

Introduction to Machine Learning

Program Assignment #1

Problem: (70%)

For this assignment, you need to implement **ID3 algorithm** to construct a decision tree with C, C++, Java or python2/3, and use K-fold cross validation (K=5) to validate classification performance by outputting precision and recall for each class and total accuracy. **You CANNOT use packages to do the jobs for you.**

ID3 algorithm

Algorithm 4.1 Pseudocode description of the ID3 algorithm.

Require: set of descriptive features \mathbf{d}

Require: set of training instances \mathcal{D}

- 1: **if** all the instances in \mathcal{D} have the same target level C **then**
- 2: **return** a decision tree consisting of a leaf node with label C

3: **else if** d is empty **then**

4: **return** a decision tree consisting of a leaf node
with the label of the majority target level in \mathcal{D}

5: **else if** \mathcal{D} is empty **then**

6: **return** a decision tree consisting
with the label of the majority
target level of the dataset of the
immediate parent node

7: **else**

8: $\mathbf{d} [best] \leftarrow \arg \max IG (d, \mathcal{D}) \quad d \in \mathbf{d}$

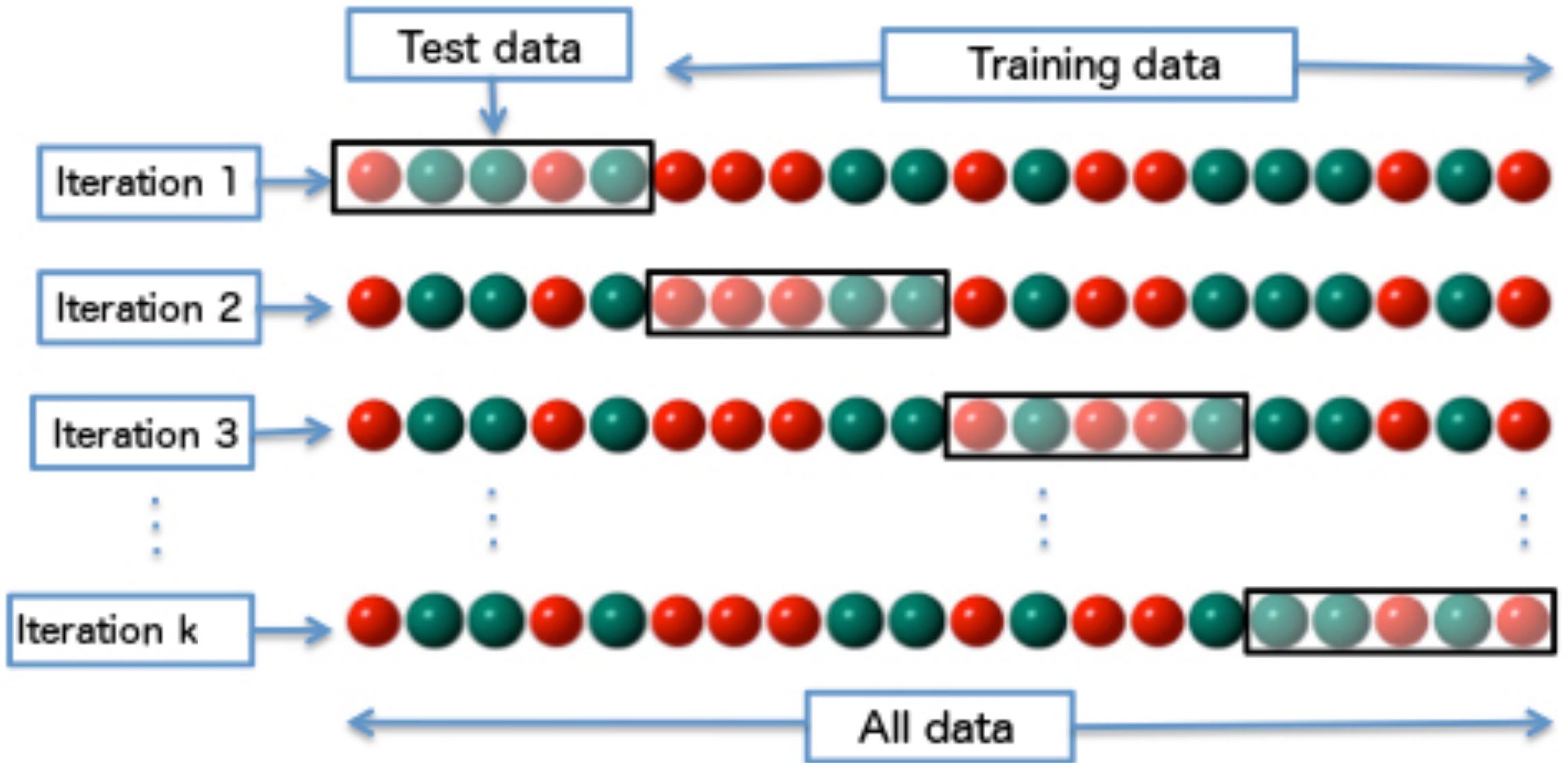
10: partition \mathcal{D} using $\mathbf{d} [best]$

