CS 260: Foundations of Data Science

Prof. Thao Nguyen

Fall 2025

HAVERFORD COLLEGE

Materials by Sara Mathieson

Practice Midterm 1 Solutions

#1-3

1. multi-class classification (discrete)

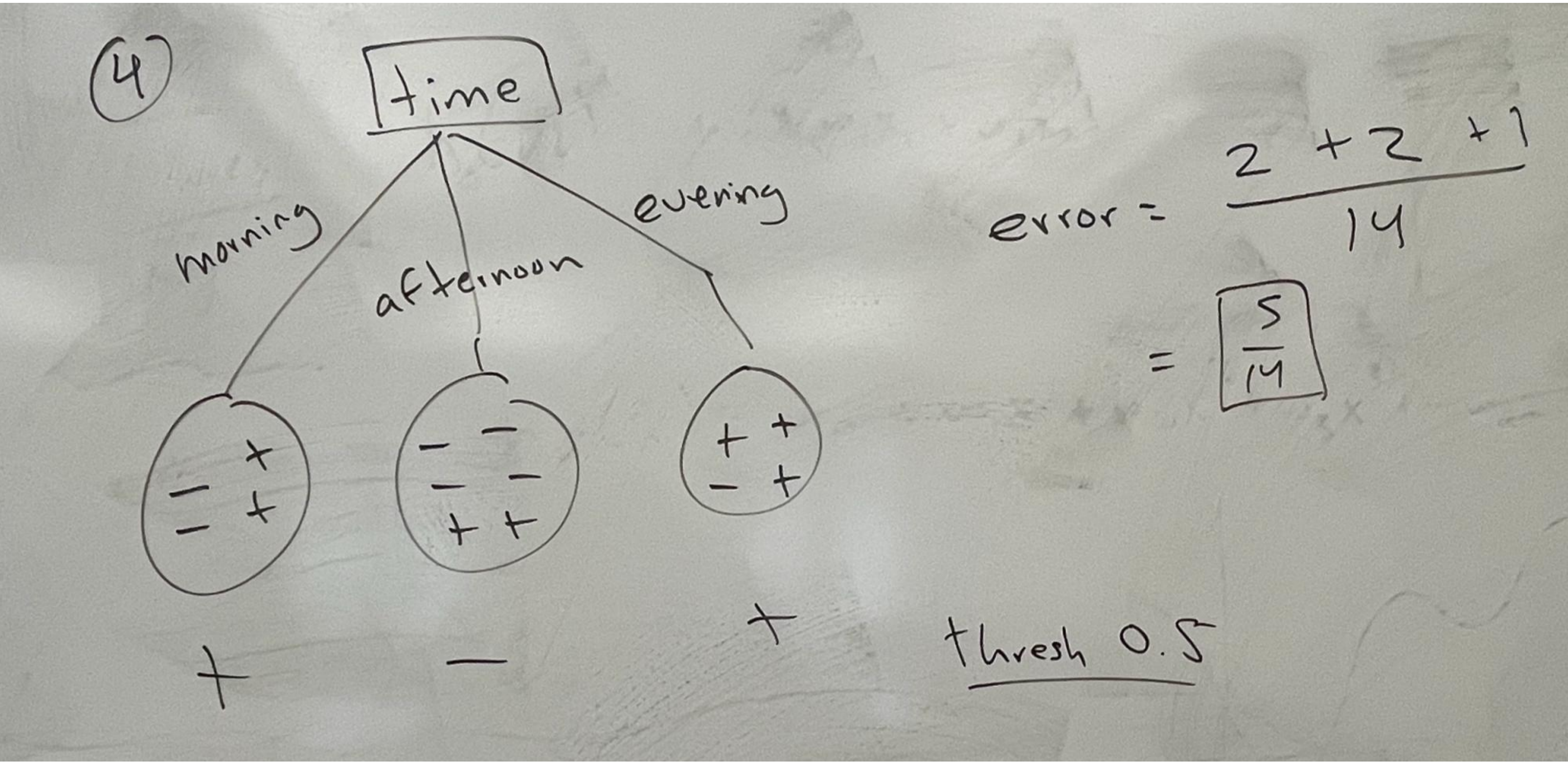
label

2. regression (continuous)

