

CS 260: Foundations of Data Science

Prof. Sara Mathieson

Spring 2025



Haverford
COLLEGE

Admin

- Sit somewhere new!
- Lab 1 due tomorrow (finish in lab today)
 - If you're *completely* finished with no questions, you do not have to attend lab today (but email me)
- Lab 2 posted tomorrow (start ASAP!)

TA Hour / Office Hour Schedule

Mondays	3:45-5pm	Sara
Mondays	8-10pm	Dan
Wednesdays	6-8pm	Edgar
Thursdays	8-10pm	Rachel

Peer Tutors

- Darshan Mehta
- Fejiro Anigboro
- Lei Lei

Recap Class 2 – die example

```
class Die:

    def __init__(self, num_sides):
        """Construct a new die with the given number of sides."""
        self.sides = num_sides
        self.value = 1 # default starting value

    def get_value(self):
        """Getter for the die's current value."""
        return self.value

    def roll(self):
        """Choose a new random value for the die, i.e. roll it."""
        self.value = random.randrange(1,self.sides+1)

    def __str__(self):
        """String representation of the die (with current value)."""
        return f"{self.sides}-sided die, current value: {self.value}"
```

Recap Class 2 – die example

```
def main():  
    # create 8-sided dice  
    die1 = Die(8)  
    die2 = Die(8)  
  
    # roll both until we get the same value  
    same = False  
    while not same:  
        die1.roll()  
        die2.roll()  
        print(die1)  
        print(die2)  
        print()  
        # check if the values are the same  
        same = (die1.get_value() == die2.get_value())  
  
    print("Rolled the same value!")
```

Class 2 feedback forms

- Understand well: python in general, some OOP concepts (and translating to Python okay)
- New to
 - matplotlib
 - numpy (especially concatenation)
 - Git (push/pull)
 - Command line

Numpy concatenation example

```
a = [[3,4,2],[7,8,9],[2,1,0]]  
b = [[4,9,7],[3,0,1],[3,8,4]]
```

```
a_arr = np.array(a)  
b_arr = np.array(b)
```

```
>>> many_rows.shape  
(6, 3)  
>>> many_cols.shape  
(3, 6)
```

```
>>> a_arr  
array([[3, 4, 2],  
       [7, 8, 9],  
       [2, 1, 0]])  
>>> b_arr  
array([[4, 9, 7],  
       [3, 0, 1],  
       [3, 8, 4]])
```

```
>>> many_rows = np.concatenate((a_arr,b_arr), axis=0)  
>>>  
>>> many_rows  
array([[3, 4, 2],  
       [7, 8, 9],  
       [2, 1, 0],  
       [4, 9, 7],  
       [3, 0, 1],  
       [3, 8, 4]])
```

```
>>> many_cols = np.concatenate((a_arr,b_arr), axis=1)  
>>>  
>>> many_cols  
array([[3, 4, 2, 4, 9, 7],  
       [7, 8, 9, 3, 0, 1],  
       [2, 1, 0, 3, 8, 4]])
```

Outline

- Data representation and featurization
- Introduction to modeling
- Why are models useful?
- Begin: linear models

Outline

- Data representation and featurization
- Introduction to modeling
- Why are models useful?
- Begin: linear models

Tennis Data

Day	Outlook	Temperature	Humidity	Wind	PlayTennis (y)
x_1	Sunny	Hot	High	Weak	No
x_2	Sunny	Hot	High	Strong	No
x_3	Overcast	Hot	High	Weak	Yes
x_4	Rain	Mild	High	Weak	Yes
x_5	Rain	Cool	Normal	Weak	Yes
x_6	Rain	Cool	Normal	Strong	No
x_7	Overcast	Cool	Normal	Strong	Yes
x_8	Sunny	Mild	High	Weak	No
x_9	Sunny	Cool	Normal	Weak	Yes
x_{10}	Rain	Mild	Normal	Weak	Yes

Data from Machine Learning by Tom Mitchell (Table 3.2)

- Input or **features**: outlook, temp, humidity, wind
- Output or “**label**”: play tennis (yes or no)

Sea Ice data (Lab 2)

Year **Sea Ice Extent***

1996	7.88
1997	6.74
1998	6.56
1999	6.24
2000	6.32
2001	6.75
2002	5.96
2003	6.15
2004	6.05
2005	5.57
2006	5.92
2007	4.3
2008	4.63

