Machine Learning

Traditional Software Development

Traditional Software Development

Estimate **Max HR** from **Age**

Traditional Software Development

Estimate **Max HR** from **Age**

Input:

Output:

Traditional Software Development

Estimate **Max HR** from

Input: **Age**

Output:

Traditional Software Development

Input: **Age**

Relationship: **Max HR** =

Output:

Traditional Software Development

Input: **Age**

Relationship: **Max HR** = 220 - **Age**

Output:

Traditional Software Development

Input: **Age**

Relationship: **Max HR** = 220 - **Age**

Output: estimated **Max HR**

Traditional Software Development

Traditional Software Development

Interpret adult **BMI** **status**

Traditional Software Development

Interpret adult **BMI** **status**

Input:

Output:

Traditional Software Development

Interpret adult **status**

Input: **BMI**

Output:

Traditional Software Development

Interpret adult **status**

Input: **BMI**

Rules:

Output:

Traditional Software Development

Interpret adult

Input: **BMI**

Rules:

if **BMI** < 18.5: **status** = **underweight**

Output:

Traditional Software Development

Interpret adult

Input: **BMI**

Rules:

if **BMI** < 18.5: **status** = **underweight**

else if **BMI** < 25: **status** = **healthy**

Output:

Traditional Software Development

Interpret adult

Input: **BMI**

Rules:

if **BMI** < 18.5: **status** = **underweight**

else if **BMI** < 25: **status** = **healthy**

else: **status** = **overweight**

Output:

Traditional Software Development

Interpret adult

Input: **BMI**

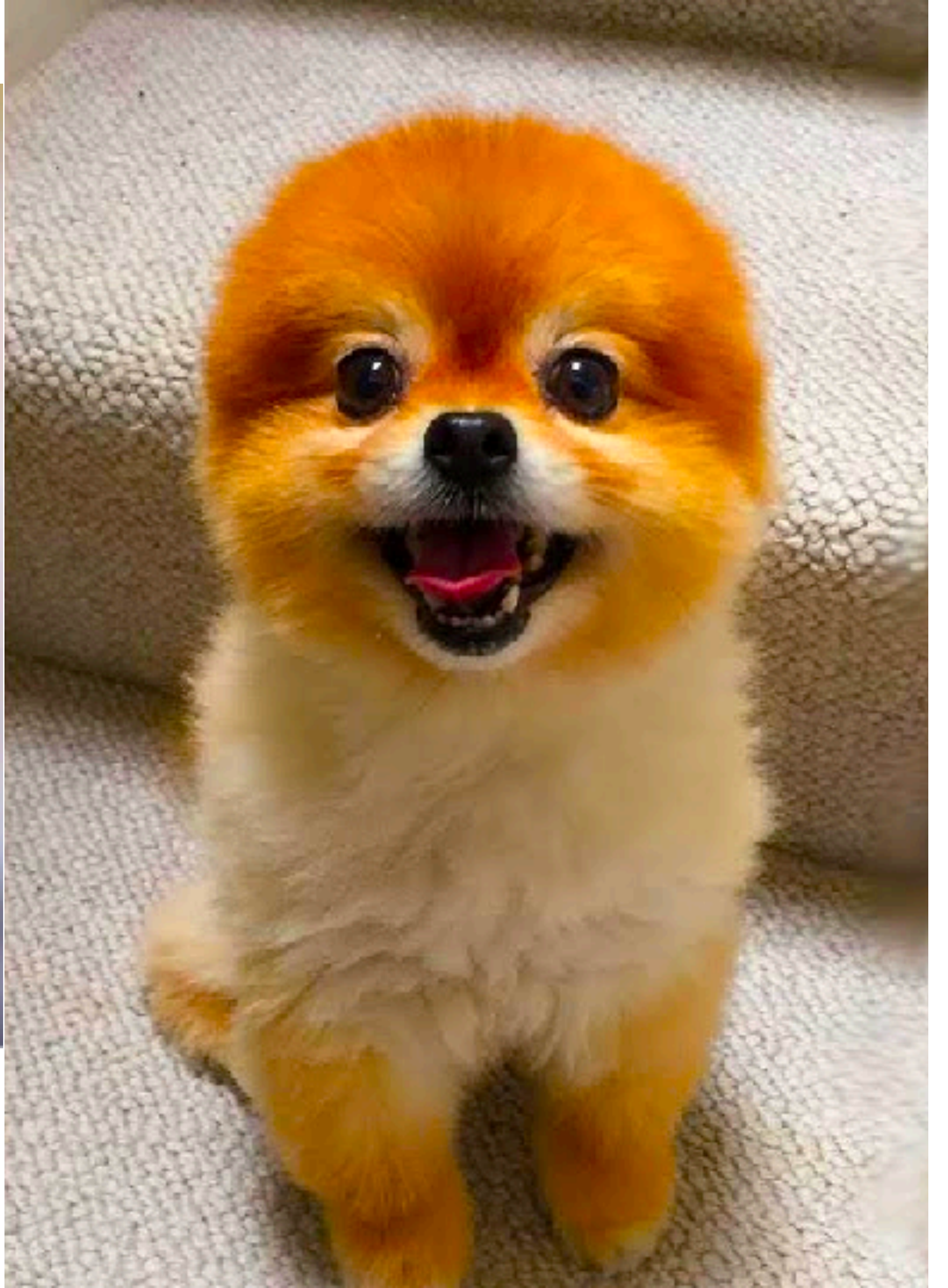
Rules:

```
     if BMI < 18.5:  status = underweight
else if BMI < 25:    status = healthy
else:                status = overweight
```

