HW3 M146 Gawun Kim

①

ⓐ $H(\text{IsGoodRestaurant}) = -\frac{5}{8}\log\frac{5}{8} - \frac{3}{8}\log\frac{3}{8} = \boxed{0.954434}$

Sample #	HasOutdoorSeating	IsGoodRestaurant
1	0	1
2	1	0
3	0	1
4	0	0
5	1	0
6	1	1
7	1	1
8	0	1
9	0	?
10	1	?

ⓑ $H(IsGoodRestaurant|HasOutdoorSeating)$
$= P(HasOutdoorSeating = 0)H(IsGoodRestaurant|HasOutdoorSeating = 0)$
$+ P(HasOutdoorSeating = 1)H(IsGoodRestaurant|HasOutdoorSeating = 1).$

$$H(IGR|HOS) = P(HOS=0)H(IGR|HOS=0) + P(HOS=1)H(IGR|HOS=1)$$

$P(HOS=0) = \frac{4}{8}$

$P(HOS=1) = \frac{4}{8}$

$$H(IGR|HOS=0) = -P(IGR=0|HOS=0)\log P(IGR=0|HOS=0) - P(IGR=1|HOS=0)\log P(IGR=1|HOS=0)$$
$$= \boxed{-\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4}}$$

$$H(IGR|HOS=1) = -P(IGR=0|HOS=1)\log P(IGR=0|HOS=1) - P(IGR=1|HOS=1)\log P(IGR=1|HOS=1)$$
$$= -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = \boxed{1}$$

$$H(IGR|HOS) = P(HOS=0)H(IGR|HOS=0) + P(HOS=1)H(IGR|HOS=1)$$
$$= \frac{4}{8}\left(-\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4}\right) + \frac{4}{8}(1) \approx \boxed{0.9056}$$

ⓒ $H(IGR|X)$ $\quad X \in \{HB, IC, HGA\}$

X_1 HB
X_2 IC
X_3 HGA

Sample #	HasBar (X_1)	IsClean (X_2)	HasGoodAtmosphere (X_3)	IsGoodRestaurant
1	0	1	1	1
2	0	0	0	0
3	1	1	1	1
4	0	0	0	0
5	1	0	0	0
6	0	1	0	1
7	0	0	1	1
8	0	1	1	1
9	1	0	1	?
10	1	1	1	?

$$H(IGR|X_1) = P(X_1=0)H(IGR|X_1=0) + P(X_1=1)H(IGR|X_1=1) \qquad X1 = HB$$
$$= \frac{6}{8}\left(-\frac{4}{6}\log\frac{4}{6} - \frac{2}{6}\log\frac{2}{6}\right) + \frac{2}{8}\left(-\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2}\right) = \boxed{0.9387}$$

Sample #	HasBar (X_1)	IsClean (X_2)	HasGoodAtmosphere (X_3)	IsGoodRestaurant
1	0	1	1	1
2	0	0	0	0
3	1	1	1	1
4	0	0	0	0
5	1	0	0	0
6	0	1	0	1
7	0	0	1	1
8	0	1	1	1
9	1	0	1	?
10	1	1	1	?

$$H(IGR|X_2) = H(IGR|X_2=0)P(X_2=0) + H(IGR|X_2=1)P(X_2=1)$$
$$= \frac{1}{2}\left(-\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4}\right) + \frac{1}{2}\left(-1\cdot\log 1 - 0\cdot\log 0\right)$$
$$= \frac{1}{2}\left(-\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\right) = \boxed{0.405639}$$

$$H(IGR|X_3) = H(IGR|X_3=0)P(X_3=0) + H(IGR|X_3=1)P(X_3=1)$$
$$= \frac{1}{2}\left(-\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4}\right) + \frac{1}{2}\left(-1\cdot\log 1 - 0\cdot\log 0\right)$$
$$= \frac{1}{2}\left(-\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4}\right) = \boxed{0.405639}$$