- Input or **feature**: year
- Output or “**label**”: sea ice

*Arctic sea ice extend (1,000,000 sq km)

Data Representation Notation

Data Representation

X matrix = $\left[\begin{array}{c|c} \text{name} & \text{year} & \text{Id} & \text{classes} \end{array} \right]$

$\left[\text{---} \quad \vec{x}_i^T \quad \text{---} \right]$

n examples

p features

$n \times p$

Usually : want to model y as some function of x

label/output

$\vec{y} = \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right]$

n

$n \times 1$

i.e. major

Feature Terminology

- *Features*: feature names
 - i.e. shape
 - i.e. sea ice extent
- *Feature values*: what values are possible
 - i.e. {circle, square, triangle}
 - i.e. all non-negative values
- *Feature vector*: values for a particular example
 - i.e. $\mathbf{x} = [x_1, x_2, x_3, \dots, x_p]$

• Regression : $y \in \mathbb{R}$

• Binary classification :

• Multi-class classification

Humidity $\in \{\text{normal, high}\}$
 \Downarrow \Downarrow
 0 1

Outlook $\in \{\text{sunny, overcast, rain}\}$
 \Downarrow \Downarrow \Downarrow
 0 1 2

(continuous)

$y \in \{0, 1\}$

$y \in \{1, 2, \dots, K\}$ (image recognition)

Shape $\in \{0, \Delta, \square\}$
 \Downarrow \Downarrow \Downarrow
 0 1 2

	is 0?	is Δ ?	is \square ?
\square	0	0	1
Δ	0	1	0
Δ	0	1	0

$n=3$ { binary

Featurization: make numerical

Featurization: make numerical

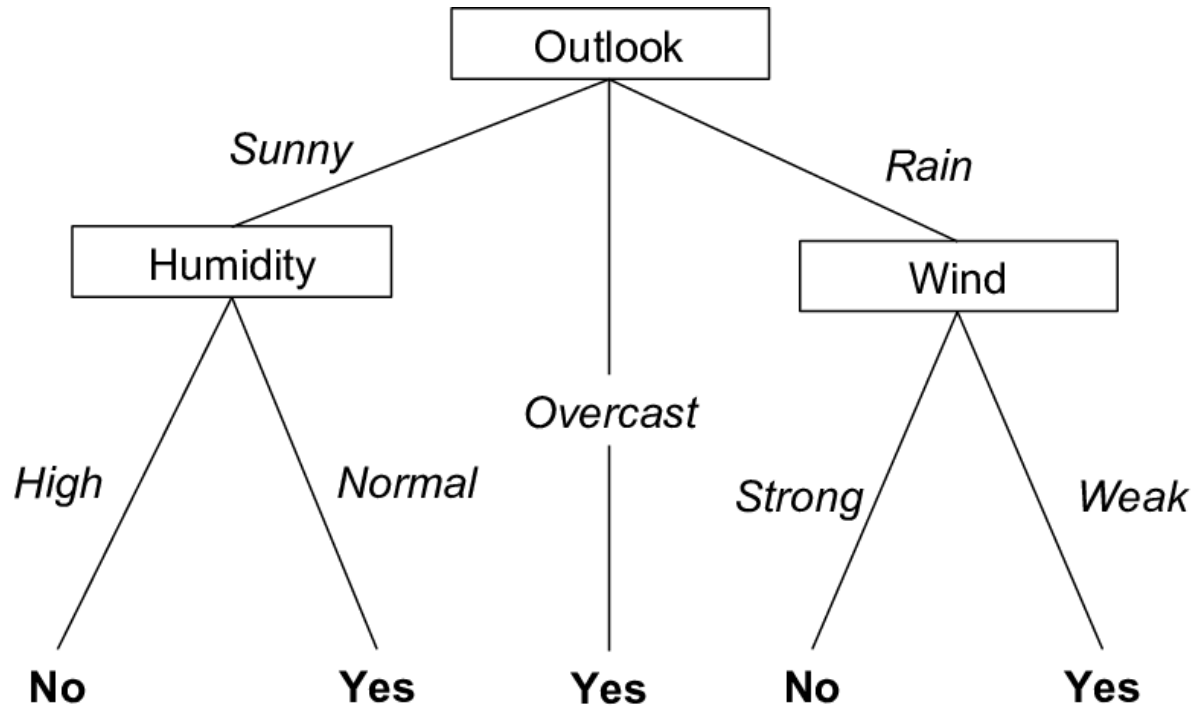
- Real-valued features get copied directly. *Duame, Chap 3*
- Binary features become 0 (for false) or 1 (for true).
- Categorical features with V possible values get mapped to V -many binary indicator features.

