Grade 9 Computing

Unit 9 · Machine Learning

WORKSHEET

Full marks 33

Name _____ Class _____ Roll no. ____ Date _____ Marks _____ / 33

1 Key words [4]

Fill in the blanks with words from the box. Two words in the box are not needed. Each blank carries 0.5 mark.

baseline	datasheet	Euclidean distance	features	k nearest neighbors	label
neural network	noise	scaling	test set		

The numbers measured about each example, such as height and slope, are its _____, and the right answer, such as the belt, is its _____. A model that lets the k examples most like a new thing vote is _____. The straight-line distance worked out with the Pythagorean theorem is the _____. Turning every feature into a number from 0 to 1 before measuring is _____. The examples kept back to check a model make up the _____. The score of the simplest possible guess, such as always saying Hill, is a _____. A feature that doesn't change with the label, and only makes the model worse, is _____.

2 Multiple choice [5]

Choose the correct option for each of the following questions.

i. A model uses the 3 nearest neighbors. A new district's neighbors are Hill, Mountain and Mountain. What does the model say? [1]

a. Hill, because the nearest one is Hill

b. Mountain, because two of the three are Mountain

c. It can't decide, because the votes are split

d. Hill and Mountain, both

ii. Heights in a dataset run from 100 m to 5,100 m. What is a height of 2,600 m after min–max scaling? [1]

a. 0.26 **b.** 0.5

c. 2.6 **d.** 0.51

iii. Adding the rain feature made the district model worse. Why? [1]

a. Rain is measured in millimeters.

b. Rain doesn't change much with the belt, so it adds gaps to the distances that have nothing to do with the label.

c. Models can only use two features.

d. The rain data was deleted.

iv. A model is right for 70 of 77 districts. A "model" that always says Hill is right for 40. What does the comparison show? [1]

a. The model is no better than guessing.

b. The model has learned something: it beats the simplest guess by a lot.

c. 70 of 77 is too low to be useful.

d. Always saying Hill is a good model.

v. Which task fits a **neural network** best? **[1]**

a. Working out a bus fare from a fare table

b. Guessing a district's belt from its height and slope

c. Recognizing which of 50 crop diseases a leaf has, from a photo

d. Adding up a shop's bill

3 Calculations [3]

Show your working.

Copy and complete this confusion table for a model tested on 30 districts, then find its accuracy and say which mistake it makes most.

True belt	said Mountain	said Hill	said Terai
Mountain	5	2	0
Hill	3	13	1
Terai	0	0	?

Answer: _____

4 What does it print? [3]

Write exactly what the program prints, in the box under it.

Each example is a district's scaled height and slope, and its belt.

```
1  import math
2
3  def distance(a, b):
4      return math.sqrt((a[0] - b[0]) ** 2 + (a[1] - b[1]) ** 2)
5
6  new = (0.4, 0.7)
7  examples = [(0.7, 0.3, "Hill"), (0.1, 0.8, "Mountain"), (0.4, 0.3, "Terai")]
8  pairs = []
9  for x, y, belt in examples:
10     pairs.append((round(distance(new, (x, y)), 2), belt))
11 pairs.sort()
12 print(pairs)
13 print("k = 1:", pairs[0][1])
```

```
OUTPUT
```

5 Trace the program [4]

Fill in the trace table: one row for each time round the first loop, then what the program prints. The first row shows the values before the loop.

`pairs` holds the five nearest examples' distances and belts, already sorted.

```
pairs = [(0.10, "Hill"), (0.17, "Terai"), (0.24, "Terai"), (0.24, "Hill"),
(0.25, "Terai")]
k = 5
votes = {}
for d, belt in pairs[:k]:
    votes[belt] = votes.get(belt, 0) + 1
best = pairs[0][1]
for belt in votes:
    if votes[belt] > votes[best]:
        best = belt
print(votes, best)
```

d	belt	votes	**Output**
(before)		{}	

6 Find and fix the bug [3]

Read the program, then answer.

`read_districts()`, `scale()` and `predict()` are the unit's functions. This leave-one-out test prints `77 of 77 right`.

```
rows = read_districts("districts.csv")
scale(rows, ["height", "slope"])
features = ["height_s", "slope_s"]
right = 0
for i in range(len(rows)):
    guess = predict(rows[i], rows, features, 1)
    if guess == rows[i]["belt"]:
        right = right + 1
print(right, "of", len(rows), "right")
```

i. Why is 77 of 77 too good to be true?

ii. Which line is wrong? Write it, and the line before it, correctly.

7 Write a program [4]

Write the program in Python on the lines. Any program that gives the right result gets the marks.

`labels` is a list of the true belts of a test set of 20 districts. Write a program that counts each belt with a dictionary, finds the commonest, and prints the baseline: how often a model that always says that belt is right, as a percentage. For a test set of 12 Hill, 5 Terai and 3 Mountain districts it should print `Always Hill: 60%`.

8 Short answers [5]

Write short answers to the following questions.

a. A student builds a nearest-neighbor model with two features: a person's height in centimeters (140 to 190) and their monthly pocket money in rupees (0 to 5,000). What will go wrong, and how is it fixed? [2]

b. A bank in Nepal wants to decide who gets a loan with a model trained on its last 20 years of loans. Write three questions about the data the bank should ask, and why each matters. **[3]**

9 Think **[2]**

Think, then answer.

A model that reads X-rays and says "healthy" or "needs a doctor" is 95% accurate, the same as the district model. Should it be used the same way as the district model? Give your reasons.

10 Activity: Seventy-seven districts (not marked)

At a computer or phone, open the unit's page (the QR code) and find the activity **Seventy-seven districts**. Test the 15 test districts with 1, 3 and 5 neighbors. Then press **New random split** three times, testing with 3 neighbors each time.

Test	Neighbors (k)	Right out of 15	Districts it got wrong
First split	1		
First split	3		
First split	5		
New split 1	3		
New split 2	3		
New split 3	3		