When X = HasBar $\quad H(IGR|X) = 0.9387$

When X = IsClean $\quad H(IGR|X) = 0.405639$

When X = Has Good Atmosphere $\quad H(IGR|X) = 0.405639$

(d) $I(IGR;X) = H(IGR) - H(IGR|X)$ $\quad X \in \{HOS, HB, IC, HGA\}$

When $X = HOS$ $\quad I(IGR;X) = 0.9544 - 0.9056 = 0.0488$

When $X = HB$ $\quad I(IGR;X) = 0.9544 - 0.9387 = 0.0157$

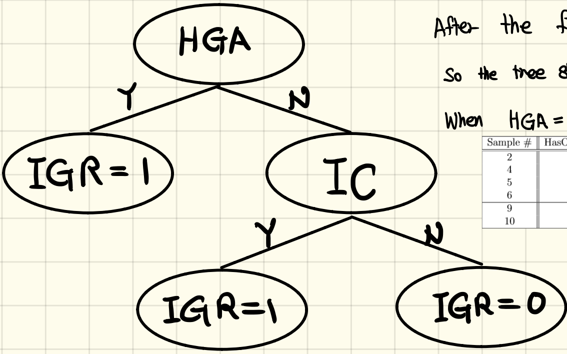
When $X = IC$ $\quad I(IGR;X) = 0.9544 - 0.4056 = 0.5488$

When $X = HGA$ $\quad I(IGR;X) = 0.9544 - 0.4056 = 0.5488$

(e) attribute with High information gain, → when X is IC and HGA

First attribute to split on is IsClean and Has Good Atmosphere.

(f)



After the first, H(Is Good Restaurant | Have Good Atm = 1) = 0.

So the tree stops growing on that branch.

When HGA = 0 (NO),

Sample #	HasOutdoorSeating	HasBar	IsClean	IsGoodRestaurant
2	1	0	0	0
4	0	0	0	0
5	1	1	0	0
6	1	0	1	1
9	0	1	0	?
10	1	1	1	?

In the table

$H(IGR|IC) = 0$ for the reduced set

Therefore splitting using the feature IsClean maximize the information gain

(g) restaurant 9 and 10 are good. since. The first is HasGood Atmosphere and they are all belonged to that. It shows highly information gains and results in truth of them of "Is Good Restaurant"

②

$$J(w_0, w_1) = \sum_{n=1}^{N} \alpha_n (w_0 + w_1 x_{n,1} - y_n)^2$$

$$\left.\begin{aligned} \frac{\partial J}{\partial w_0} &= \sum_{n=1}^{N} 2\alpha_n (w_0 + w_1 x_{n,1} - y_n) \\ \frac{\partial J}{\partial w_1} &= \sum_{n=1}^{N} 2\alpha_n \cdot x_{n,1} (w_0 + w_1 x_{n,1} - y_n) \end{aligned}\right] \text{ from HW2}$$

$$\frac{\partial^2 J}{\partial w_0^2} = \sum_{n=1}^{N} 2\alpha_n \qquad \frac{\partial^2 J}{\partial w_1 \partial w_0} = \sum_{n=1}^{N} 2\alpha_n x_{n,1}$$

$$\frac{\partial^2 J}{\partial w_1^2} = \sum_{n=1}^{N} 2\alpha_n (x_{n,1})^2 \qquad \frac{\partial J}{\partial w_0 \partial w_1} = \sum_{n=1}^{N} 2\alpha_n x_{n,1}$$

$$H = \begin{bmatrix} \frac{\partial^2 J(w_0, w_1)}{\partial w_0^2} & \frac{\partial^2 J(w_0, w_1)}{\partial w_0 \partial w_1} \\ \frac{\partial^2 J(w_0, w_1)}{\partial w_1 \partial w_0} & \frac{\partial^2 J(w_0, w_1)}{\partial w_1^2} \end{bmatrix}$$

$$H = \begin{bmatrix} \sum_{n=1}^{N} 2\alpha_n & \sum_{n=1}^{N} 2\alpha_n x_{n,1} \\ \sum_{n=1}^{N} 2\alpha_n x_{n,1} & \sum_{n=1}^{N} 2\alpha_n (x_{n,1})^2 \end{bmatrix}$$

let $z = \begin{bmatrix} A \\ B \end{bmatrix}$, $z^T H z \geq 0$ $\quad$ * $\alpha_n \geq 0$