Q: what about features that might already be on a spectrum
(i.e. sunny, rain, overcast)?

Outline

- Data representation and featurization
- **Introduction to modeling**
- Why are models useful?
- Begin: linear models

Example of a model



- Each internal node: one feature
- Each branch from node: selects one value of the feature
- Each leaf node: predict y

Model Examples

What is a model?

(informal) way of explaining phenomenon through data

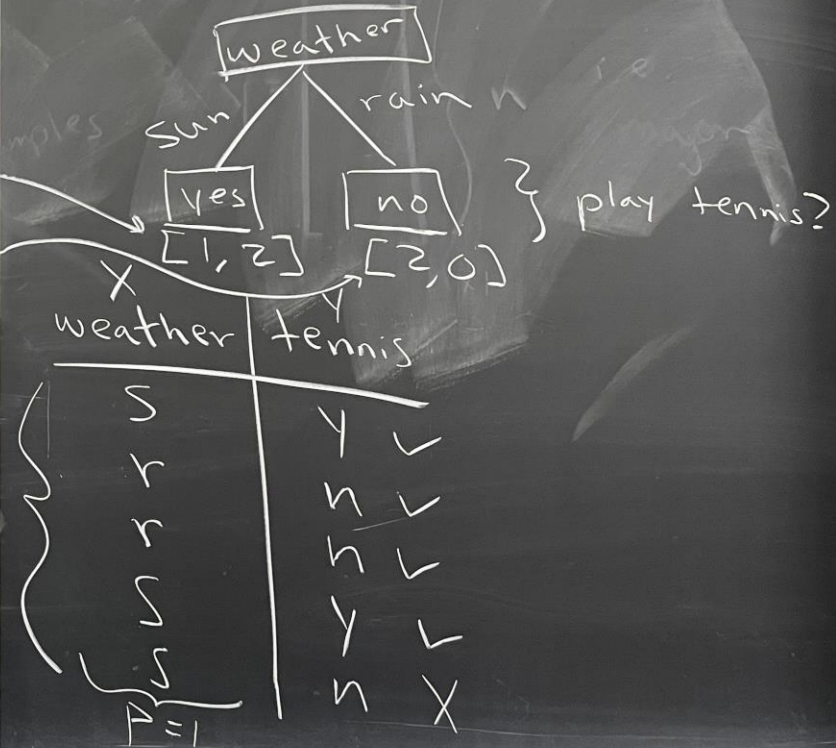
(formal) distribution that captures our data

