CAP 5615 Introduction to Neural Networks

2021 Summer

Homework 2 [17 Pts, Due: June 5 2021. Late Penalty: -2/day]: **Solutions**

[If two homework submissions are found to be similar to each other, both submissions will receive 0 grade]

~~[Homework solutions must be submitted through Canvas. No email submission is accepted. If you have multiple files, please include all files as one zip file, and submit zip file online (only zip, pdf, or word files are allowed). You can always update your submissions. I will only grade the latest version.]~~

- Homework solutions must be submitted through Canvas. No email submission is accepted.
- Please try to put all results in one file (e.g., one pdf or word file).
- If you have multiple files, please upload them separately (only **pdf**, **word**, and **html** files are allowed).
- You can always update your submissions. Only the latest version will be graded.]

Question 1 [2 points: 0.25/each]: Please use your own language to briefly explain the following concepts (Must use your own language. No credit if descriptions are copied from external sources):

- Training data vs. test data

In machine learning, training dataset is used to learn/train the models and tune parameters. The test (or testing) dataset is used to validate the performance of the trained models. Test dataset should not be used to train the model but are only used for validation.

- Decision Trees:

A decision tree is a type of decision models which consist of a set of internal nodes/clauses (with testing conditions) and leaf nodes which specify the decisions. A decision tree is an upside-down tree structure. The root node consists of the first feature used to partition the data. One can traverse the tree from the root node down to a path, till reach a leaf node, to make a prediction.

- Overfitting:

In machine learning, overfitting denotes that the underlying learning model overfit to the training data and has low accuracy on the test data. In other words, a learning model may have high accuracies on training data but have a low accuracy on test data. More specifically, given two models h and h1, if the error rate of h on the training data is less than the error rate of h1 on the same training data, but the error rate of h1 on the test data is higher than the error rate of h1 on the same training data, then we conclude that h overfit to the data.

- Prepruning and Postpruning for decision tree learning:

Prepruning and postpruning are two approaches to prune decision trees to avoid overfitting. Preprunning stops growing a branch of the decision tree once certain conditions are met. For example, one can stop split the instance subset once the number of instances inside the subset is less than a certain number (e.g., 3).

Postprunning grows the decision tree in full (i.e., allowing overfitting). After that, it will start to prune some branches and combine instances within the pruned connected-branches as a decision node.

- Entropy:

Entropy measures the expectation of information of a system. The higher the entropy, the more information the system has. If the entropy of the system is zero, there is no information.

Entropy equals to the sum of the multiplication of the probability of each event (or probability of each type of samples) and its log value as follows,

$$\sum_i - p_i \log_2 p_i$$

- Information Gain:

Information Gain measures the difference between the entropy of the system before the Split and the expected entropy of the system after the split.

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

- Information Gain Ratio:

If an attribute has a large number of unique attribute values, it will likely result in large Information Gain. To reduce such bias, information gain ratio is used to divide information gain of the attribute by its split info which measures how broadly and uniformly the attribute splits the data.

GainRatio(S,A) = G(S,A)/SplitInformation(S,A)

Where SplitInformation(S,A) = - Σvi=1 (|Si|/|S|) log2 (|Si|/|S|),

v is the number of values of Attribute A.

- Gini-Index:

Gini-index measure the purity of a set by evaluating the percentage of instances belonging to different classes. Given a set S, its gini-index is calculated using following formula, where where fl is the relative frequency of class l in S.

$$Gini(S) = 1 - \sum_{l=1}^{L} f_l^2$$

With certain splitting criteria T, if we split S into two subsets S1 and S2 with sizes N1 and N2 respectively, the gini index gini(S, T) is defined as

$$Gini_{Split}(S, T) = \frac{N_1}{N} Gini(S_1) + \frac{N_2}{N} Gini(S_2)$$

Question 2 [2 pts]: The following figure shows a toy dataset with two numeric attributes/features (x_1 and x_2) and nine types of instances (color coded using different shapes). The sub-figure on the right panel shows a constructed decision tree from the toy dataset. Please explain

- Roles of interior nodes vs leaf node of the decision trees [0.5 pt].

Each interior node is a selected feature/attribute, and the node is split into two or more branches by using some selected attribute values. The role of each interior node is to partition instances into different subsets for better separation.

Each leaf node is a decision node, determining the label of the instances falling into the current leaf node. To classify an instance, the instance is traversed from the root node downwards toward leaf nodes. The label of the instance is determined by the label of the corresponding label of the leaf node.

