Machine Learning Exercise Sheet 2

k-Nearest Neighbors and Decision Trees

1 kNN Classification

Problem 1: You are given the following dataset, with points of two different classes:

Name	x_1	x_2	class
A	1.0	1.0	1
B	2.0	0.5	1
C	1.0	2.5	1
D	3.0	3.5	2
E	5.5	3.5	2
F	5.5	2.5	2

We perform 1-NN classification with leave-one-out cross validation on the data in the plot.

a) Compute the distance between each point and its nearest neighbor using L_1-norm as distance measure.

b) Compute the distance between each point and its nearest neighbor using L_2-norm as distance measure.

c) What can you say about classification if you compare the two distance measures?

	a)	b)
A -> B	L1 dist = abs(1-2) + abs(1-0.5) = 1.5	L2 dist = 1.12
A -> C	L1 dist = abs(1-1) + abs(1-2.5) = 1.5	L2 dist = 1.50
A -> D	L1 dist = abs(1-3) + abs(1-3.5) = 4.5	L2 dist = 3.20
A -> E	L1 dist = abs(1-5.5) + abs(1-3.5) = 7	L2 dist = 5.15
A -> F	L1 dist = abs(1-5.5) + abs(1-2.5) = 6	L2 dist = 4.74
B -> C	L1 dist = abs(2-1) + abs(0.5-2.5) = 3	L2 dist = 2.24
B -> D	L1 dist = abs(2-3) + abs(0.5-3.5) = 4	L2 dist = 3.16
B -> E	L1 dist = abs(2-5.5) + abs(0.5-3.5) = 6.5	L2 dist = 4.61
B -> F	L1 dist = abs(2-5.5) + abs(0.5-2.5) = 5.5	L2 dist = 4.03

C -> D	L1 dist = abs(1-3) + abs(2.5-3.5) = 3	L2 dist = 2.24
C -> E	L1 dist = abs(1-5.5) + abs(2.5-3.5) = 5.5	L2 dist = 4.61
C -> F	L1 dist = abs(1-5.5) + abs(2.5-2.5) = 4.5	L2 dist = 4.50
D -> E	L1 dist = abs(3-5.5) + abs(3.5-3.5) = 2.5	L2 dist = 2.50
D -> F	L1 dist = abs(3-5.5) + abs(3.5-2.5) = 3.5	L2 dist = 2.69
E -> F	L1 dist = abs(5.5-5.5) + abs(3.5-2.5) = 1	L2 dist = 1.00

L1 Classification	L2 Classification
A - CLASS 1	A - CLASS 1
B – CLASS 1	B – CLASS 1
C – CLASS 1	C – CLASS 1
D – CLASS 2	D – CLASS 1
E – CLASS 2	E – CLASS 2
F – CLASS 2	F – CLASS 2

c)

The distances between the points and nearest neighbor are similar between L1 and L2 calculations. At some points, the L2 calculation gives a more conservative distance value.

All point classifications are the same except D point. L2 classifies D as 2nd which is wrong when we compare it to our label. For this experiment, L1 distance seems to be more accurate than L2. However, we can opt for L2 distance for a larger data sample since the computation is less complex hence more computationally efficient.

Problem 2: Consider a dataset with 3 classes $C = \{A, B, C\}$, with the following class distribution $N_A = 16$, $N_B = 32$, $N_C = 64$. We use unweighted k-NN classifier, and set k to be equal to the number of data points, i.e. $k = N_A + N_B + N_C =: N$.

a) What can we say about the prediction for a new point x_{new}?

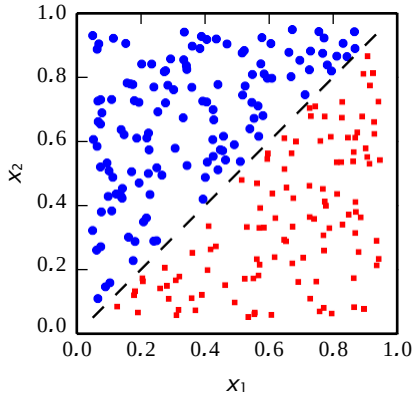
b) How about if we use the weighted (by distance) version of k-Nearest Neighbors?

a) Xnew will always be classified as C because C occurences 64 will always be the most occurring class in the 112 neighbors.

b) Xnew can be anyone of the three classes because weights are assigned to closer neighbors. For example, 10 close neighbors of class A might weigh more than 35 far away neighbors of class C hence resulting in a classification of A.

b) Decision Trees

Problem 3: The plot below shows data of two classes that can easily be separated by a single (diagonal) line. Does there exist a decision tree of depth 1 that classifies this dataset with 100% accuracy? Justify your answer.



Answer:

A decision tree of depth 1 can classify this dataset with 100% accuracy. The decision will be if x2 is larger than x1. If it is larger, the point will be classified as blue. Else, it will be classified as red.

Problem 4: You are developing a model to classify games at which machine learning will beat the world champion within five years. The following table contains the data you have collected.