accuracy: 67%

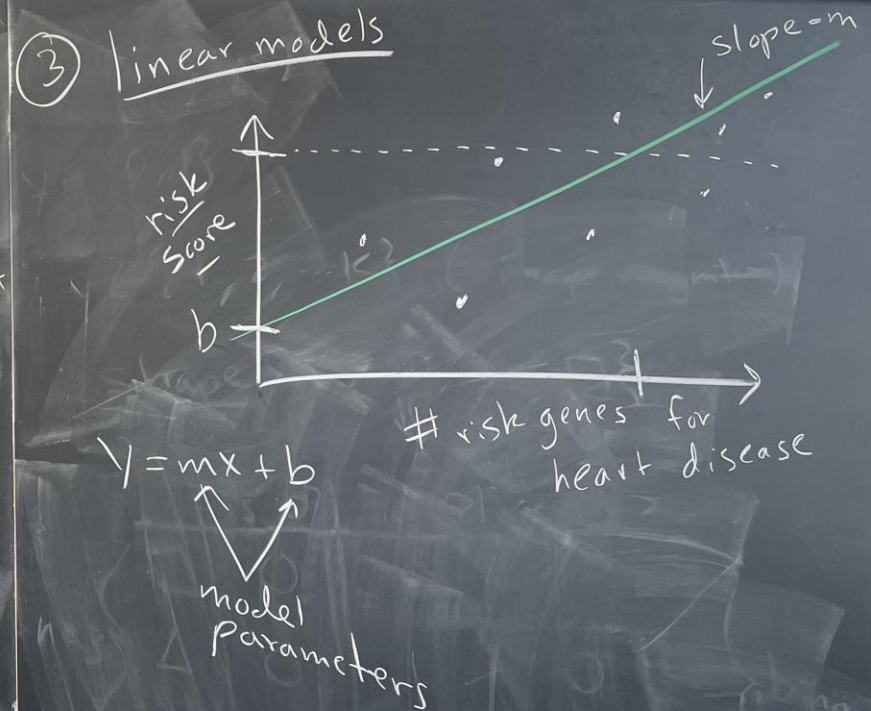
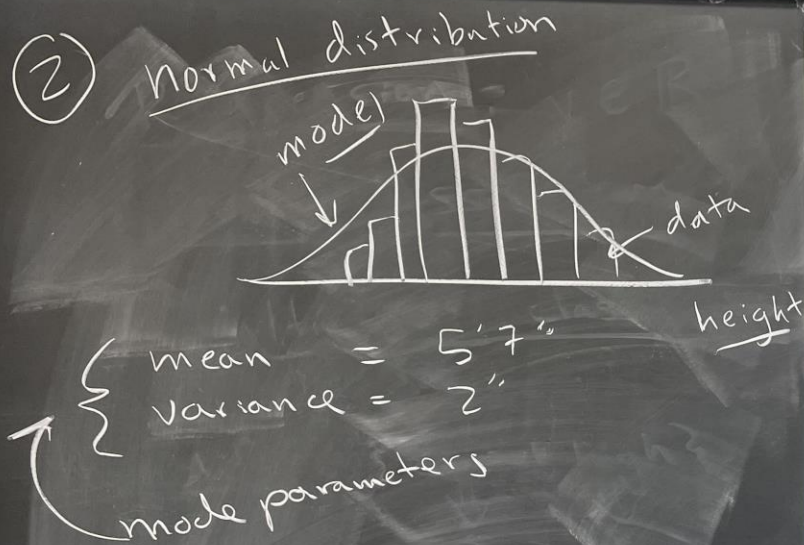
$$\Rightarrow \frac{4}{5} = 80\% \text{ accurate overall}$$

100%
n=5

① decision tree



Model Examples



Handout 3 (find your random partner)

Handout 3

Q1: $n=10, p=4$

Day	Outlook	Temperature	Humidity	Wind	PlayTennis (y)
x_1	Sunny	Hot	High	Weak	No
x_2	Sunny	Hot	High	Strong	No
x_3	Overcast	Hot	High	Weak	Yes
x_4	Rain	Mild	High	Weak	Yes
x_5	Rain	Cool	Normal	Weak	Yes
x_6	Rain	Cool	Normal	Strong	No
x_7	Overcast	Cool	Normal	Strong	Yes
x_8	Sunny	Mild	High	Weak	No
x_9	Sunny	Cool	Normal	Weak	Yes
x_{10}	Rain	Mild	Normal	Weak	Yes

Q2

Sunny: {0,1}
 Overcast: {0,1}
 Rain: {0,1}
 Temperature: {0, 1, 2} (Cool, Mild, Hot)
 Humidity: {0,1} (Normal, High)
 Wind {0,1} (Weak, Strong)

Data from Machine Learning by Tom Mitchell (Table 3.2)