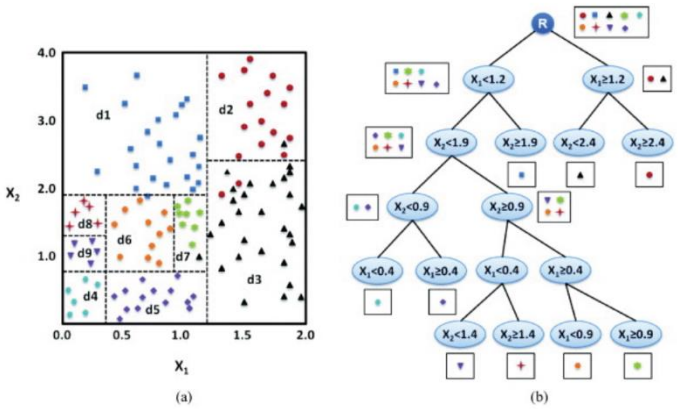
- Explain the process of building a decision tree though recursive partitioning of the feature space [1.0 pt].

When building a decision tree, a selected feature (i.e., an attribute) is used to partition instances into two or more subsets, by using some selected threshold values. Instances corresponding to the selected feature values are partitioned into subsets. This process continues, with one feature being selected in each iteration, and the instances are

partitioned into smaller and smaller subsets, until a termination condition is reached (e.g., all instances in the subset belong to one class).

- What are the meaning of the dashed lines on the left panel, and how does each dashed line correspond to the node on the right panel [0.5 pt].

Each dashed line in the left panel corresponds to an interior node (and the selected threshold value(s)). For each feature, the selected threshold value (e.g., x1<1.2) determines which branches, left or right, the instances belong to. This forms a vertical line perpendicular to the selected feature (i.e., the dashed line).



ID	Days	Outlook	Temperature	Humidity	Wind	Class
1	Mon	Sunny	Hot	High	Weak	No
2	Tue	Sunny	Hot	High	Strong	No
3	Wed	Overcast	Hot	High	Weak	Yes
4	Thu	Rain	Mild	High	Weak	Yes
5	Fri	Rain	Cool	Normal	Weak	Yes
6	Sat	Rain	Cool	Normal	Strong	No
7	Sun	Overcast	Cool	Normal	Strong	Yes
8	Mon	Sunny	Mild	High	Weak	No
9	Tue	Sunny	Cool	Normal	Weak	Yes
10	Wed	Rain	Mild	Normal	Weak	Yes
11	Thu	Sunny	Mild	Normal	Strong	Yes
12	Fri	Overcast	Mild	High	Strong	Yes
13	Sat	Overcast	Mild	Normal	Weak	No
14	Sun	Rain	Hot	High	Strong	Yes
15	Mon	Rain	Mild	High	Strong	No

Table 1

Question 3 [2 pts]: In database showing in Table 1, please calculate the Entropy of the whole dataset (0.5 pt). Use information gain to determine which attribute has the highest Information Gain (1.5 pts) (List major steps)

Entropy of the whole dataset:

In given examples we have 9 positive examples and 6 negative examples:

$$Entropy(S) = -p_1 \log_2 p_1 - p_2 \log_2 p_2$$

p_1=9/15 p_2=6/15

Entropy of the system wil be:

$$Entropy(S) = -\frac{9}{15}\log_2 \frac{9}{15} - \frac{6}{15}\log_2 \frac{6}{15} = 0.4421+0.5287=0.9708$$

Information gain of each individual attribute:

Days

Mon: (1(-), 8(-), 15(-)) Entropy(S_{mon})=0
Tue: (2(-), 9(+)) Entropy(S_{Tue})=1
Wed: (3(+), 10(+)) Entropy(S_{Wed})=0
Thu: (4(+), 11(+)) Entropy(S_{Thu})=0
Fri: (5(+), 12(+)) Entropy(S_{Fri})=0
Sat: (6(-), 13(-)) Entropy(S_{Sat})=0
Sun: (7(+), 14(+)) Entropy(S_{Sun})=0

Conditional Entropy
Entropy(S, Days)=2/15*1=0.1333
Gain(S, Days)=0.9708-0.1333=0.8375