No.	x_1 (Team or Individual)	x_2 (Mental or Physical)	x_3 (Skill or Chance)	y (Win or Lose)
1	T	M	S	W
2	I	M	S	W
3	T	P	S	W
4	I	P	C	W
5	T	P	C	L
6	I	M	C	L
7	T	M	S	L
8	I	P	S	L
9	T	P	C	L
10	I	P	C	L

a. Calculate the entropy $i_H(y)$ of the class labels y.

b. Build the optimal decision tree of depth 1 using entropy as the impurity measure.

a)

10 Instances

4 W – 6 L

$i_H(y)$ = - [(4/10 * log2(4/10)) + (6/10 * log2(6/10))] = 0.97

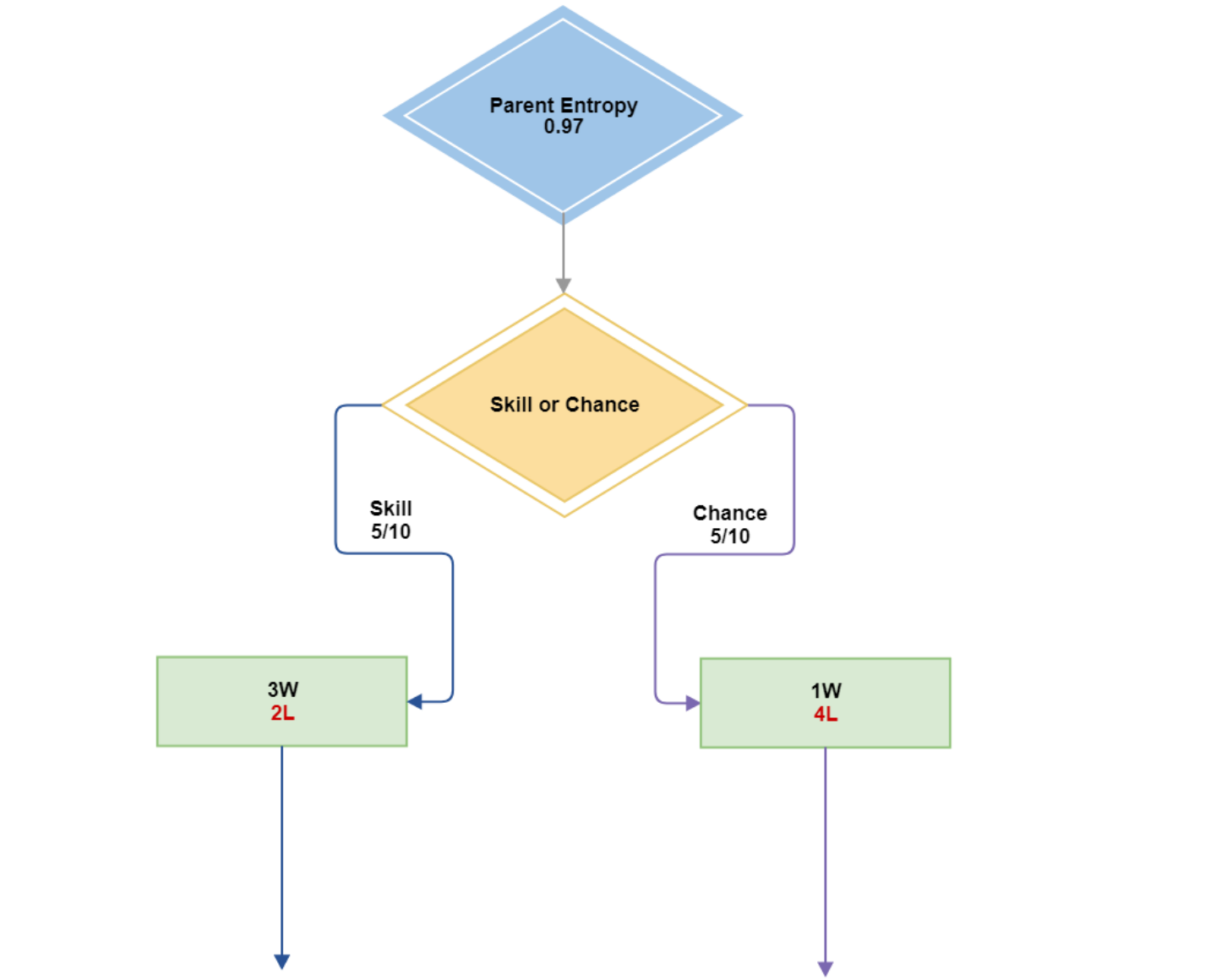
b)

$i_H(X1)$ = $i_H(y)$ - (5/10 * - [(2/5 * log2(2/5)) + (3/5 * log2(3/5))]) +
(5/10 * - [(2/5 * log2(2/5)) + (3/5 * log2(3/5))]) = 0

$i_H(X2)$ = $i_H(y)$ - (4/10 * - [(2/4 * log2(2/4)) + (2/4 * log2(2/4))]) +
(6/10 * - [(2/6 * log2(2/6)) + (4/6 * log2(4/6))]) = 0.054

Selection of decision tree based on information gain

$i_H(X3)$ = $i_H(y)$ - (5/10 * - [3/5 * log2(3/5)) + (2/5 * log2(2/5))]) +
(5/10 * - [(1/5 * log2(1/5)) + (4/5 * log2(4/5))]) = 0.155


Parent Entropy
0.97
Skill or Chance
Skill
5/10
Chance
5/10
3W
2L
1W
4L
Parent entropy - ((Weight Skill * Skiill Entropy) + (Weight Chance * Chance Entropy)) = Information Gain
0.97 - ((5/10 * 0.97) + (5/10 * 0.72)) = 0.155

Programming Task

Problem 5: Load the notebook `exercise_02_notebook.ipynb` from Piazza. Fill in the missing code and run the notebook. Convert the evaluated notebook to HTML using nbconvert and add it to your submission.

Note: We suggest that you use Anaconda for installing Python and Jupyter, as well as for managing packages. We recommend that you use Python 3.

For more information on Jupyter notebooks, consult the Jupyter documentation. Instructions for converting the Jupyter notebooks to PDF are provided within the notebook.