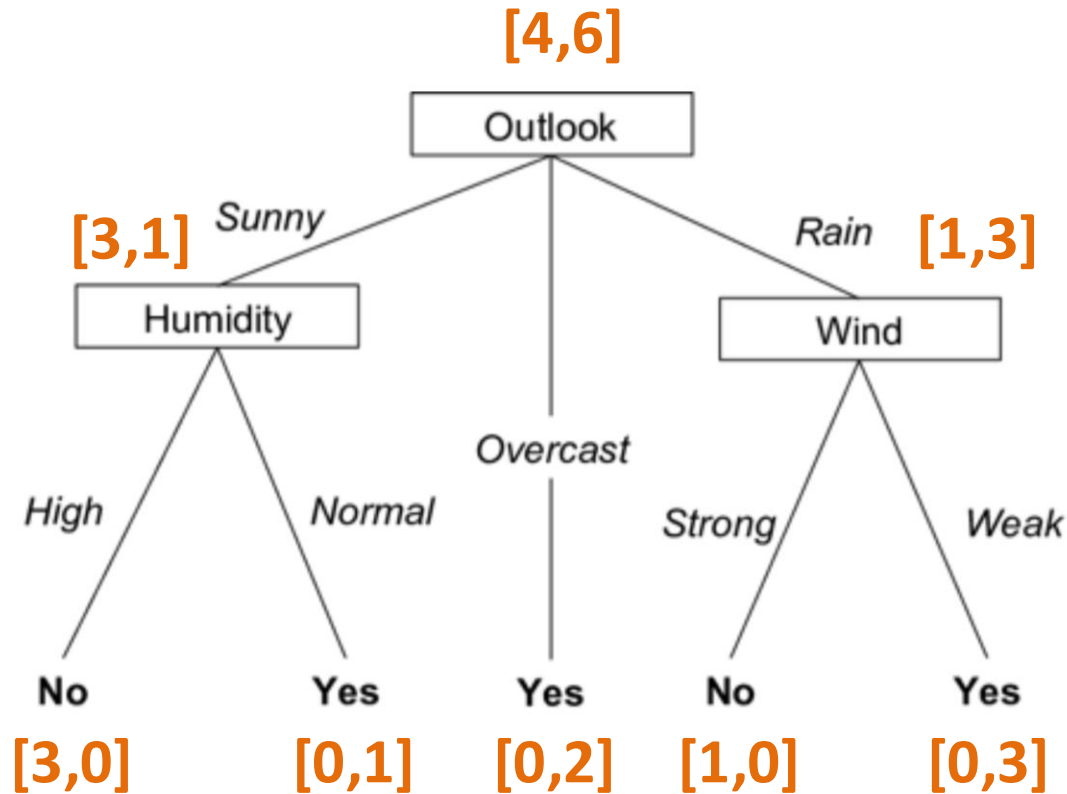
Q3

	Sunny	Overcast	Rain	Temp	Humidity	Wind
x_1	1	0	0	2	1	0

Handout 3

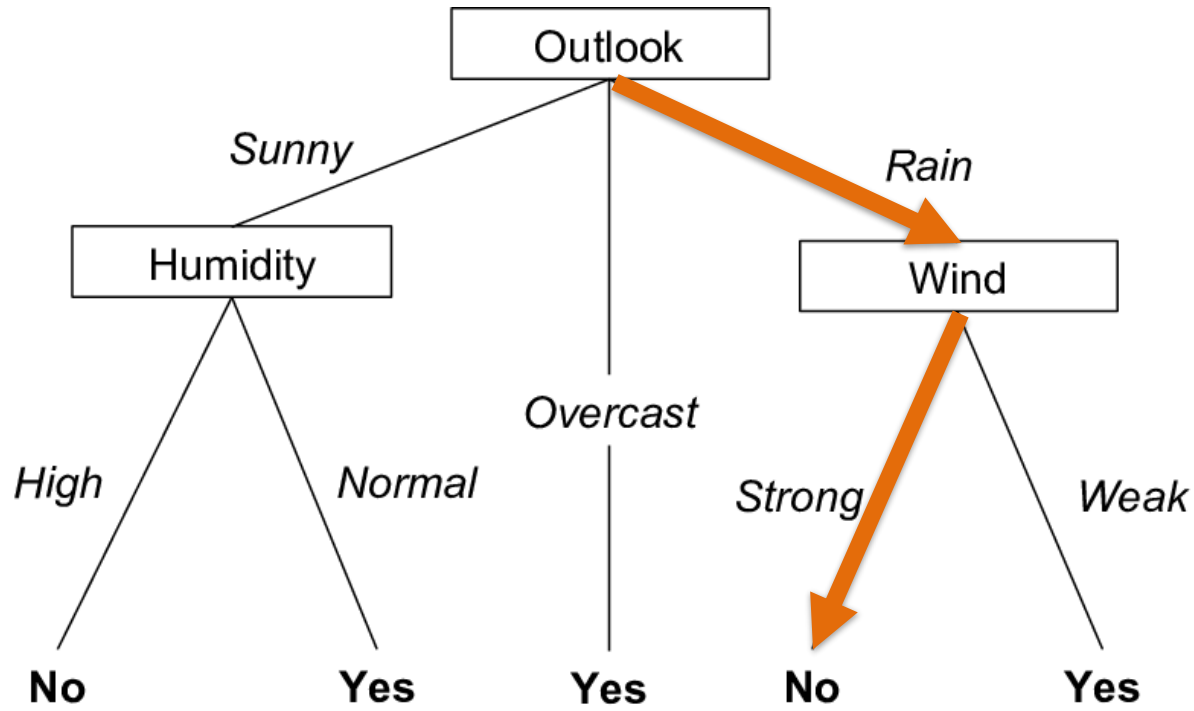
Q4

In the model below, the children of each node divide the data into partitions. Label each node (both internal nodes and leaves) with the counts of “No” and “Yes” labels based on the partition. For example, the counts for the node labeled *Outlook* would be [4,6]. Does this model perfectly classify all examples?



Handout 3

Q5



(test example) $x =$

Outlook	Temp	Humidity	Wind
Rain	Hot	High	Strong

$y_{pred} = \text{No}$

Outline

- Data representation and featurization
- Introduction to modeling
- **Why are models useful?**
- Begin: linear models

Why are models useful?

- Understand/explain/interpret the phenomenon
- Predict outcomes for future examples

What are the most important features?

X

Color	Shape	Size
red	square	big
blue	square	big
red	circle	small
blue	square	small
red	circle	big

Y

Likes toy?
+
+
-
-
+

What are the most important features?

X

Y

Color	Shape	Size
red	square	big
blue	square	big
red	circle	big
blue	square	big
red	circle	big

Likes toy?
+
+
-
-
+

Outline