Temperature

---

hot= (1(-), 2(-), 3(+), 14(+))
mild= (4(+), 8(-), 10(+), 11(+), 12(+), 13(-), 15(-))
cool= (5(+), 6(-), 7(+), 9(+))

$$Entropy(S_{hot}) = -0.5\log_2 0.5 - 0.5\log_2 0.5 = 1$$

$$Entropy(S_{mild}) = -\frac{4}{7}\log_2\frac{4}{7} - \frac{3}{7}\log_2\frac{3}{7} = -(0.5714)\log_2 0.5714 - (0.4285)\log_2 0.4285 = 0.9851$$

$$Entropy(S_{cool}) = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} = 0.8112$$

So Final: The conditional Entropy and the information gain of "Temperature" are

$$Entropy(S, Temperature) = \frac{4}{15}Entropy(S_{hot}) + \frac{7}{15}Entropy(S_{mild}) + \frac{4}{15}Entropy(S_{cool}) = 0.9426$$

$$Gain(S, Temperature) = Entropy(S) - Entropy(S, Temperature) = 0.9708 - 0.9426 = 0.028$$

Outlook

---

sunny= (1(-), 2(-), 8(-), 9(+), 11(+))
overcast = (3(+), 7(+), 12(+), 13(-))
rain= (4(+), 5(+), 6(-), 10(+), 14(+), 15(-))

Entropy(S_{sunny})=-0.4log0.4-0.6log0.6 =0.5287-(-0.4421) =0.9708

Entropy($S_{overcast}$)=-0.75log0.75-0.25log0.25 =0.8112

Entropy(S_{rain})=-0.6666log0.6666-0.3333log0.3333 =0.918

So Final: The conditional Entropy and the information gain of "outlook" are

$Entropy(S, outlook)$ =(5/15)0.9708+(4/15)0.8112+(6/15)0.918=0.3236+0.2163+0.3672= 0.9071

$$Gain(S, outlook) = Entropy(S) - Entropy(S, outlook) = 0.9708 - 0.9071 = 0.064$$

Humidity

---

high= (1(-),2(-),3(+),4(+),8(-),12(+),14(+),15(-))
normal= (5(+),6(-),7(+),9(+),10(+),11(+),13(-))

Entropy(S_{high})=-0.5log0.5-0.5log0.5=1

Entropy(S_{normal})=-0.7142log0.7142-0.2857log0.2857=0.3468-(-0.5163)=0.8631

So Final: The conditional Entropy and the information gain of "Humidity" are

$Entropy(S, humidity)$ =(8/15)+(7/15)*0.8631=0.9361
$Gain(S, humidity) = Entropy(S) - Entropy(S, humidity) = 0.9708 - 0.9361 = 0.035$

Wind

---

weak= (1(-),3(+),4(+),5(+),8(-),9(+),10(+),13(-))
strong= (2(-),6(-),7(+),11(+),12(+),14(+),15(-))

Entropy(S_{weak})=-0.625log0.625-0.375log0.375 = 0.4237-(-0.5306) =0.9543

Entropy(S_{strong})=-0.5714log0.5714-0.4285log0.4285 = 0.4613-(-5238) =0.9851

So Final: The conditional Entropy and the information gain of "Wind" are

$Entropy(S, wind)$ =(8/15)*0.9543+(7/15)*0.9851=0.5089+0.4597 = 0.9686
$Gain(S, Wind) = Entropy(S) - Entropy(S, Wind) = 0.9708 - 0.9686 = 0.002$

Therefore, "Days" has the highest information gain.

Question 4 [2.5 pts]: In the dataset showing in Table 1, please calculate Information Gain Ratio of each attributes, and report the **Information Gain Ratio** values of each attribute [2 pts], and determine which attribute should be selected as the root note of the decision tree [0.5 pt].

Splint Information for each attribute:

(Days)=-3/15log(3/15)-6*2/15log(2/15)=0.464+2.325=2.789

(Outlook)=-0.3333log0.3333-0.2666log0.2666-0.4log0.4
=0.5283-(-0.5084)-(-0.5287)=1.5654
(Temperature)=-0.2666log0.2666-0.2666log0.2666-0.4666log0.4666
=0.5084-(-0.5084)-(-0.5131)=1.5299
(Humidity)=-0.4666log0.4666-0.5333log0.5333
=0.5131-(-0.4836)=0.9967
(Wind)=0.9967