$A \neq 0$
$B \neq 0$

$$z^T H z = [A \;\; B] \begin{bmatrix} \sum_{n=1}^{N} 2\alpha_n & \sum_{n=1}^{N} 2\alpha_n x_{n,1} \\ \sum_{n=1}^{N} 2\alpha_n x_{n,1} & \sum_{n=1}^{N} 2\alpha_n (x_{n,1})^2 \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix}$$

$$= A^2\left(\sum_{n=1}^{N} 2\alpha_n\right) + AB\left(\sum_{n=1}^{N} 2\alpha_n x_{n,1}\right) + AB\left(\sum_{n=1}^{N} 2\alpha_n x_{n,1}\right) + B^2\left(\sum_{n=1}^{N} 2\alpha_n (x_{n,1})^2\right)$$

$$= A^2\left(\sum_{n=1}^{N} 2\alpha_n\right) + B^2\left(\sum_{n=1}^{N} 2\alpha_n (x_{n,1})^2\right) + 2AB\left(\sum_{n=1}^{N} 2\alpha_n x_{n,1}\right)$$

$$= \; '' \; + \left[\sum_{n=1}^{N} 2\alpha_n \left(B^2 (x_{n,1})^2\right) + \sum_{n=1}^{N} 2\alpha_n \left(2AB\, x_{n,1}\right)\right]$$

$$= \sum_{n=1}^{N} 2\alpha_n \left(B^2 (x_{n,1})^2 + 2AB (x_{n,1}) + A^2\right)$$

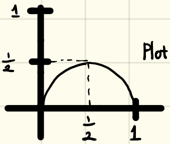
$$= \sum_{n=1}^{N} 2\alpha_n \left(B (x_{n,1}) + A\right)^2 \geq 0$$

H is non-negative. so that positive semi-definite.

③ ⓐ ✓plot Gini index.
✓Binary entropy function. for

$y = \text{gini}(g(v)) = 2[g(v)(1 - g(v))]$

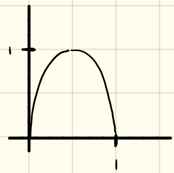
Let $x = g(v)$ $\quad 0 \leq x \leq 1$, then $y = 2x(1-x)$



Plot of the Gini index

Let $x = g(v)$ $\quad 0 \leq x \leq 1$

then $y = -x\log x - (1-x)\log(1-x)$



Plot of the binary entropy function

For the both plots, they start with $y=0$ and end by $y=0$, and they have similar shape of graph. when $x = \frac{1}{2}$, the slope is 0 and concave down.

On the other hand, the only difference is each of values in $x = \frac{1}{2}$

In case of gini, $\text{Gini}(g(v) = \frac{1}{2})$ is $\frac{1}{2}$ and it is a max value. of the graph

In case of binary entropy, $H(g(v) = \frac{1}{2})$ is 1 and it is a max value of the graph.

ⓑ Prove :

If $i(g(V))$ is concave in $g(V)$

$\downarrow$

$I(V_1, V_2, V) \geqq 0,\ V_1 \cup V_2 = V,\ V_1 \cap V_2 = \emptyset$

Let $i(g(V))$ is concave.

$P(V_1, V) = \frac{|V_1|}{|V|} \qquad P(V_2, V) = \frac{|V_2|}{|V|} = 1 - P(V_1, V)$

$I(V_1, V_2, V) = i(g(V)) - (P(V_1,V) \cdot i(g(V_1)) - P(V_2, V)\, i(g(V_2))$

$= i(g(V)) - (P(V_1,V)\, i(g(V_1)) - (1-P(V_1,V))\, i(g(V_2)))$

in this moment $P(V_1, V)$ can be λ

and λ is $\lambda \in [0,1]$

also $g(V_2) = V - g(V_1)$

$= i(g(V)) - \left[\underbrace{\lambda\, i(g(V_1)) + (1-\lambda)\, i(V - g(V_1))}\right]$

$$\boxed{\begin{aligned}&\lambda\, i(g(V_1)) + (1-\lambda)\, i(V - g(V_1)) \\ &\leqq i[\lambda g(V_1) + (1-\lambda)(V - g(V_1))]\end{aligned}}$$

$\geqq i(g(V)) - i[\lambda g(V_1) + (1-\lambda)(V - g(V_1))]$

$\uparrow$

V is meaning larger scale than V_1 and the right one will be at most 1 at least 0