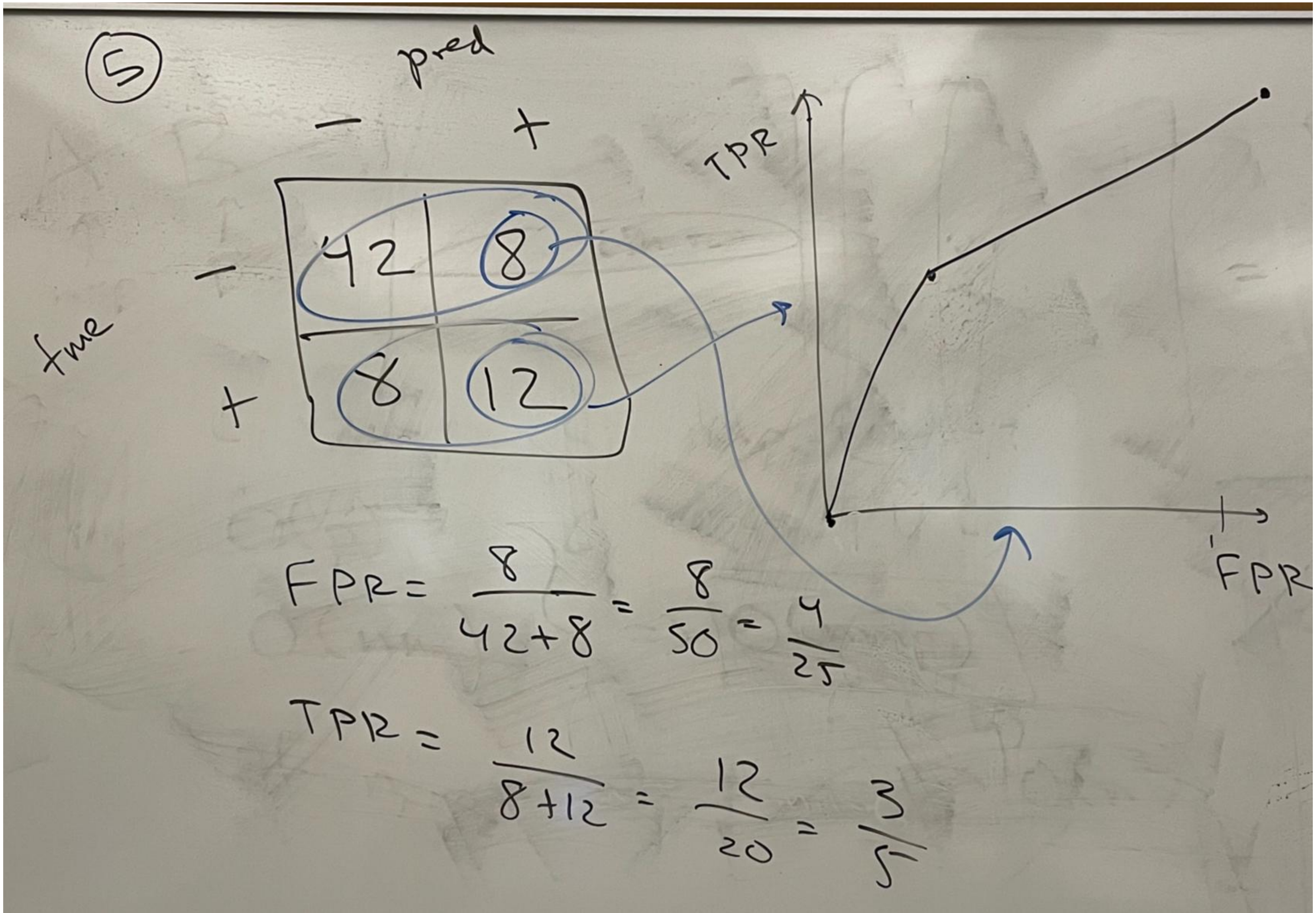
3.