11: remove $\mathbf{d} [best]$ from \mathbf{d}

12: **for** each partition \mathcal{D}_i of \mathcal{D} **do**

13: grow a branch from $Node_{\mathbf{d}[best]}$
by rerunning *ID3* with $\mathcal{D} = \mathcal{D}_i$

K – Fold Cross Validation



After iteration i ,

$ACCURACY_i$, $PRECISION_i$, $RECALL_i$ will be obtained.

In this assignment, output the average accuracy, precision and recall:

$$ACCURACY_{avg} = \frac{\sum_{i=0}^{K-1} ACCURACY_i}{K}$$

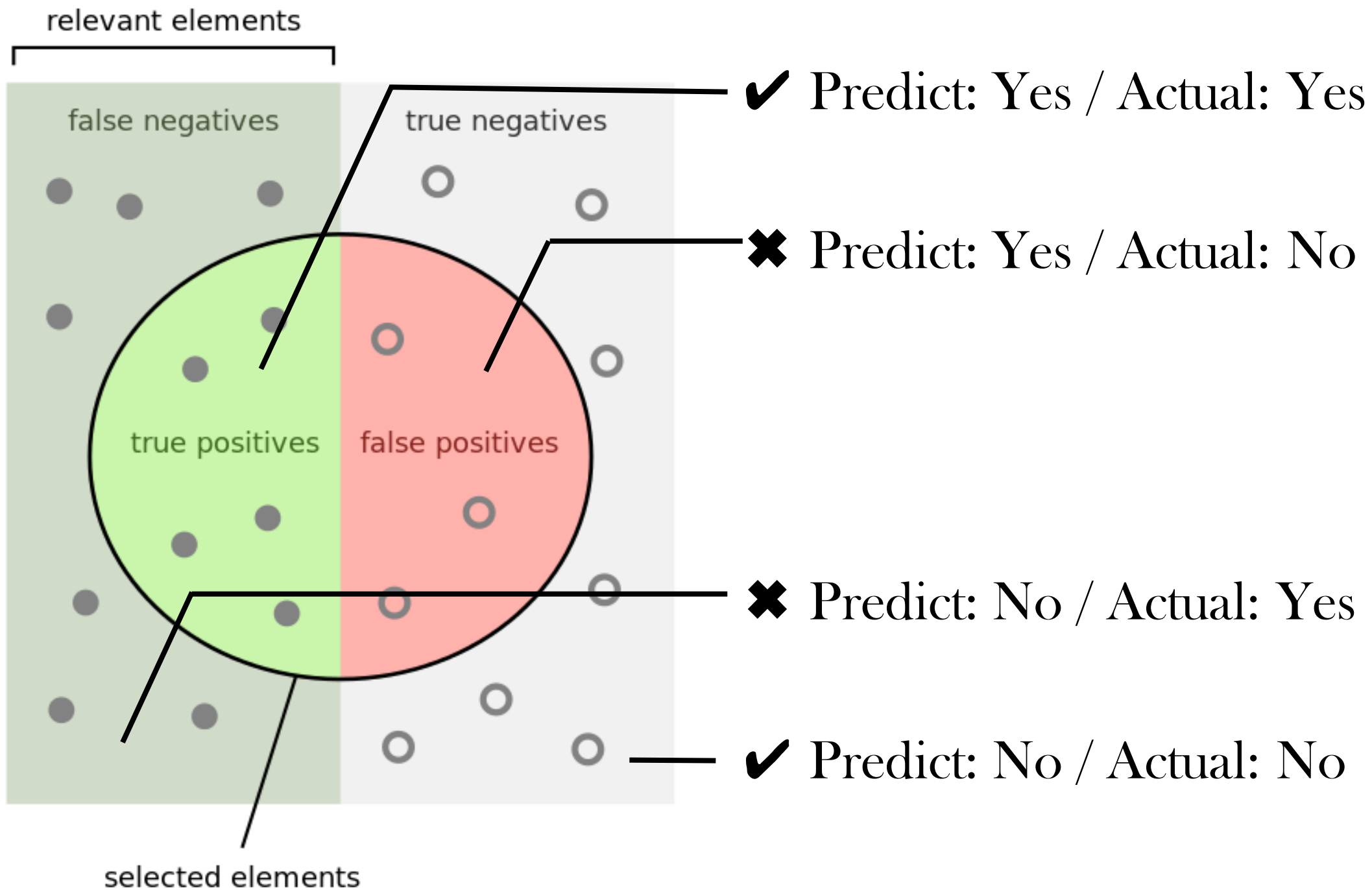
$$PRECISION_{avg} = \frac{\sum_{i=0}^{K-1} PRECISION_i}{K}$$