So that $I(V_1, V_2, V) \geqq 0$

ⓒ $H(g(V)) = -g(V)\log_2 g(V) - (1-g(V))\log_2(1-g(V))$

Let $g(V)$ is x, $0 \leq x \leq 1$

$H(x) = -x \log_2 x - (1-x)\log_2(1-x)$

$= -\left[x \frac{1}{\ln 2} \ln x + (1-x)\frac{1}{\ln 2}\ln(1-x)\right]$

$\frac{d}{dx} H(x) = -\left[\frac{1}{\ln 2}\ln x + \cancel{\frac{1}{\ln 2}} - \frac{1}{\ln 2}\ln(1-x) - \cancel{\frac{1}{\ln 2}}\right]$

$\frac{d}{dx}(H(x)) = -\left(\frac{1}{\ln 2}\ln x - \frac{1}{\ln 2}\ln(1-x)\right)$

$\frac{d}{dx}\left(\frac{d}{dx}H(x)\right) = -\left(\frac{1}{\ln 2}\frac{1}{x} + \frac{1}{\ln 2}\frac{1}{1-x}\right)$

$= -\frac{1}{\ln 2}\left(\underbrace{\frac{1}{x} + \frac{1}{1-x}}\right)$

Since $0 \leqq x \leqq 1$, $\frac{d}{dx}\left(\frac{d}{dx}H(x)\right)$ is always non-positive. So that entropy is concave.

ⓓ $gini(g(V)) = 2g(V)(1-g(V))$

let $g(V) = x$.

$gini(x) = 2x(1-x)$

$\frac{d}{dx} gini(x) = 2(1-x) - 2x$

$= 2 - 4x$

$\frac{d}{dx}\left(\frac{d}{dx} gini(x)\right) = -4$

Since double derivative value is negative it is concave.

(4)

(a) In Figure 1,

when $K=1$ error $= \frac{10}{14}$

when $K=3$ error $\frac{6}{14}$

when $K=5$ error $\frac{4}{14}$

when $K=7$ error $\frac{4}{14}$

The value of K is 5 or 7 and error is $\frac{4}{14}$

In Figure 2

when $K=1$ $\frac{2}{14}$

when $K=3$ $\frac{4}{14}$

when $K=5$ $\frac{4}{14}$

when $K=7$ 1

The value of K is 1 and error is $\frac{2}{14}$

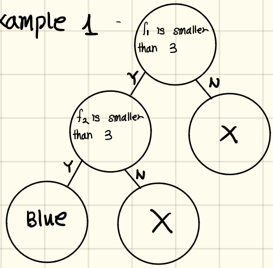
(b) When K value is too big it would be more likely to miss classifying every all data.

When K is too small, the accuracy is decreasing and overfitting.

In figure 1, $K=1 \rightarrow \frac{2}{14}$

$K=13 \rightarrow 1$

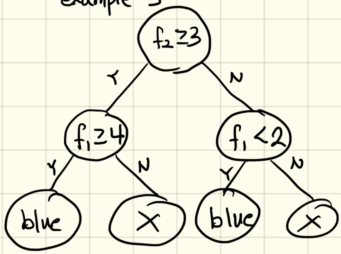
(5) example 1 -



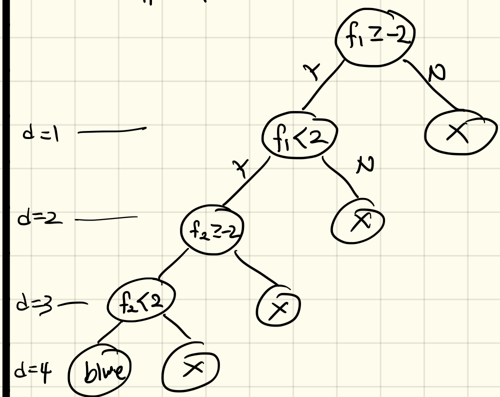
example 2

In each node, a question can be asked with horizontal axis and vertical axis, respectively So that it will be complicated.

example 3



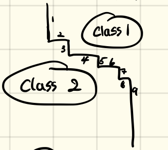
example 4



ⓐ example 1 & example 3
are fully separable using depth 2
decision tree

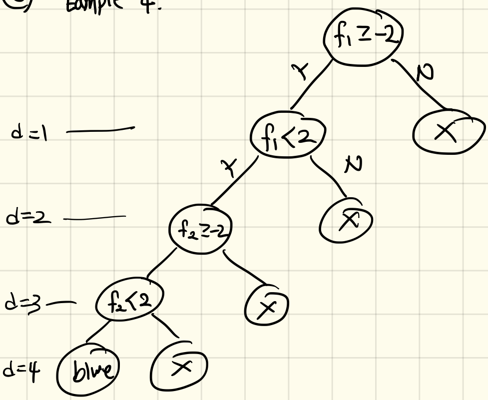
ⓑ Example 2 is the most complicated.
∴ each node can hold only
vertical or horizontal restriction, respectively
So for that it would need a lot of node to
classify two classes.

