

CS 260: Foundations of Data Science

Prof. Thao Nguyen

Fall 2026



HAVERFORD
COLLEGE

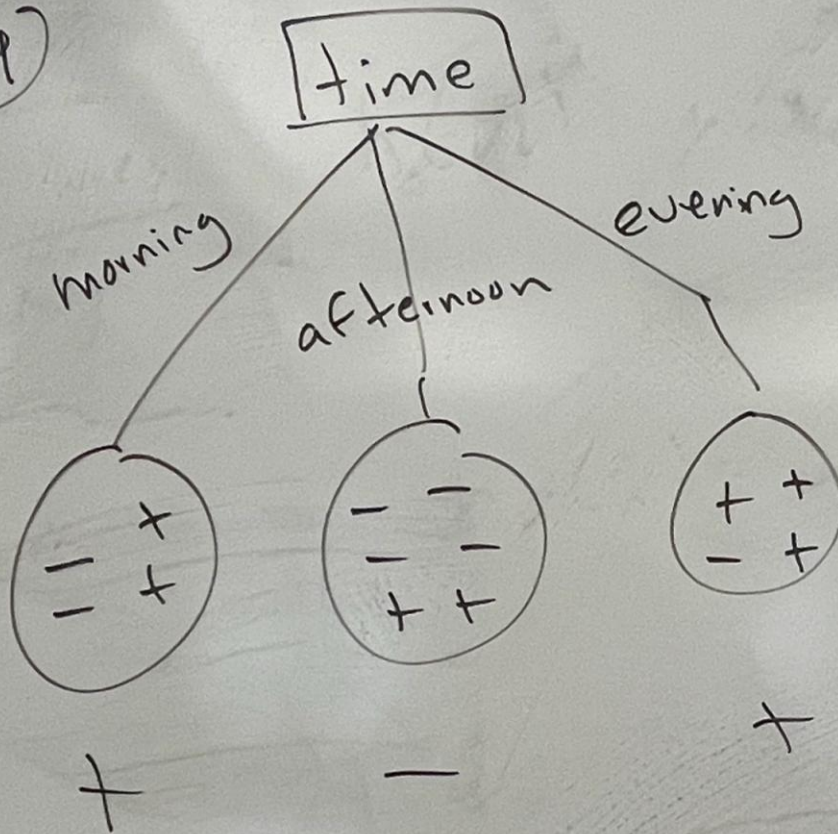
Practice Midterm 1 Solutions

#1-3

- ① multi-class classification (discrete)
label
- ② regression (continuous)
- ③ internal nodes — class labels
branches — feature names
leaves — feature values

#4

(4)