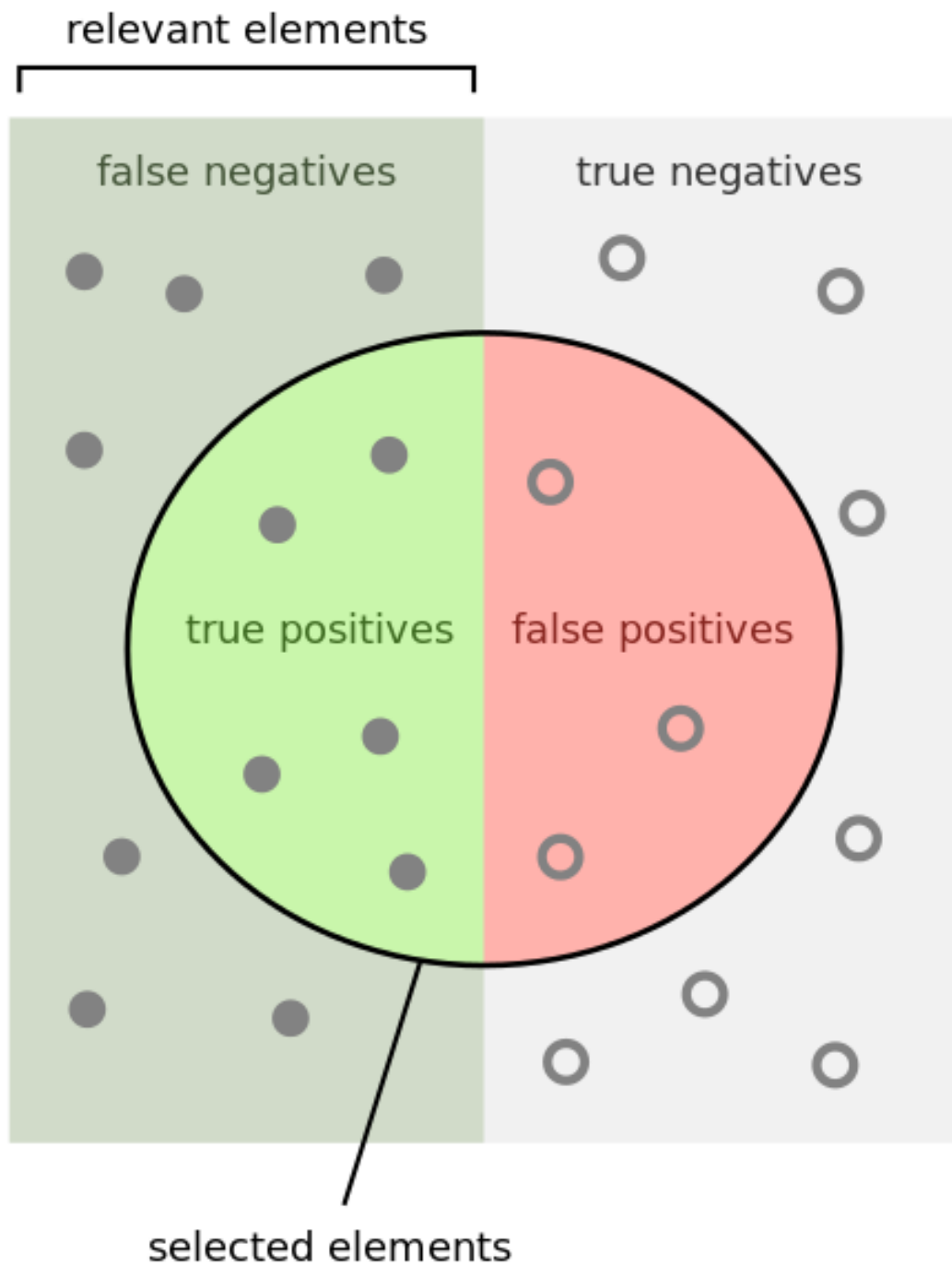
$$RECALL_{avg} = \frac{\sum_{i=0}^{K-1} RECALL_i}{K}$$