Such as. like below.



as an example, at least 9 nodes are required

ⓒ Eample 4.



6

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.metrics import accuracy_score
from sklearn import tree

####################################################################
Train_X  = np.genfromtxt("dataTraining_X.csv", delimiter=',', dtype=float)
Train_Y  = np.genfromtxt("dataTraining_Y.csv", delimiter=',', dtype=float)
Test_X   = np.genfromtxt("dataTesting_X.csv", delimiter=',', dtype=float)
Test_Y   = np.genfromtxt("dataTesting_Y.csv", delimiter=',', dtype=float)
Titanic  = np.genfromtxt("titanic.csv", delimiter=',', dtype=float)
####################################################################
```

a

```
############################# Question (A) ######################
print("The accuracy of   TEST   and   TRAIN:")
TWO_DATA = [Test_Y, Train_Y]
for i in TWO_DATA:
    CLASS = []
    COUNT = []
    for data in i:
        index = 0
        while index < len(CLASS):
            if CLASS[index] == data:
                COUNT[index] = COUNT[index] + 1
                break
            else:
                index = index + 1
        if index >= len(CLASS):
            CLASS.append(data)
            COUNT.append(1)
    print("    %f"   %(max(COUNT)/sum(COUNT)))
####################################################################
```

Testing Y : 0.610
Training Y : 0.615

b

```
############################# Question (b) ######################
decision_tree = tree.DecisionTreeClassifier()
decision_tree = decision_tree.fit(Train_X, Train_Y)
res_pred = decision_tree.predict(Test_X)
score = accuracy_score(Test_Y, res_pred)
print("In Question 6-b, the score is %f" %score)
####################################################################
```

0.666

c

```
############################# Question (C) ######################
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()
knn.fit(Train_X, Train_Y)
print("In Question 6-c, the score is %f" %knn.score(Test_X,Test_Y))
####################################################################
```

0.779

d

```
############################# Question (D) ######################
from sklearn.model_selection import train_test_split
from sklearn import metrics

X_train, X_test, y_train, y_test = train_test_split(Train_X, Train_Y, random_state=6)
knn = KNeighborsClassifier(n_neighbors=10)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
print("In Question 6-d, the score is %f" %metrics.accuracy_score(y_test, y_pred))
####################################################################
```

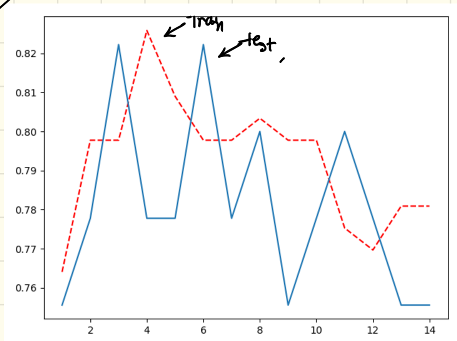
0.797

e

```
############################# Question (E) ######################
PLT_1 = []
PLT_2 = []
for i in range(1, 15, 1):
    X_train, X_test, y_train, y_test = train_test_split(Train_X, Train_Y, random_state=6)
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)
    PLT_1.append(metrics.accuracy_score(y_test, y_pred))

    X_train2, X_test2, y_train2, y_test2 = train_test_split(Test_X, Test_Y, random_state=6)
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train2, y_train2)
    y_pred = knn.predict(X_test2)
    PLT_2.append(metrics.accuracy_score(y_test2, y_pred))

plt.plot(range(1,15,1), PLT_1, 'r--')
plt.plot(range(1,15,1), PLT_2)
plt.show()
####################################################################
```



In the both graphs, depends on the increase of K value, error of each graph is decreasing.

f KNN accuracy was higher than decision tree. So I will use KNN for the Titanic dataset