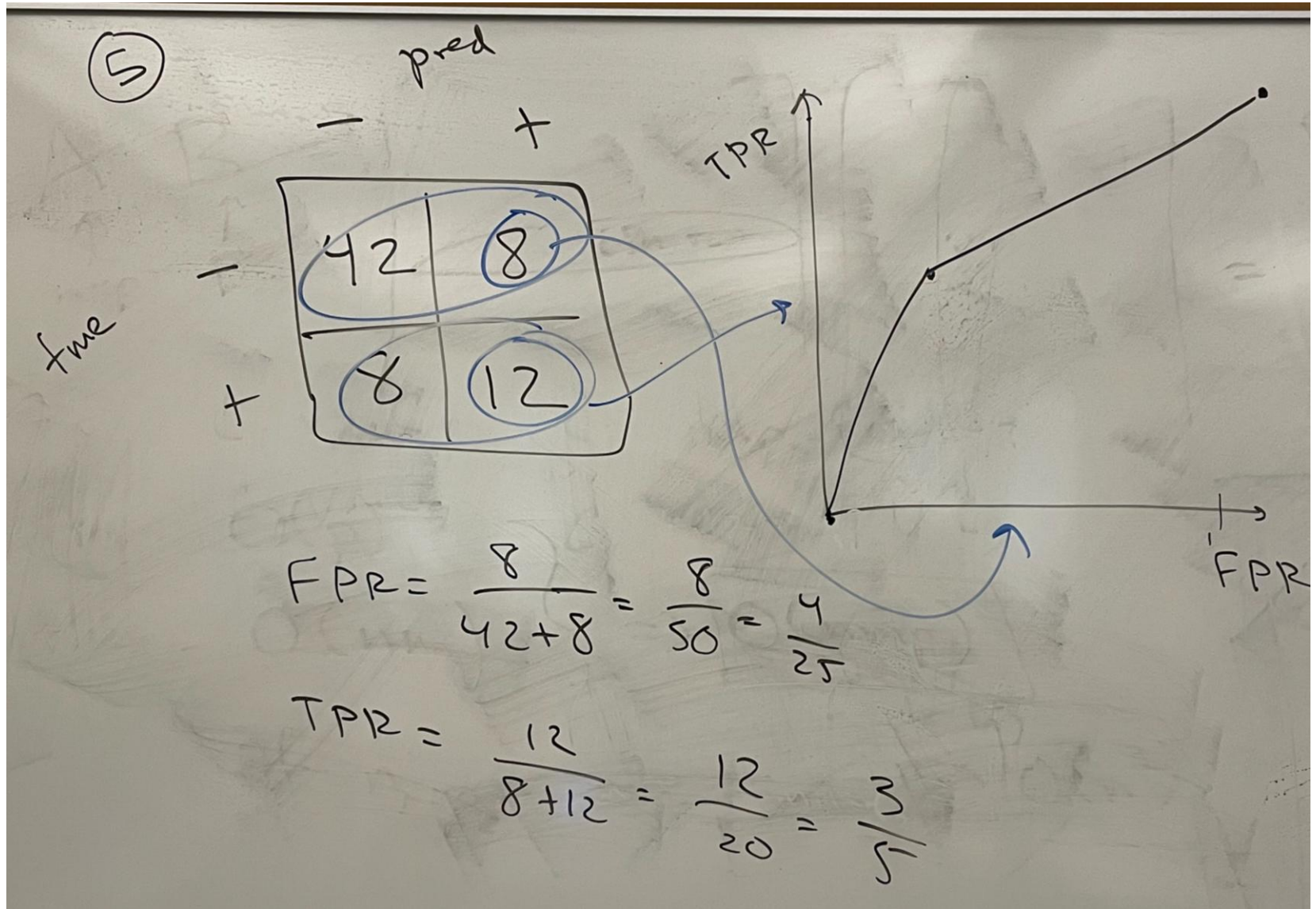
$$\text{error} = \frac{2 + 2 + 1}{14}$$

$$= \boxed{\frac{5}{14}}$$

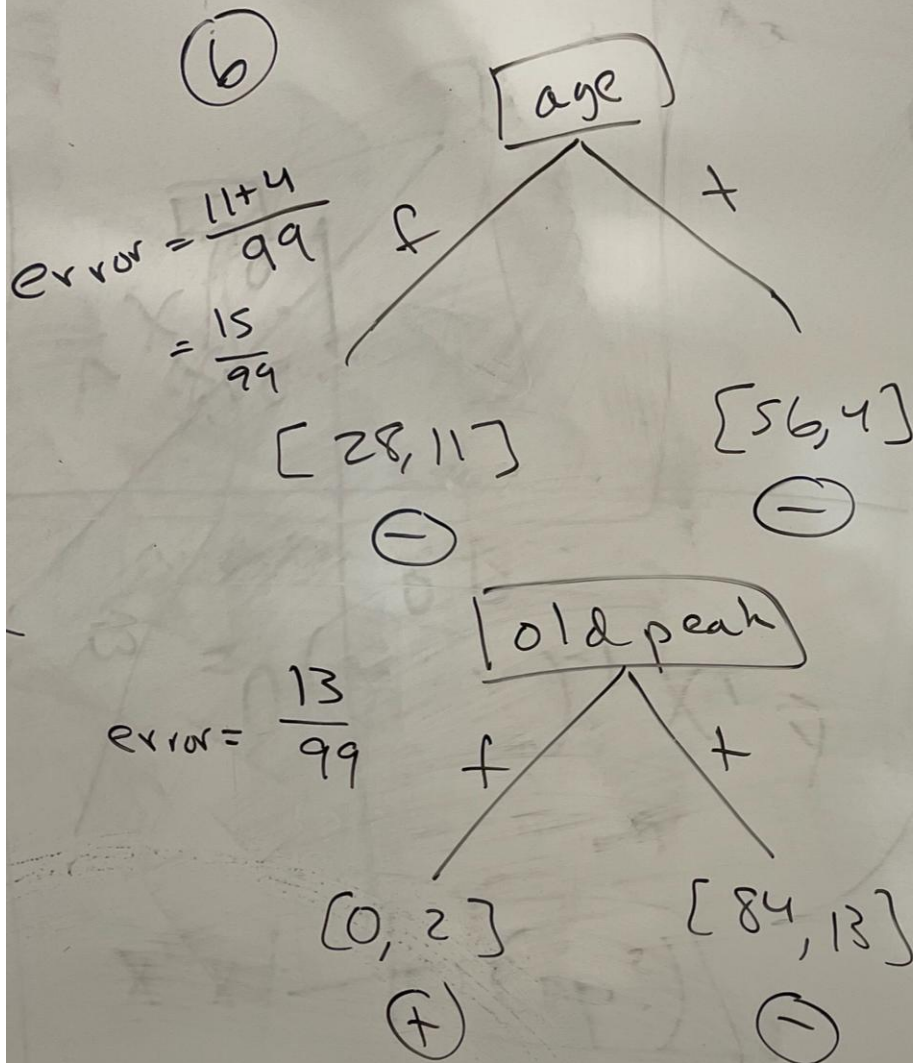
thresh 0.5

#5

a) The model made 8 false negative predictions, which means it missed 40% of the test subjects who have diabetes, who may then not receive treatment.