ACCURACY_i

PRECISION_i

RECALL_i





How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

$$\text{Accuracy} = \frac{\text{true positives} + \text{true negatives}}{\text{true positives} + \text{false positives} + \text{false negatives} + \text{true negatives}}$$

You should upload a single **[student-id].ZIP** file which contains a 'run.sh' shell script, source files, data and a report. The 'run.sh' should compile the source code (for C/C++ and java) and execute the program which output the results by a single **'./run.sh'** command.

Note that, the instances of data must be randomly shuffled before constructing decision trees. The accuracy, precision and recall must be floating numbers within 0 and 1 and be arranged with the following format.

[Total accuracy]

[Precision of class 0] [Recall of class 0]

[Precision of class 1] [Recall of class 1]

[Precision of class 2] [Recall of class 2]

03:25:02

...Projects/tmp/test

\$ l

total 8

drwxr-xr-x	3	toosyou	staff	102B	10	17	03:25	.
drwxr-xr-x	4	toosyou	staff	136B	10	16	01:54	..
-rw-r--r--	1	toosyou	staff	711B	10	17	03:24	0316313.zip

03:25:03

...Projects/tmp/test

\$ unzip 0316313.zip && chmod +x 0316313/run.sh && ./0316313/run.sh

Archive: 0316313.zip

creating: 0316313/

extracting: 0316313/data.csv

inflating: 0316313/run.sh

extracting: 0316313/sourcefiles.py

0.998

[Total accuracy]

0.987 0.840 [Precision of class 0] [Recall of class 0]

0.809 0.892 [Precision of class 1] [Recall of class 1]

0.960 0.694 [Precision of class 2] [Recall of class 2]

第十二章、學習 Shell Scripts

Data:

<https://archive.ics.uci.edu/ml/datasets/Iris>

including 150 number of instances with 4 attributes.

Attribute Information:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class:
 - Iris Setosa
 - Iris Versicolour
 - Iris Virginica

Handling Continuous Descriptive Features

ID	STREAM	SLOPE	ELEVATION	VEGETATION
1	false	steep	3,900	chapparal
2	true	moderate	300	riparian
3	true	steep	1,500	riparian
4	false	steep	1,200	chapparal
5	false	flat	4,450	conifer
6	true	steep	5,000	conifer
7	true	steep	3,000	chapparal

ID	STREAM	SLOPE	ELEVATION	VEGETATION
2	true	moderate	300	riparian
4	false	steep	1,200	chapparal
3	true	steep	1,500	riparian
7	true	steep	3,000	chapparal
1	false	steep	3,900	chapparal
5	false	flat	4,450	conifer
6	true	steep	5,000	conifer