- Data representation and featurization
- Introduction to modeling
- Why are models useful?
- **Begin: linear models**

Linear Models

* features: \vec{x} (p=1
call it x)

* label: $y \in \mathbb{R}$
output

Goals

① describe linear dependence

② predict output given
new data

model

$$h_{\vec{w}}(x) = \overset{\text{"b"}}{w_0} + \overset{\text{"m"}}{w_1 x} = \hat{y}$$

$$\vec{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

how good is our model? prediction

residuals

$$\begin{array}{ccc} y_i & - & \hat{y}_i \\ \uparrow & & \uparrow \\ \text{truth} & & \text{prediction} \end{array} \quad \left. \vphantom{\begin{array}{ccc} y_i & - & \hat{y}_i \\ \uparrow & & \uparrow \\ \text{truth} & & \text{prediction} \end{array}} \right\} \text{one example}$$

Overall

want to
minimize

$$\sum_{i=1}^n (y_i - \hat{y}_i)^2$$

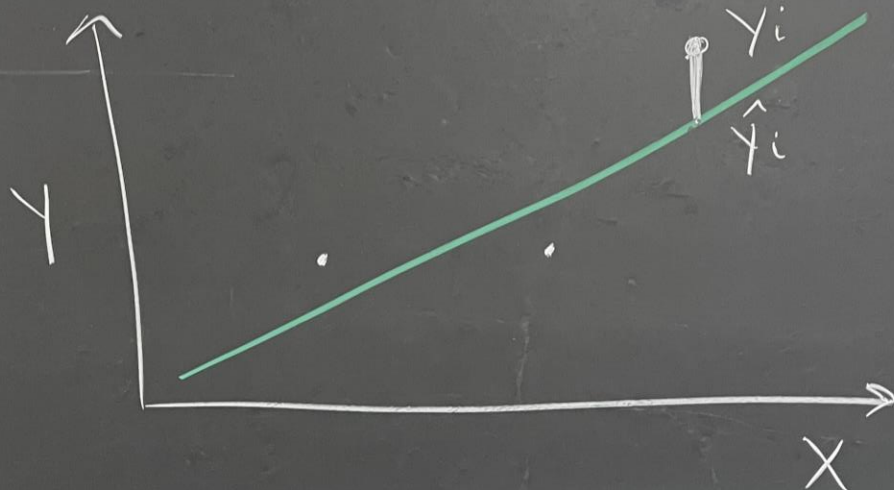
\uparrow
 $(w_0 + w_1 x_i)$

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

prediction

one example

iction



RSS : residual sum of squares

SSE : sum of squared errors

Quote of the week

“The greatest of all mistakes is to do nothing because you think you can only do a little.” — Zig Ziglar