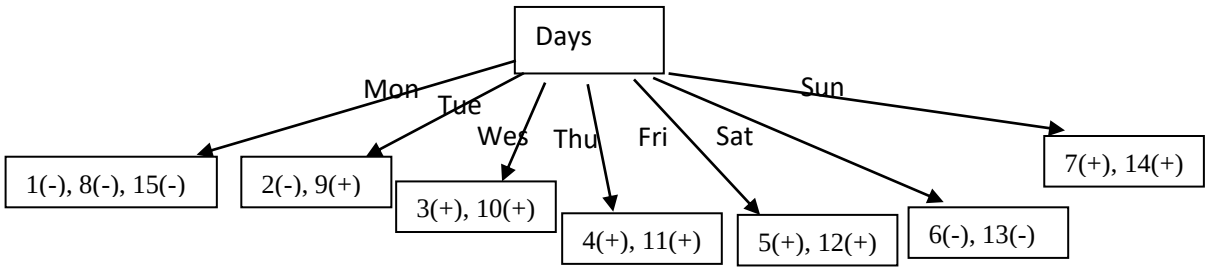
Information Gagin Ration:

Days=(0.8375)/(2.789)=0.3002
OUTLOOK=(0.0637)/(1.5654)=0.0406
TEMP=(0.0280)/(1.5299)=0.0183
HUMIDITY=(0.0347)/(0.9967)=0.0348
WIND=(0.0022)/(0.9967)=0.0022

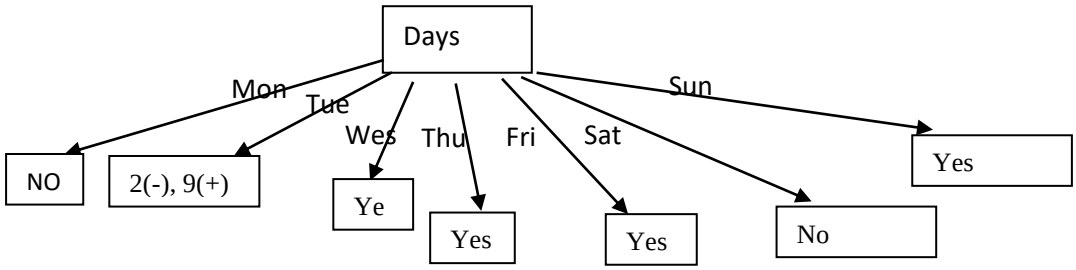
Therefore, "Days" should be selected as the root node.

Question 5 [2 pts]: Using dataset in Table 1 and Information Gain Ratio to create a decision tree with maximum two layers (i.e., the maximum depth of the tree is 2. The root node has 0 depth). List major steps of the tree constructions and report the final decision tree [2 pts]

1. **Determine root node:**
   According to Q4 solutions, "Days" is the attribute with the highest Information Gain Ratio, so we will select "Days" as the root node. Because "Days" has seven attribute values, the root node will have seven branches



2. **Determine second layer label:**
   For each branch, if all instances in the subject belong to one class, we will label the node using corresponding class label, and make the node as leaf node (decision node)



3. **Determine feature used to split second layer:**

There is only one subject at the second layer mixed with positive and negative samples {2(+) and 9(-)}.

Total Entropy on two instances: 1 (because half positive and half negative)

Calculate Information Gain of each remaining attribute on this feature subset

IG(Outlook)=1-1=0

IG (Temperature)=1-0=1

IG(Humidity)=1-0=1

IG(Wind)=1-0=1

+++++

Information Gain Ratio

IGR(Outlook)=0/1.5654=0

IGR(Temperature)=1/1.5299=0.654

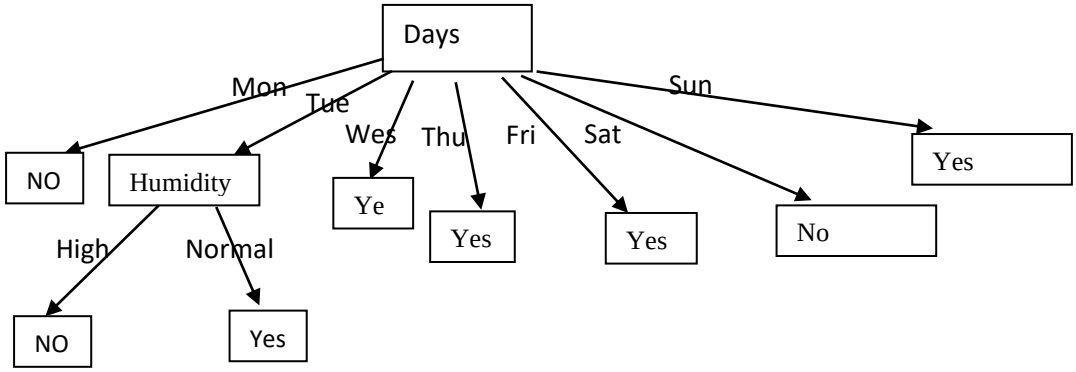
IGR(Humidity)=1/0.9967=1.003

IGR(Wind)=1/0.9967=1.003

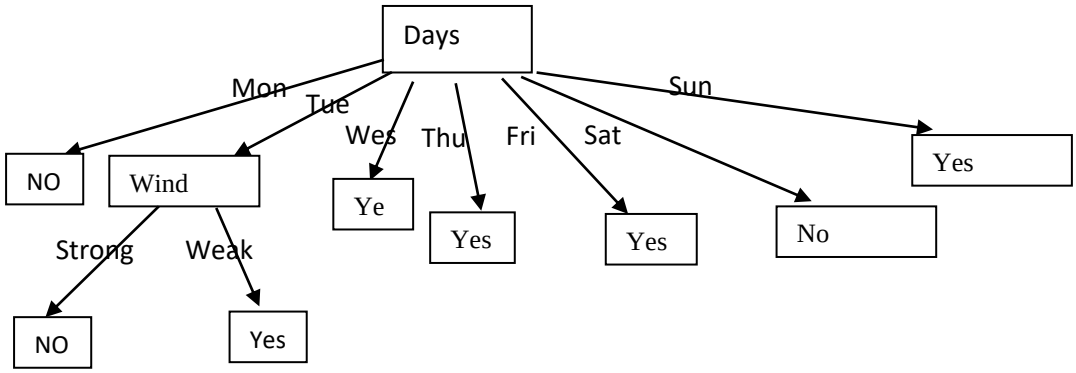
+++++

Humidity and Wind has the highest Information Gain Ratio, so we can random choose one.

Solution 1:



Solution 2:



Question 6 [2.5 pts]: In the dataset showing in Table 1, please use Gini Index to calculate the correlation between each of the five attributes (days, outlook, temperature, humidity, wind) to the Class label, respectively [2 pts]. Please rank and select the most important attribute to build the root node of the decision tree [0.5 pt].

Gini index formula for a set S:

$$Gini(S) = 1 - \sum_{l=1}^{L} f_l^2$$

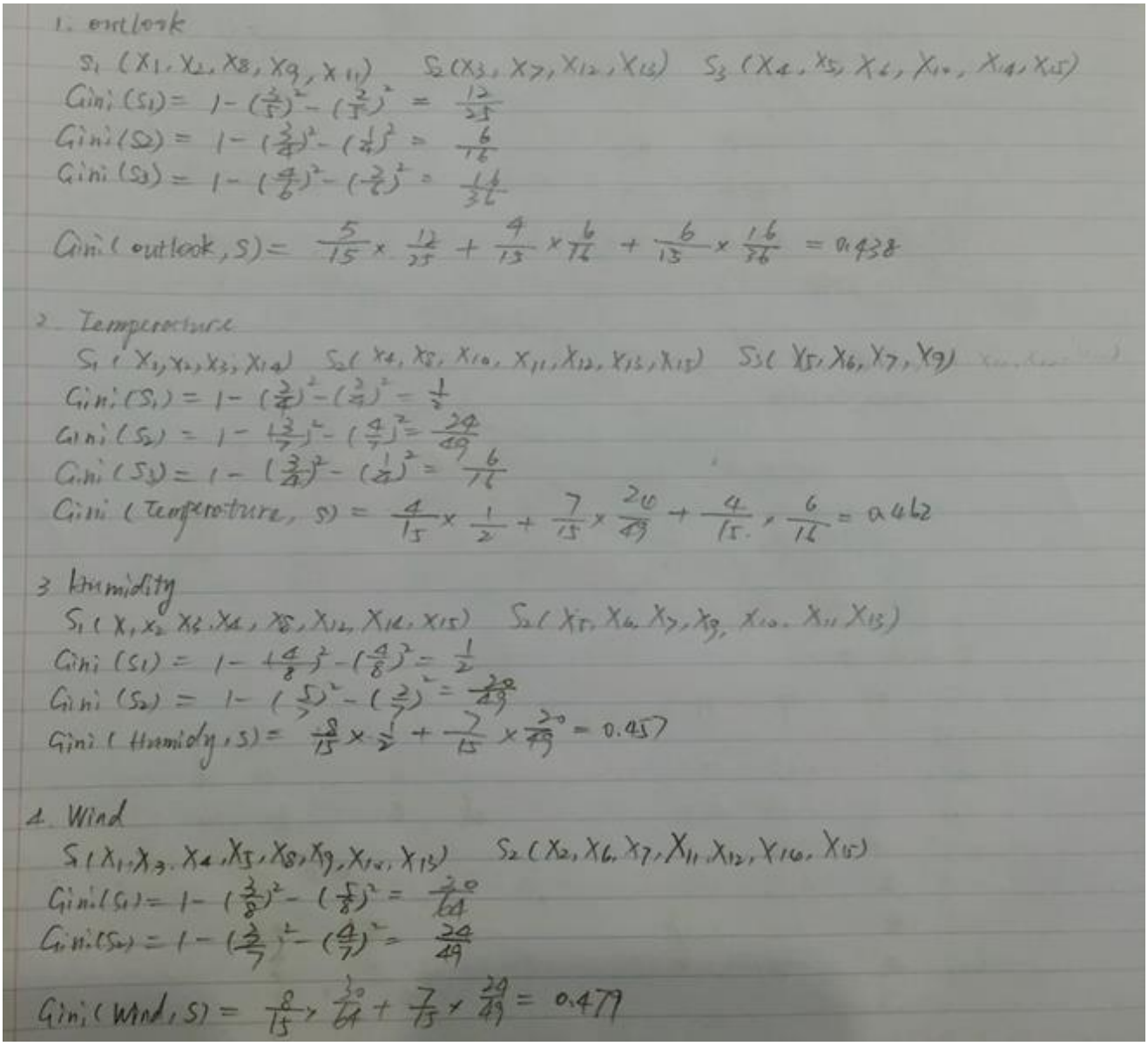
For each feature splitting of set S, using splitting criterion T, the Gini index is calculated as follows:

$$Gini_{Split}(S, T) = \frac{N_1}{N} Gini(S_1) + \frac{N_2}{N} Gini(S_2)$$

For "Days":

S_{Mon}: (1(-), 8(-), 15(-)) Gini(S_{mon})=0
S_{Tue}: (2(-), 9(+)) Gini (S_{Tue})=1/2
S_{Wed}: (3(+), 10(+)) Gini (S_{Wed})=0
S_{Thu}: (4(+), 11(+)) Gini (S_{Thu})=0
S_{Fri}: (5(+), 12(+)) Gini (S_{Fri})=0
S_{Sat}: (6(-), 12(-)) Gini (S_{Sat})=0
S_{Sun}: (7(+), 14(+)) Gini (S_{Sun})=0

Gini(Days, S)=2/15*1/2=1/15=0.067



So we have
Gini(Wind,S)>Gini(Temperature,S)>Gini(Humidity,S)>Gini(Outlook,S)>Gini(Days,S)

Days is selected as the most improtant feature (the root node)

For all programming tasks, please submit the Notebook (or Markdown) as html files for grading (Please do NOT submit .ipynb file)

Question 7 [1 pt]: Please download housing.header.binary.txt dataset from Canvas, and use Python (or R) to implement tasks below (This dataset is the same as housing.header.txt, except that the last feature Medv is discretized with Medv value of the house greater than 230k being 1, or 0 otherwise.)

- Use “Crim” and “Rm” as independent variables to train a decision tree (0.5 pt) to predict whether a house value Medv is 1 or 0 (i.e., whether the house value is greater than 230k or not).
- Visualize the tree (including nodes and lables), and explain the meaning of the values showing in the root node (0.5 pt).

Question 8 [3 pts]: Please download housing.header.binary.txt dataset from Canvas, and use Python (or R) to implement following tasks.

- Please use 80% of instances in the “housing.header.binary.txt” dataset to build a decision tree classifier (using all features) to predict house value Medv [0.5 pt].
- Report the performance of the classifier on the remaining 20% of instances in the “housing.header.binary.txt”
    - Report confusion table, TPR, FPR, and the Accuracy [0.5 pt]
    - Report the ROC curve [0.5 pt]
    - Report the AUC value [0.5 pt]
- Create a new instance with “Crim=0.03, Zn=13, Indus=3.5, Chas=0.3, Nox=0.58, Rm=4.1, Age=68, Dis=4.98, Rad =3, Tax=225, Ptratio=17, B=396, Lstat=7.56”, and predict the Medv value of the instance. Report the classification result [1.0 pt]