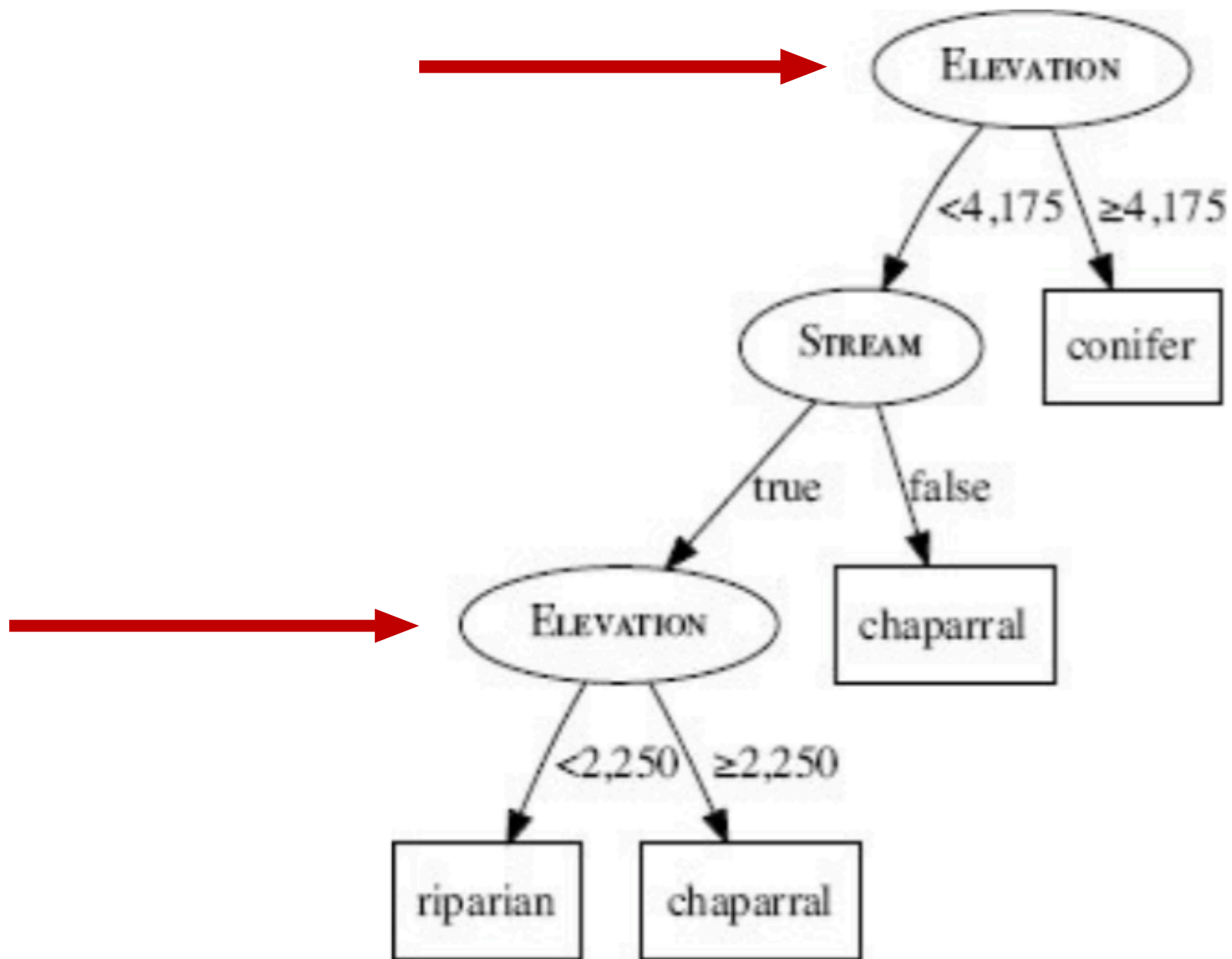
- the boundary between \mathbf{d}_2 and \mathbf{d}_4 is $\frac{300 + 1,200}{2} = 750$
- the boundary between \mathbf{d}_4 and \mathbf{d}_3 is $\frac{1,200 + 1,500}{2} = 1,350$
- the boundary between \mathbf{d}_3 and \mathbf{d}_7 is $\frac{1,500 + 3,000}{2} = 2,250$
- the boundary between \mathbf{d}_1 and \mathbf{d}_5 is $\frac{3,900 + 4,450}{2} = 4,175$

Split by Threshold	Part.	Instances	Partition Entropy	Rem.	Info. Gain
≥ 750	\mathcal{D}_1	\mathbf{d}_2	0.0	1.2507	0.3060
	\mathcal{D}_2	$\mathbf{d}_4, \mathbf{d}_3, \mathbf{d}_7, \mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_6$	1.4591		
$\geq 1,350$	\mathcal{D}_3	$\mathbf{d}_2, \mathbf{d}_4$	1.0	1.3728	0.1839
	\mathcal{D}_4	$\mathbf{d}_3, \mathbf{d}_7, \mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_6$	1.5219		
$\geq 2,250$	\mathcal{D}_5	$\mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_3$	0.9183	0.9650	0.5917
	\mathcal{D}_6	$\mathbf{d}_7, \mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_6$	1.0		
$\geq 4,175$	\mathcal{D}_7	$\mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_3, \mathbf{d}_7, \mathbf{d}_1$	0.9710	0.6935	0.8631
	\mathcal{D}_8	$\mathbf{d}_5, \mathbf{d}_6$	0.0		

Unlike categorical features,

continuous features can be used at multiple points

, although the threshold will be different.



Report: (10%)

The report should include the results, environment, using library and language, explain of your code and how to use it.

Accuracy, Precision and Recall: (20%)

We will test your source code and score base on your rank of following metrics:

$$1.5 \times Accuracy + \sum_{i=0}^2 (Precision_i + Recall_i)$$

Environment:

Your program will be executed on the following environment:

- Ubuntu 16.04.3 LTS
- gcc 5.4.0
- openjdk 1.8.0_131
- python 2.7.12
- python 3.5.2

Bonus: (20%)

Implement Random Forest algorithm and make a 'RF.sh' shell script to output the result with the same format.