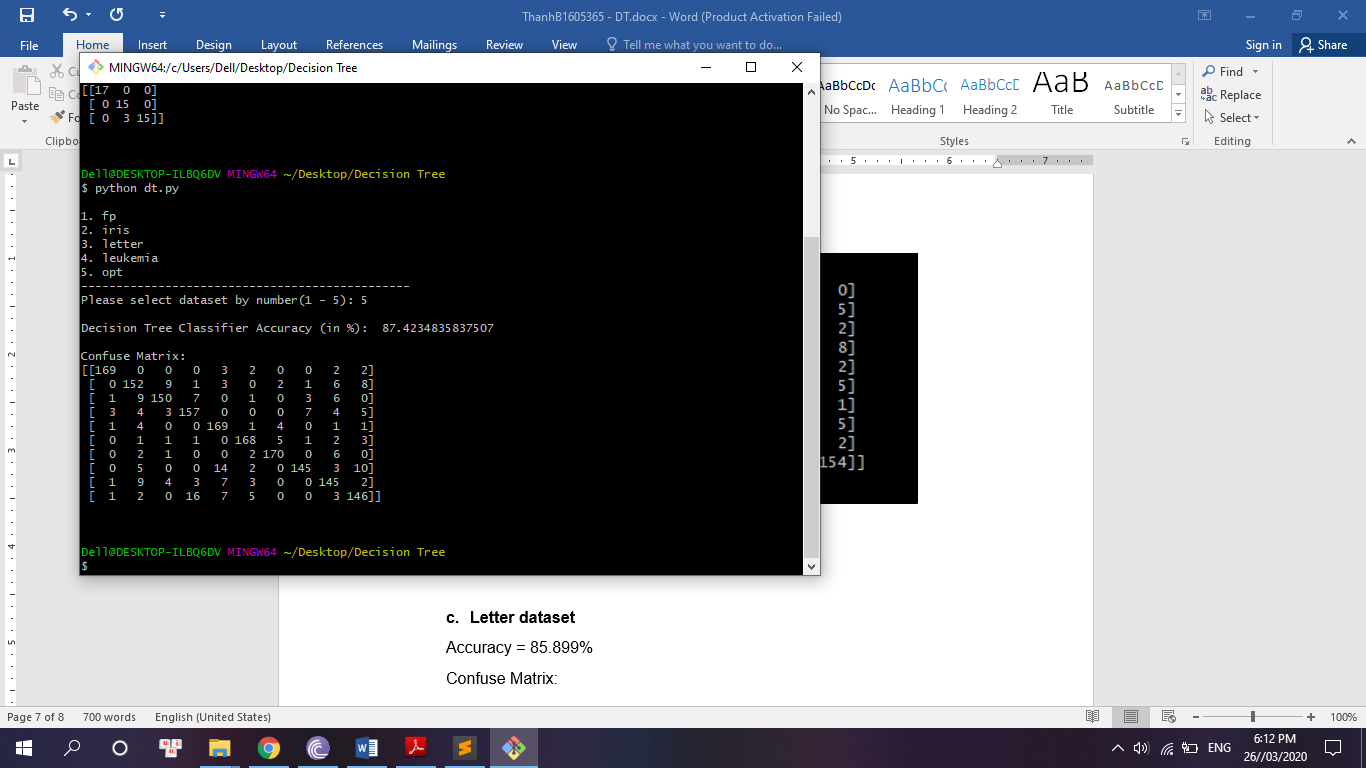
Confuse Matrix:



* 1. **Optics dataset**

Accuracy = 87.423%

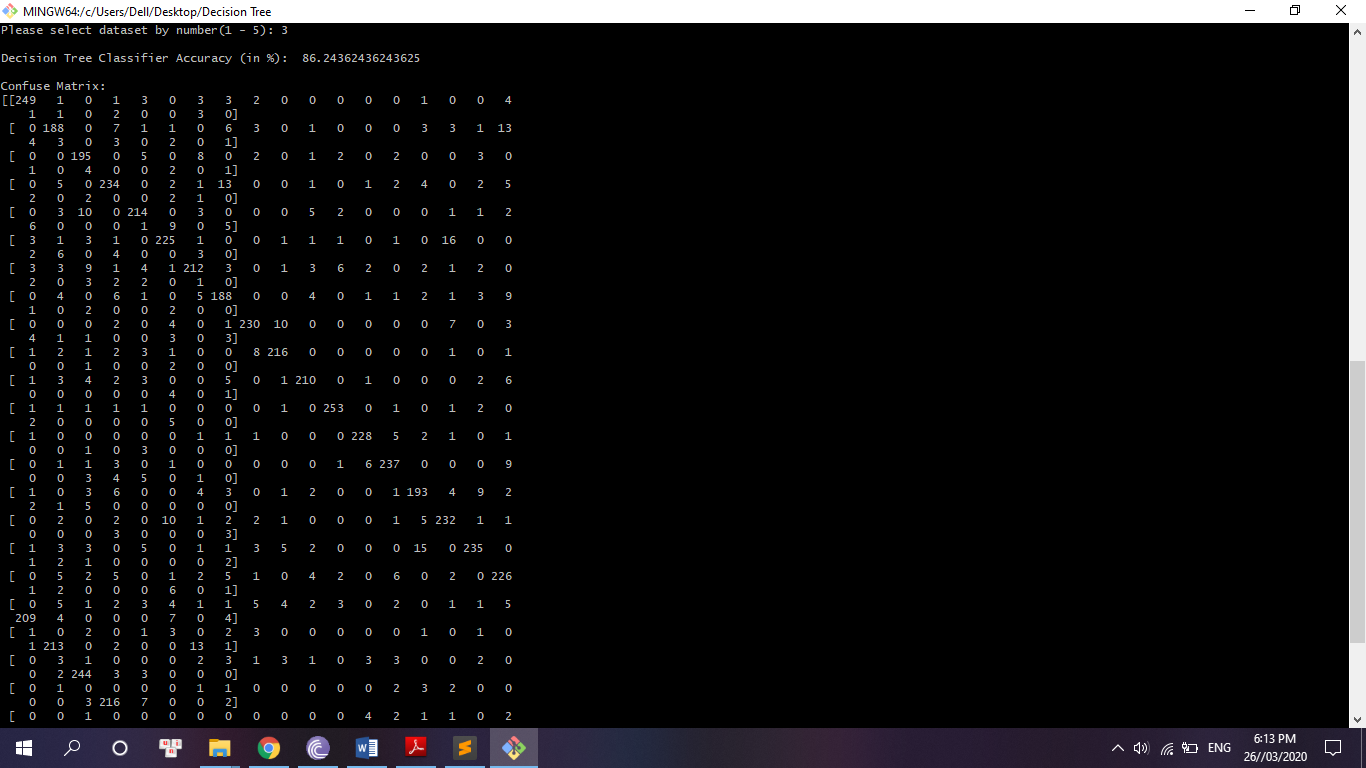
Confuse Matrix:



* 1. **Letter dataset**

Accuracy = 86.243%

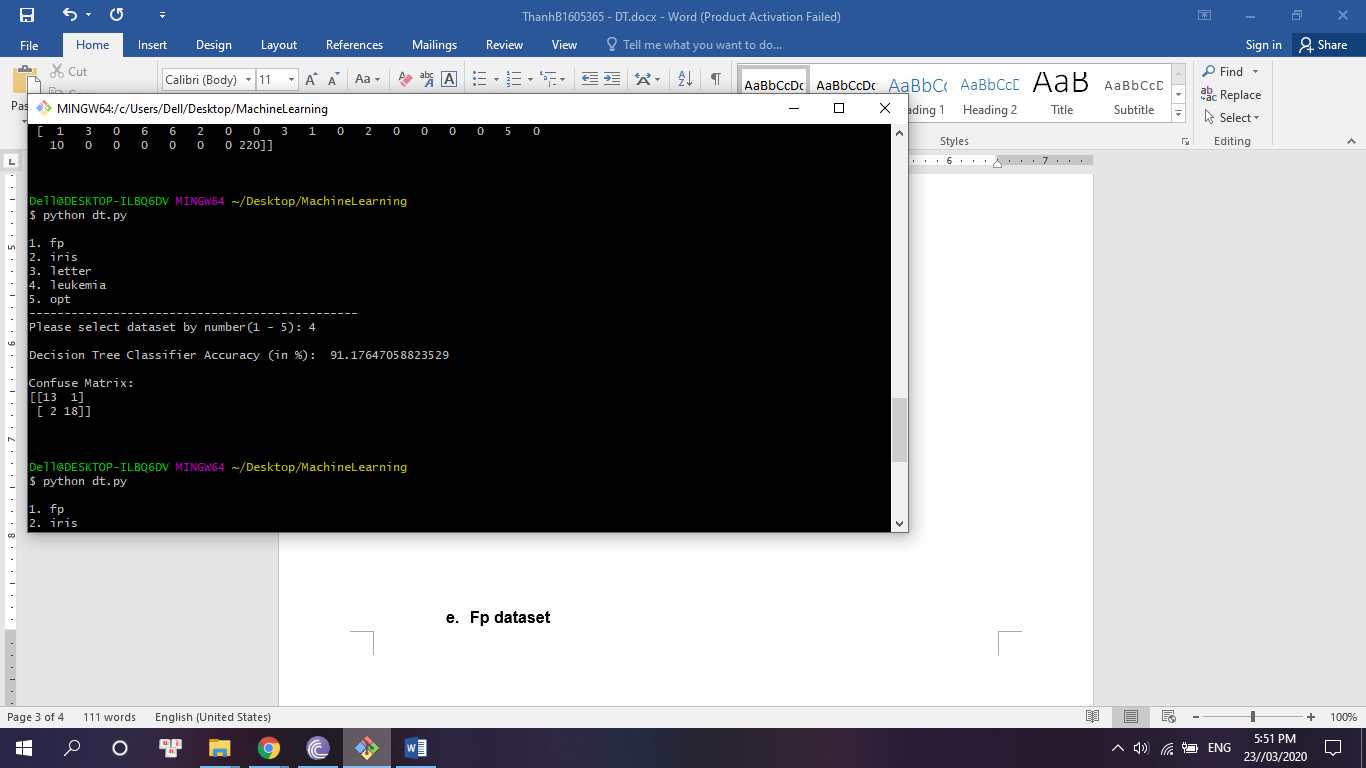
Confuse Matrix:



* 1. **Leukemia dataset**

Accuracy = 91.176

Confuse Matrix:



* 1. **Fp dataset**

Accuracy = 75%

Confuse Matrix:

