CSE 40647/60647: Data Science	**Fall 2019**
Homework 2: Written Assignment	
Handed Out: September 17, 2019	*Due: October 7, 2019 11:55pm*

Save your homework submission as *NETID-hw2-written.pdf*.

1 Naïve Bayes and Deision Tree (40 points)

Consider the following training set:

A	B	C	Y
0	1	1	0
1	1	1	0
0	0	0	0
1	1	0	1
0	1	0	1
1	0	1	1

Here A, B, and C are features; Y is label. Each row is a data object.

(a) Make prediction for $(A = 1, B = 0, C = 0)$ using Naïve Bayes. Learn the Naïve Beyes classifier by estimating all necessary probabilities (including posterori probabilities, prior probabilities, and likelihood probabilities).

(b) Learn a Decision Tree from the above training set using the Information Gain criterion.

Solution:
(a) [20′]
$P(Y = 1) = 0.5$ [2′], $P(Y = 0) = 0.5$ [2′]
$P(A = 1|Y = 1) = 0.67$ [2′]
$P(A = 1|Y = 0) = 0.33$ [2′]
$P(B = 0|Y = 1) = 0.33$ [2′]
$P(B = 0|Y = 0) = 0.33$ [2′]
$P(C = 0|Y = 1) = 0.67$ [2′]
$P(C = 0|Y = 0) = 0.33$ [2′]
$P(X|Y = 1) = 0.67 * 0.33 * 0.67 = 0.15$
$P(X|Y = 0) = 0.33 * 0.33 * 0.33 = 0.04$
$P(Y = 1|X) = 0.5 * 0.15/P(X) = 0.08/P(X)$ [2′]
$P(Y = 0|X) = 0.5 * 0.04/P(X) = 0.02/P(X)$ [2′]
Thus the prediction would be $Y = 1$.

(b) [20′] (-3 for wrong tree, -5 for each wrong probability equation, -2 for wrong calculation)

$Y = \{1*3, 0*3\}$, $H(Y) = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1$
$X_A = \{1*3, 0*3\}$
$H(Y|X_A) = H(Y|A=1) + H(Y|A=0)$
$= \frac{3}{6}(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}) + \frac{3}{6}(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}) = 0.92$
$IG(Y|X_A) = H(Y) - H(Y|X_A) = 1 - 0.92 = 0.08$
$X_B = \{1*4, 0*2\}$
$H(Y|X_B) = H(Y|B=1) + H(Y|B=0) = \frac{4}{6}(-\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}) + \frac{2}{6}(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}) = 1$
$IG(Y|X_B) = H(Y) - H(Y|X_B) = 1 - 1 = 0$
$X_C = \{1*3, 0*3\}$
$H(Y|X_C) = H(Y|C=1) + H(Y|C=0)$
$= \frac{3}{6}(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}) + \frac{3}{6}(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}) = 0.92$
$IG(Y|X_C) = H(Y) - H(Y|X_C) = 1 - 0.92 = 0.08$

Note that $IG(Y|X_A) = IG(Y|X_C)$, so we can pick either A or C as our root node for the decision tree, here we pick C.

When $C = 0$:
$Y = \{1*2, 0*1\}$, $H(Y) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.92$
$X_A = \{1*1, 0*2\}$
$H(Y|X_A) = H(Y|A=1) + H(Y|A=0) = \frac{1}{3}(-\frac{1}{1}\log_2\frac{1}{1}) + \frac{2}{3}(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}) = 0.67$
$IG(Y|X_A) = H(Y) - H(Y|X_A) = 0.92 - 0.67 = 0.25$
$X_B = \{1*2, 0*1\}$
$H(Y|X_B) = H(Y|B=1) + H(Y|B=0) = \frac{2}{3}(-\frac{2}{2}\log_2\frac{2}{2}) + \frac{1}{3}(-\frac{1}{1}\log_2\frac{1}{1}) = 0$
$IG(Y|X_B) = H(Y) - H(Y|X_B) = 0.92 - 0 = 0.92$
$IG(Y|X_B) > IG(Y|X_A)$ so we pick B here.

When $C = 1$:
$Y = \{1*1, 0*2\}$, $H(Y) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.92$
$X_A = \{1*2, 0*1\}$
$H(Y|X_A) = H(Y|A=1) + H(Y|A=0) = \frac{2}{3}(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}) + \frac{1}{3}(-\frac{1}{1}\log_2\frac{1}{1}) = 0.67$
$IG(Y|X_A) = H(Y) - H(Y|X_A) = 0.92 - 0.67 = 0.25$
$X_B = \{1*2, 0*1\}$
$H(Y|X_B) = H(Y|B=1) + H(Y|B=0) = \frac{2}{3}(-\frac{2}{2}\log_2\frac{2}{2}) + \frac{1}{3}(-\frac{1}{1}\log_2\frac{1}{1}) = 0$
$IG(Y|X_B) = H(Y) - H(Y|X_B) = 0.92 - 0 = 0.92$
$IG(Y|X_B) > IG(Y|X_A)$ so we also pick B here.

Thus the tree would be:

If we choose A as the root node, following are two possible result decision trees.

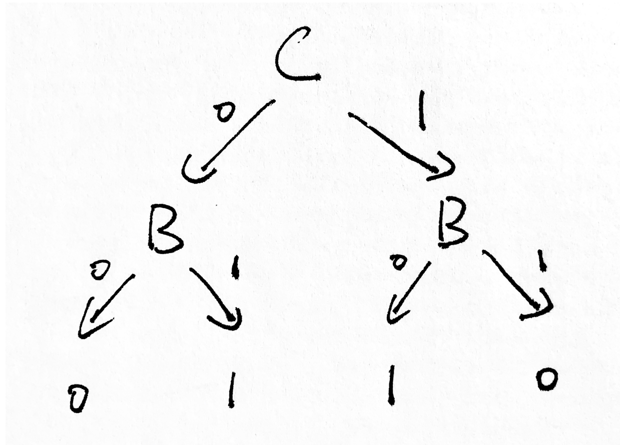

Figure 1: Decision Tree

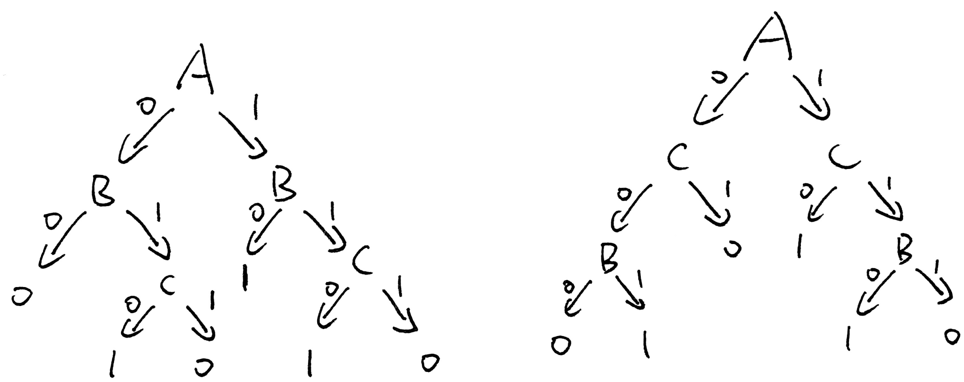

Figure 2: Decision Tree

2 Kernel Function (20 points)

In class, we showed that the quadratic kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^2$ was equivalent to mapping each x into a higher dimensional space where

$$\Phi(x) = ({x^2}_1, {x^2}_2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$$

for the case where $\mathbf{x} = (x_1, x_2)$. Now consider the cubic kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^3$. What is the corresponding Φ function (again, for the special case where $\mathbf{x} = (x_1, x_2)$)?

Solution:
Let $\mathbf{x}_i = (x_{1(i)}, x_{2(i)})$ and $\mathbf{x}_j = (x_{1(j)}, x_{2(j)})$. Then by plugging in $\mathbf{x}_i$ and $\mathbf{x}_j$ into the kernel

function, we can have:

$$
\begin{aligned}
K(\mathbf{x}_i, \mathbf{x}_i) =& (x_{1(i)}x_{1(j)} + x_{2(i)}x_{2(j)} + 1)^3 \\
=& x_{1(i)}^3 x_{1(j)}^3 + 3x_{1(i)}^2 x_{2(i)} x_{1(j)}^2 x_{2(j)} + 3x_{1(i)}^2 x_{1(j)}^2 + 3x_{1(i)} x_{2(i)}^2 x_{1(j)} x_{2(j)}^2 + 6x_{1(i)} x_{2(i)} x_{1(j)} x_{2(j)} \\
& + 3x_{1(i)} x_{1(j)} + x_{2(i)}^3 x_{2(j)}^3 + 3x_{2(i)}^2 x_{2(j)}^2 + 3x_{2(i)} x_{2(j)} + 1
\end{aligned}
$$

This function can be rewrite to the dot product of two vectors in higher dimensional space where

$$\Phi(\mathbf{x}) = (x_1^3, \sqrt{3}x_1^2 x_2, \sqrt{3}x_1^2, \sqrt{3}x_1 x_2^2, \sqrt{6}x_1 x_2, \sqrt{3}x_1, x_2^3, \sqrt{3}x_2^2, \sqrt{3}x_2, 1)$$

3 Classification Evaluation (10 points)

(Choose one answer and prove it) With which of the following conditions, F1 score is equivalent to Accuracy in binary classification?
1. TP = TN; 2. TP = FP; 3. TP = FN; 4. FP = FN
TP: number of true positives; TN: true negatives; FP: false positives; FN: false negatives.

Solution: 1. TP = TN
Proof:
When we have TP = TN:

$$
\begin{aligned}
F1score &= \frac{2}{\frac{1}{precision} + \frac{1}{recall}} \\
&= \frac{2}{\frac{TP+FP}{TP} + \frac{TP+FN}{TP}} \\
&= \frac{2}{\frac{TP+FP+TP+FN}{TP}} \\
&= \frac{2TP}{2TP + FP + FN} \\
&= \frac{TP + TN}{TP + TN + FP + FN} \\
&= Accuracy
\end{aligned}
$$

CSE 40647/60647: Data Science	**Fall 2019**
Homework 2: Programming Assignment	
Handed Out: September 17, 2019	*Due: October 7, 2019 11:55pm*

Save your homework submission as *NETID-hw2-programming.zip*. The zip file has one pdf file *NETID-hw2-programming.pdf* and multiple code files.

Binary Classification with Categorical Features on ND Game Data using no Classification Packages (70 points)

Please use **Python** to solve the problems. You are NOT allowed to directly call any Classification functions (like the decision trees or naive bayes functions in Scikit).

Data files: Dataset-football-train.csv (**training set**), Dataset-football-test.csv (**test set**)

Data introduction: Given Notre Dame's football game data for the last two seasons (2015 and 2016), can we construct three classification models to predict game results on games in 2017? Can we evaluate the model performance? See the table on the last page. Each data object (or called instance) is a game. We have three attributes: (1) "Is Home/Away?", a 2-value attribute ("Home", "Away"), (2) "Is Opponent in AP Top 25 at Preseason?", a 2-value attribute ("In", "Out"), (3) "Media", a 5-value attribute ("1-NBC", "2-ESPN", "3-FOX", "4-ABC", "5-CBS"). The label "Win/Lose" is binary ("Win", "Lose").

- **Training set:** 24 games. Please use game ID 1-24 to construct classification models. (Background color: YELLOW)
- **Test set:** 12 games. Please use your models to predict labels of game ID 25-36 and evaluate the performance of the classification models. (Background color: BLUE)
- **Labels:** Suppose "Win" is the positive label and "Lose" is the negative label. Keep it in mind when you use Precision and Recall to evaluate the models.

For decision tree model construction: We stop splitting instances into child nodes when one of the criteria is satisfied:
(1) All features have been used;
(2) Information Gain or Gain Ratio will be zero with any feature that has not yet been used.

For decision tree model usage:
(1) If the node is not pure, we predict with the majority: For example, if we have 5 positives and 1 negatives, we predict the testing case at this node to be a positive.
(2) If the node has a balance (half/half labels), e.g., 2 positives and 2 negatives, we use the majority of the root node (the entire dataset) for prediction.

Q1: ID3 model using Information Gain (20 points)

Use ID3 to construct a decision tree based on the training set (24 games). Use the tree to predict labels of instances in the testing set (12 games) based on their attributes. Calculate Accuracy, Precision, Recall, and F1 score on the testing result.

Output: Write down (1) your decision tree (either hand-drawn or electronically drawn), (2) predicted labels of the 12 testing games, and (3) evaluation results in the pdf. Save your code as NETID-hw2-1.py.

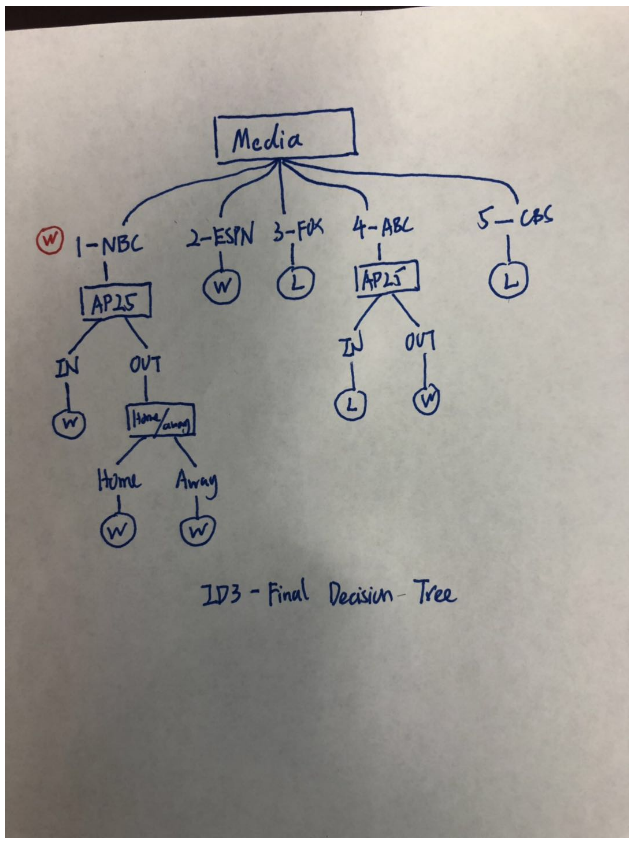

Figure 3: ID3 [3′]

Table 1: ID3 - Prediction of Testing Set [3′]

ID	Is Home or Away	Opponent AP25	Media	Prediction	Actual Label
25	Home	Out	1-NBC	Win	Win
26	Home	In	1-NBC	Win	Lose
27	Away	Out	2-ESPN	Win	Win
28	Away	Out	3-FOX	Lose	Win
29	Home	Out	1-NBC	Win	Win
30	Away	Out	4-ABC	Win	Win
31	Home	In	1-NBC	Win	Win
32	Home	Out	1-NBC	Win	Win
33	Home	Out	1-NBC	Win	Win
34	Away	In	4-ABC	Lose	Lose
35	Home	Out	1-NBC	Win	Win
36	Away	In	4-ABC	Lose	Lose

Accuracy = 0.833. [1′] Precision = 0.889. [1′] Recall = 0.889. [1′] F1 = 0.889. [1′]

[CODE 10′. If your code is not runnable, you will get up to 5‘ depending on your framework design, functions design and the clarity of the pseudo code or comments.]

Q2: C4.5 model using Gain Ratio (20 points)

Use C4.5 to construct a decision tree based on the training set (24 games). Use the tree to predict labels of instances in the testing set (12 games) based on their attributes. Calculate Accuracy, Precision, Recall, and F1 score on the testing result.

Output: Write down (1) your decision tree (either hand-drawn or electronically drawn), (2) predicted labels of the 12 testing games, and (3) evaluation results in the pdf. Save your code as NETID-hw2-2.py.

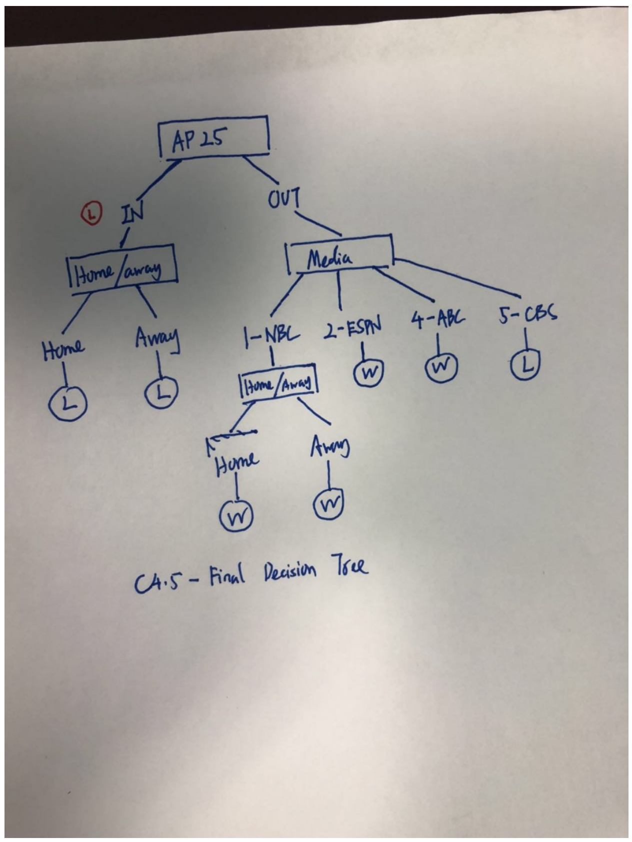

Figure 4: C4.5 [3′]

Table 2: C4.5 - Prediction of Testing Set [3′]

ID	Is Home or Away	Opponent AP25	Media	Prediction	Actual Label
25	Home	Out	1-NBC	Win	Win
26	Home	In	1-NBC	Lose	Lose
27	Away	Out	2-ESPN	Win	Win
28	Away	Out	3-FOX	Win	Win
29	Home	Out	1-NBC	Win	Win
30	Away	Out	4-ABC	Win	Win
31	Home	In	1-NBC	Lose	Win
32	Home	Out	1-NBC	Win	Win
33	Home	Out	1-NBC	Win	Win
34	Away	In	4-ABC	Lose	Lose
35	Home	Out	1-NBC	Win	Win
36	Away	In	4-ABC	Lose	Lose

Accuracy = 0.917. [1′]Precision = 1. [1′] Recall = 0.889. [1′] F1 = 0.941. [1′]

[CODE 10′. If your code is not runnable, you will get up to 5′ depending on your framework design, functions design and the clarity of the pseudo code or comments.]

Q3: Naive Bayes model without Zero Correction (30 points)

Use Naive Bayes to predict labels of instances in the test set (12 games) based on the training set (24 games). Calculate Accuracy, Precision, Recall, and F1 score on the testing result.

Output: Write down (1) predicted labels of the 12 testing games and (2) evaluation results in the pdf. Save your code as NETID-hw2-3.py.

Table 3: Naive Bayes - Prediction of Testing Set [6′]

ID	Is Home or Away	Opponent AP25	Media	Prediction	Actual Label
25	Home	Out	1-NBC	Win	Win
26	Home	In	1-NBC	Win	Lose
27	Away	Out	2-ESPN	Win	Win
28	Away	Out	3-FOX	Lose	Win
29	Home	Out	1-NBC	Win	Win
30	Away	Out	4-ABC	Lose	Win
31	Home	In	1-NBC	Win	Win
32	Home	Out	1-NBC	Win	Win
33	Home	Out	1-NBC	Win	Win
34	Away	In	4-ABC	Lose	Lose
35	Home	Out	1-NBC	Win	Win
36	Away	In	4-ABC	Lose	Lose

Accuracy = 0.75. [2′] Precision = 0.875. [2′] Recall = 0.778. [2′] F1 = 0.8235. [2′]

[CODE 16′. If your code is not runnable, you will get up to 8′ depending on your framework design, function designs and the clarity of the pseudo code or comments.]

Multi-Class Classification with Numerical Features on Film Data using CART (30 points)

You are allowed to use *any* programming language (Python recommended; R, C++, Java, etc.), however, the solutions will be in **Python**. You are allowed to use *any* public package (including Numpy and Scikit-learn) and any other kind of tools (Excel).

Please self learn the Section 1.10.1 on Web Page:
`http://scikit-learn.org/stable/modules/tree.html`.

Construct the CART model following the instruction and apply on the film data we used in Homework 1. All the 150 films will be used as training data points. You are not asked to do model usage or model evaluation.

[CODE 20'. If your code is not runnable, you will get up to 10' depending on your framework design, functions design and the clarity of the pseudo code or comments.]