Output: **status**

Traditional Software Development

Traditional Software Development





Input: [ , ]

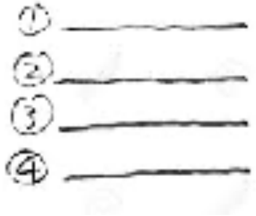
Rules:

Output:

Input: [ , ]

Rules:

if:



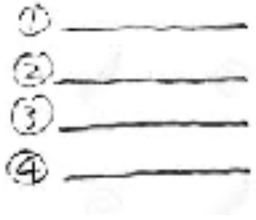
→ “cat”

Output:

Input: [ , ]

Rules:

if:



else:

"cat"

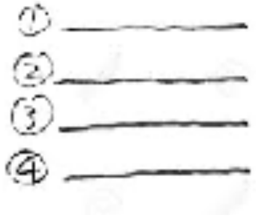
"dog"

Output:

Input: [ , ]

Rules:

if:



“cat”

else:

“dog”

Output: [**“cat”, “dog”**]

Machine Learning

Input	144	181	200	317	800
Output					

Machine Learning

Input	144	181	200	317	800
Output	256	219	200	83	-400

Output = 400 - **Input**

Input	144	181	200	317	800
Output	256	219	200	83	-400

Machine Learning

Input	144	181	200	317	800
Output	256	219	200	83	-400

Machine Learning

Input	144	181	200	317	800

Relationship:

Output	256	219	200	83	-400

Machine Learning

Input	144	181	200	317	800

Relationship: 

Output	256	219	200	83	-400

=

Common ML Algorithms

Linear Regression

Logistic Regression

Support Vector Machine

Decision Tree

K-Nearest Neighbor

Machine Learning

Input	144	181	200	317	800

Relationship: 

Output	256	219	200	83	-400

Machine Learning

Input	144	181	200	317	800

Relationship: **400 - input** ← **Model**

Output	256	219	200	83	-400

Machine Learning

ML Model

400 - input

Machine Learning

ML Model

New input: **317** → **400 - input**

Machine Learning

ML Model

New input: **317** → **400 - input** → output: **83**

$$\hat{f}(X)$$

The Prediction

Models the relationship between input and output

$$\underset{}{\overset{\text{output}}{\hat{y}}} = \hat{f}(\overset{\text{input}}{X})$$

The Prediction

Models the relationship between X and y

output input

$$\hat{y} = \hat{f}(X)$$

Input: [ , ]

Output:

Machine Learning

Input:[,  ,  , ,  , ]

Output:

Machine Learning

Input:[ ,  ,  ,  ,  , ]

Output: ["cat", "dog", "cat", "dog", "dog", "cat"]

labels

Machine Learning

Input:[, , , , , ]

Relationship: 

Output: ["cat", "dog", "cat", "dog", "dog", "cat"]

Machine Learning

Input:[, , , , , ]

Relationship: 

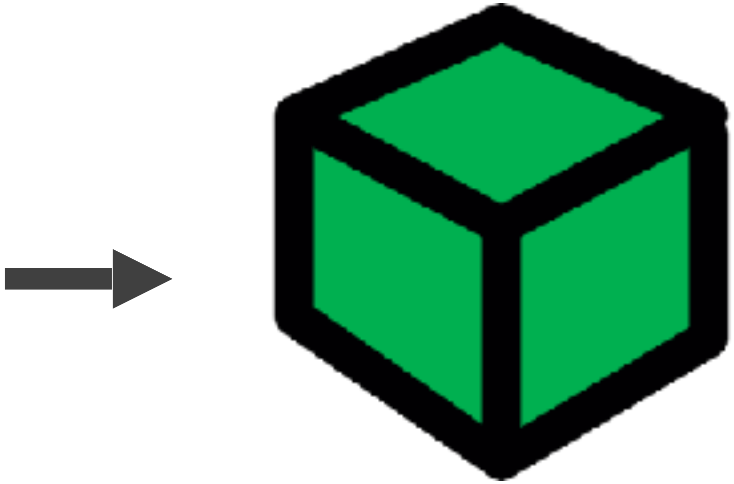
MODEL

Output: ["cat", "dog", "cat", "dog", "dog", "cat"]

Machine Learning

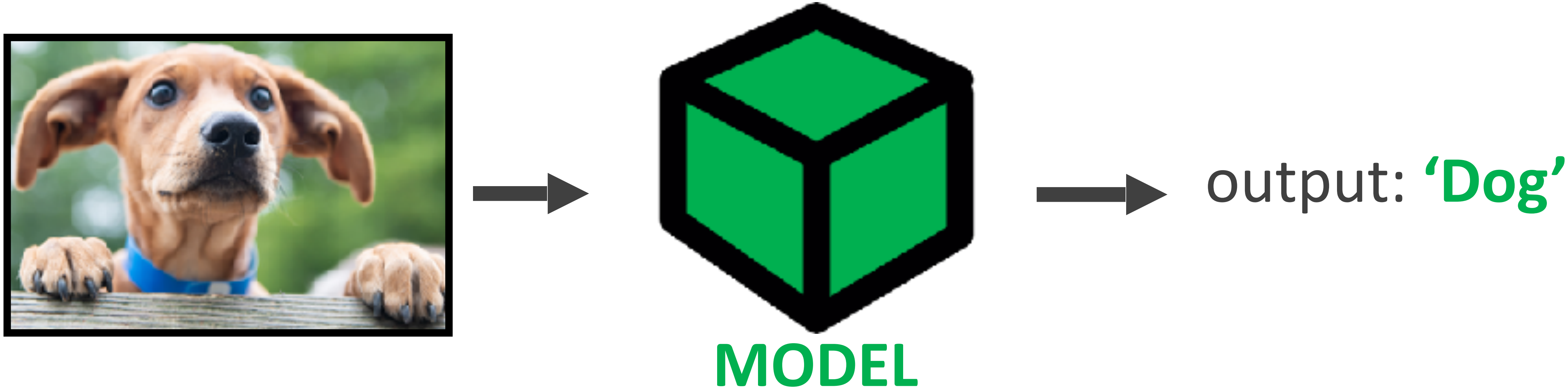
MODEL

Machine Learning



MODEL

Machine Learning



Machine Learning

MODEL

output: 1

Machine Learning

Instead of programming a computer, you give a computer **examples** and it **learns** what you want.

Pareidolia

Types of Machine Learning

Supervised **Unsupervised**

Music

Song	Artist	Genre	Liked
Breathing Light	Frameworks	Alternative Rock	Yes
Superior	Silver Maple	Pop	No
Icicle	AK	Pop	No
Jazzin	Flap Jack	R&B	Yes
The Way You Do	Schlomo	R&B	Yes
Mirror Maru	Cashmere	Rock	Yes
Never Too Far	Sorrow	Pop	No

Music

Features (X)

Song	Artist	Genre	Liked
Breathing Light	Frameworks	Alternative Rock	Yes
Superior	Silver Maple	Pop	No
Icicle	AK	Pop	No
Jazzin	Flap Jack	R&B	Yes
The Way You Do	Schlomo	R&B	Yes
Mirror Maru	Cashmere	Rock	Yes
Never Too Far	Sorrow	Pop	No

Music

Song	Artist	Genre	Liked
Breathing Light	Frameworks	Alternative Rock	Yes
Superior	Silver Maple	Pop	No
Icicle	AK	Pop	No
Jazzin	Flap Jack	R&B	Yes
The Way You Do	Schlomo	R&B	Yes
Mirror Maru	Cashmere	Rock	Yes
Never Too Far	Sorrow	Pop	No

Target (y)