#6



	-	+
-	28+56 =84	0
+	15	0

	-	+
-	84	0
+	13	2

#7a

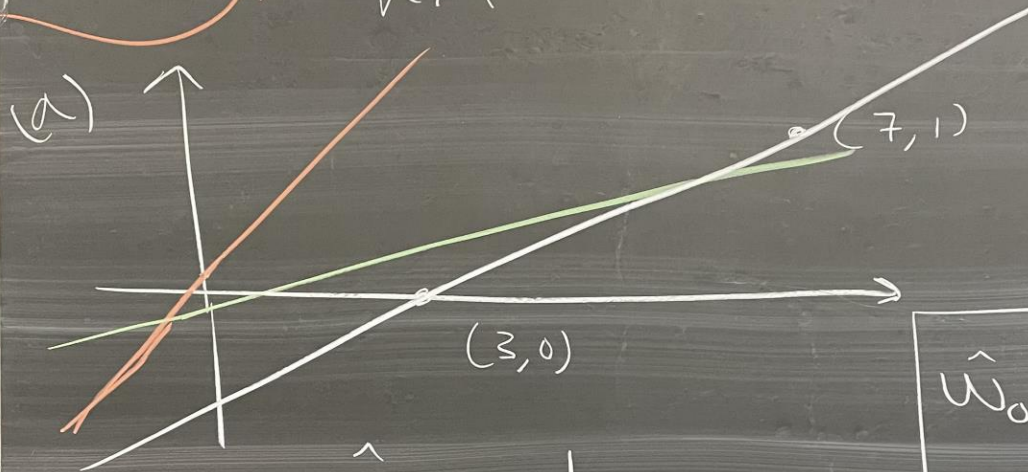
⑦

$$X = \begin{bmatrix} 1 & 3 \\ 1 & 7 \end{bmatrix}, \quad \vec{y} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$n \times (p+1)$