Output: Write down the CART model (the decision tree) that was learned from the film data and in the pdf. We strongly recommend you export your tree by in Graphviz format. For more detail of this package, here is the link: http://www.graphviz.org. Save your code as NETID-hw2-4.py.

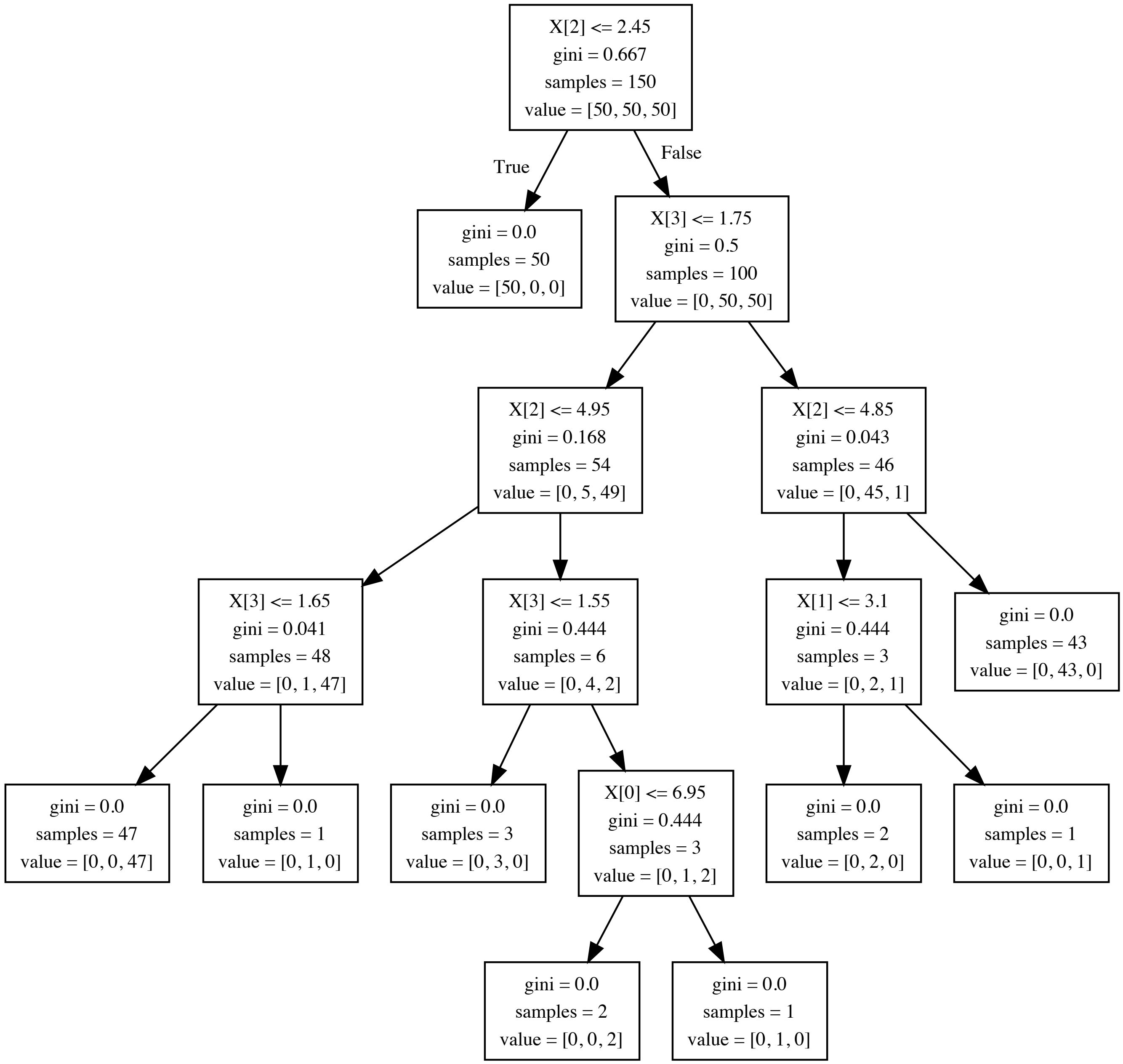

Figure 5: Tree structure [10′]