internal nodes	class labels
branches	feature names
leaves	feature values

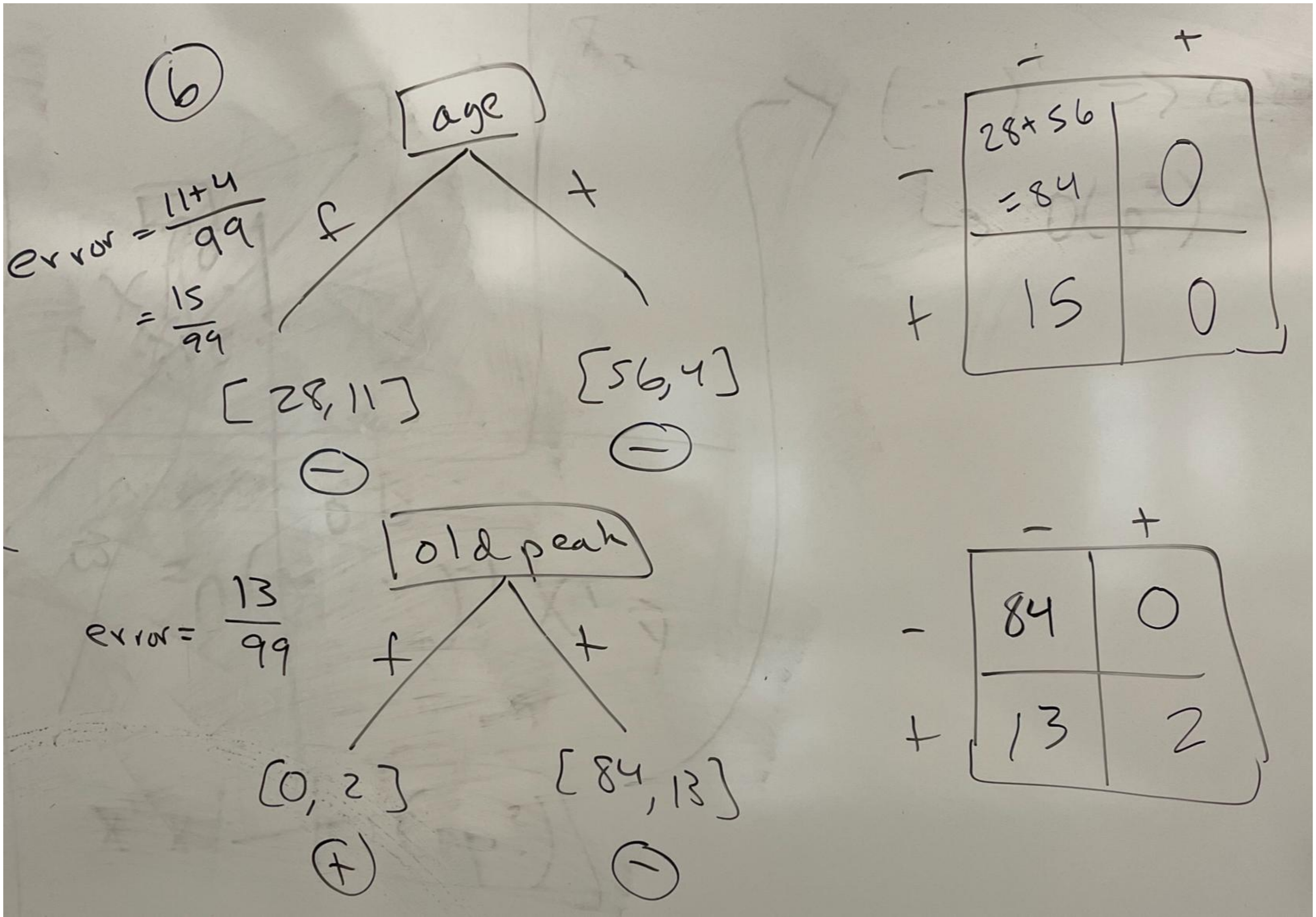
#4


(4)
time
morning
afternoon
evening
error = $\frac{2+2+1}{14}$
= $\boxed{\frac{5}{14}}$
+
−
+
thresh 0.5