$$n=2 \\ p=1$$

Column
of 1's



$$\hat{\omega}_1 = \frac{1-0}{7-3} = \frac{1}{4}$$

$$\begin{bmatrix} \hat{\omega}_0 = -1/5 \\ \hat{\omega}_1 = -1/5 \end{bmatrix}$$

$$y - 0 = \frac{1}{4}(x - 3)$$

$$y = -\frac{3}{4} + \frac{1}{4}x$$

#7b, c

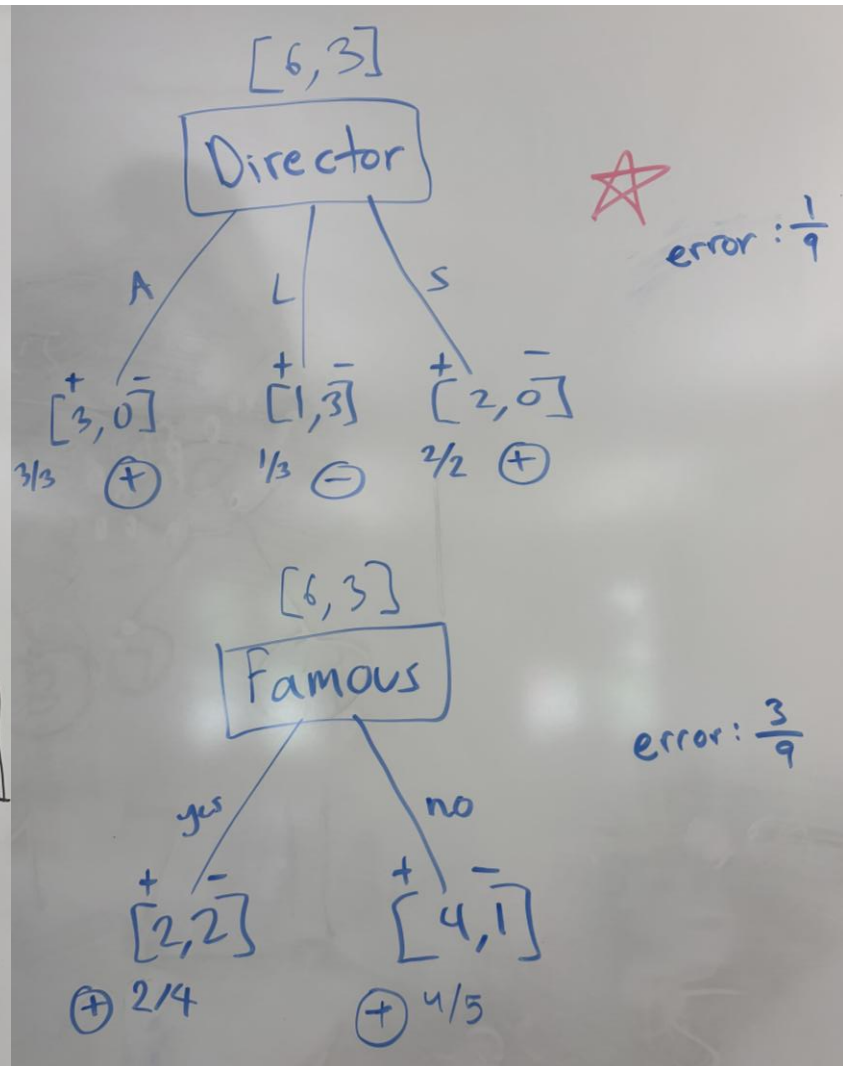
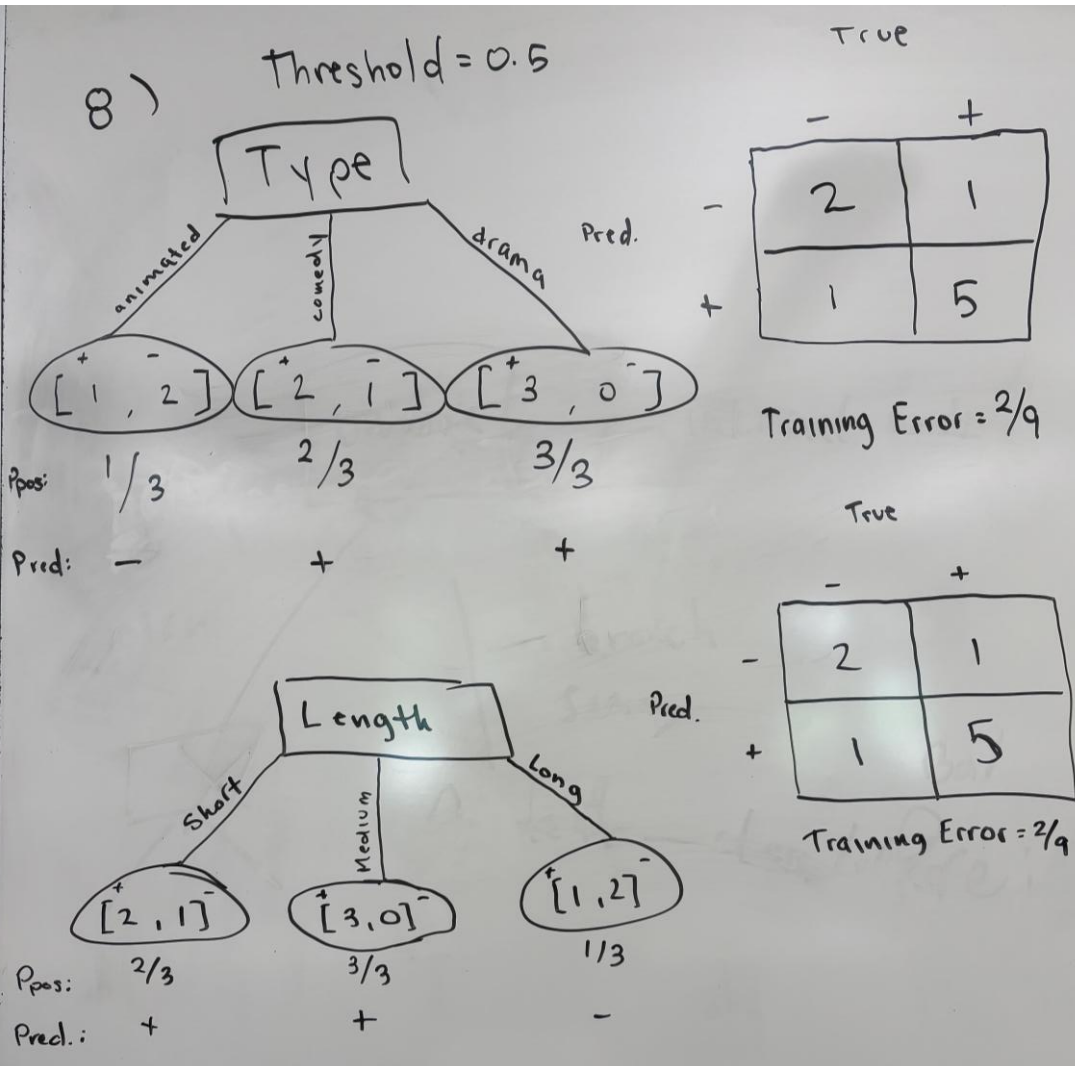
(7, 1)

$$(b) \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \leftarrow \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \underbrace{q \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 7 \end{bmatrix} - 1 \right)}_{0.1 + 0.7 = 0} \underbrace{\begin{bmatrix} 1 \\ 7 \end{bmatrix}}_{x_2}$$
$$\leftarrow \begin{bmatrix} 0.1 \\ 0.7 \end{bmatrix}$$

$$(c) \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \leftarrow \begin{bmatrix} 0.1 \\ 0.7 \end{bmatrix} - 0.1 \left(\begin{bmatrix} 0.1 \\ 0.7 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3 \end{bmatrix} - 0 \right) \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$
$$\leftarrow \begin{bmatrix} -0.12 \\ 0.04 \end{bmatrix}$$

$0.1 \cdot 1 + 0.7 \cdot 3$
 2.2

#8



#9-10

```
def matrix_multiply(A, B):  $O(m^3)$ 
    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
```