#5

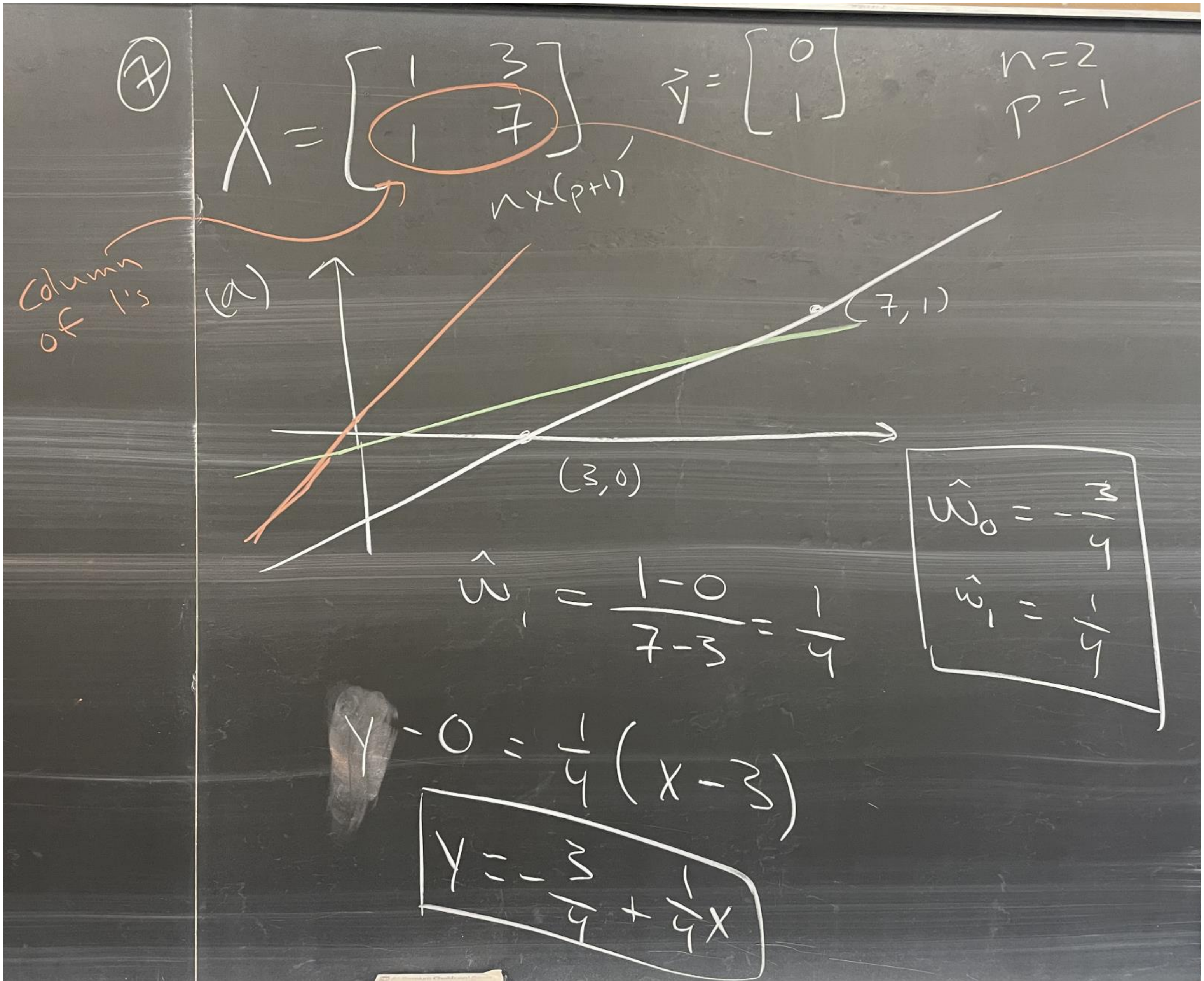


#6



6
age
f
t
error = 11+4 / 99
= 15 / 99
[28, 11]
[56, 4]
−
−
28+56
=84
0
15
0
oldpeak
error = 13 / 99
f
t
[0, 2]
[84, 13]
+
−
84
0
13
2

#7a



7
X = [1 3; 1 7]
n x (p+1)
y = [0; 1]
n=2
p=1
column of 1's
(a)
(7, 1)
(3, 0)
ŵ₀ = -3/4
ŵ₁ = 1/4
ŵ₁ = (1-0)/(7-3) = 1/4
y - 0 = 1/4 (x - 3)
y = -3/4 + 1/4 x

#7b, c

(7, 1)

(b)
$$\begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \leftarrow \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \alpha \Bigg(\underbrace{\overbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}^{\vec{w}} \cdot \overbrace{\begin{bmatrix} 1 \\ 7 \end{bmatrix}}^{\vec{x}_2}}_{0 \cdot 1 + 0 \cdot 7 = 0} - \underbrace{1}_{y_2} \Bigg) \begin{bmatrix} 1 \\ 7 \end{bmatrix}$$

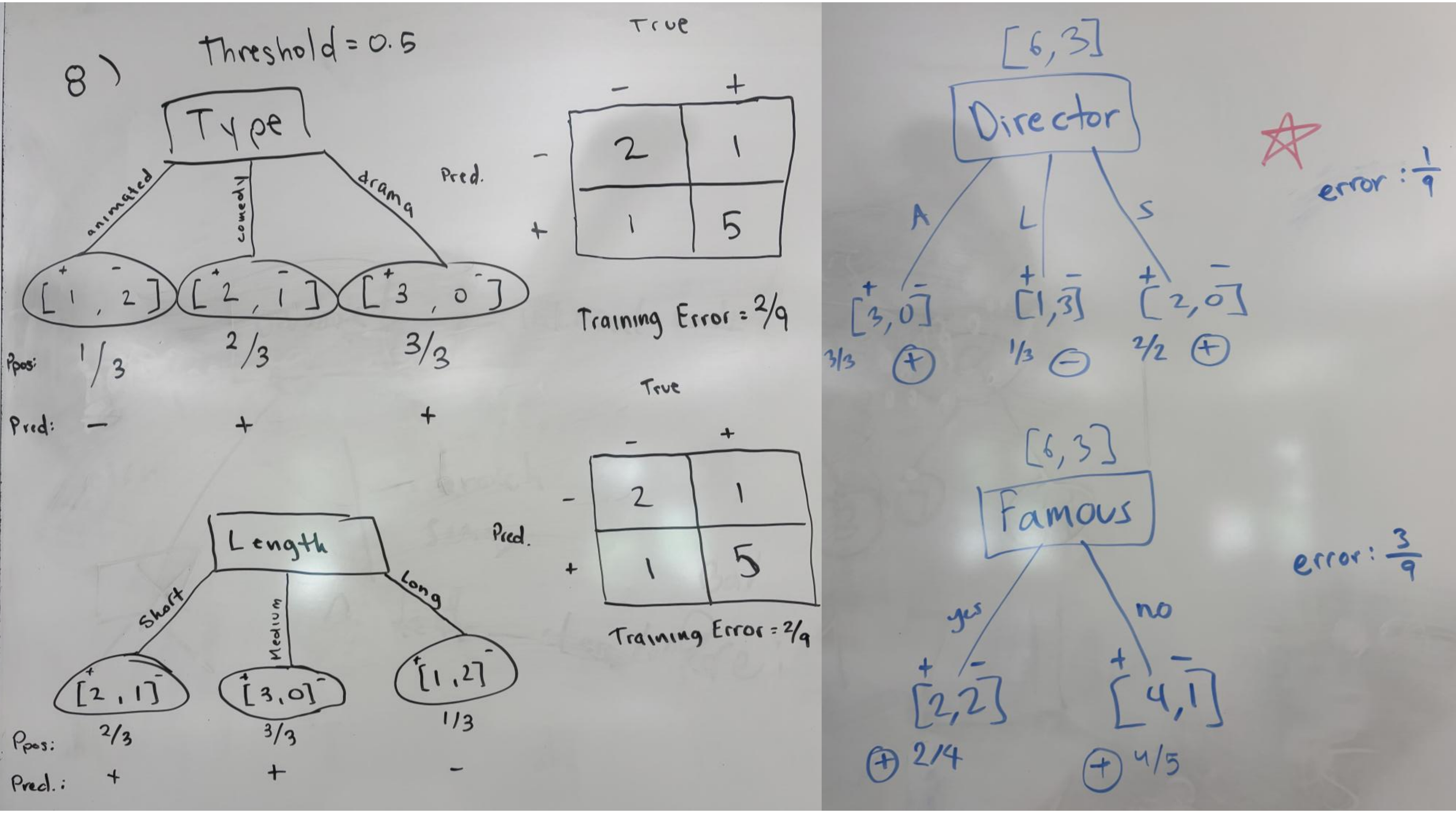
$$\leftarrow \begin{bmatrix} 0.1 \\ 0.7 \end{bmatrix}$$

(c)
$$\begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \leftarrow \begin{bmatrix} 0.1 \\ 0.7 \end{bmatrix} - 0.1 \Bigg(\begin{bmatrix} 0.1 \\ 0.7 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3 \end{bmatrix} - 0 \Bigg) \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

$$\underbrace{0.1 \cdot 1 + 0.7 \cdot 3}_{z_2}$$

$$\leftarrow \begin{bmatrix} -0.12 \\ 0.04 \end{bmatrix}$$

#8


Threshold = 0.5
8)
Type
animated
comedy
drama
[1, 2]
[2, 1]
[3, 0]
Ppos: 1/3 2/3 3/3
Pred: − + +
True
Pred.
− +
2 1
1 5
Training Error = 2/9
Length
Short
Medium
Long
[2, 1]
[3, 0]
[1, 2]
Ppos: 2/3 3/3 1/3
Pred.: + + −
True
Pred.
− +
2 1
1 5
Training Error = 2/9
[6, 3]
Director
A
L
S
[3, 0]
[1, 3]
[2, 0]
3/3 +
1/3 −
2/2 +
error: 1/9
[6, 3]
Famous
yes
no
[2, 2]
[4, 1]
+ 2/4
+ 4/5
error: 3/9

#9-10

```
def matrix_multiply(A, B):
    m, n = A.shape
    p, q = B.shape
    if n != p:
        raise ValueError(“inner dimensions must match”)
    result = np.zeros((m,q))
    for i in range(m):
        for j in range(q):
            val = 0
            for k in range(n):
                val += A[i,k]*B[k,j]
            result[i,j] = val
    return result
```

$O(m^3)$

$A[i] \cdot B[j]$