Music

Song	Artist	Genre	Liked
Breathing Light	Frameworks	Alternative Rock	Yes
Superior	Silver Maple	Pop	No
Icicle	AK	Pop	No
Jazzin	Flap Jack	R&B	Yes
The Way You Do	Schlomo	R&B	Yes
Mirror Maru	Cashmere	Rock	Yes
Never Too Far	Sorrow	Pop	No

Labels

Supervised

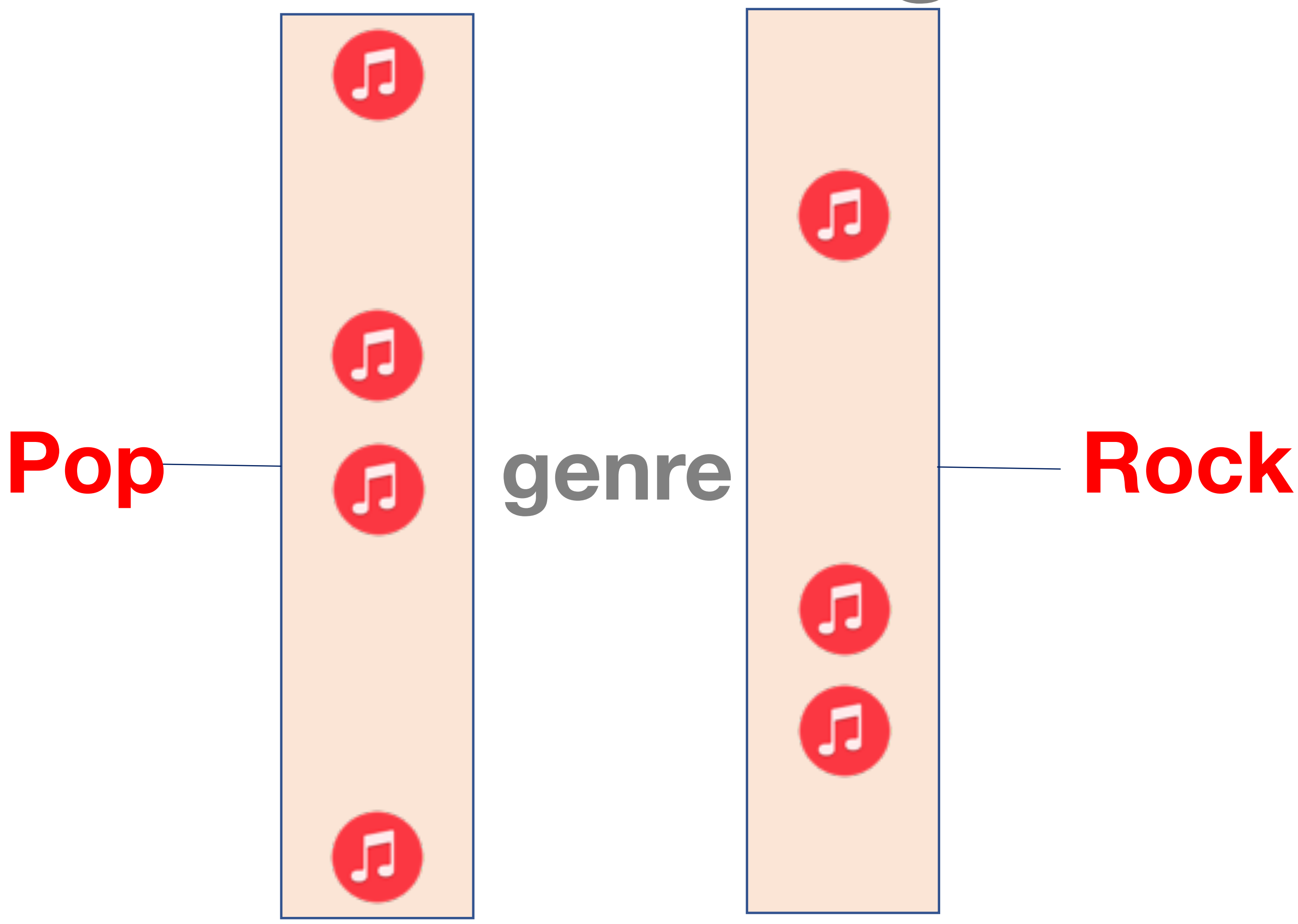
Features	Label
	Yes
	No
	No
	Yes
	Yes
	Yes
	No

Unsupervised

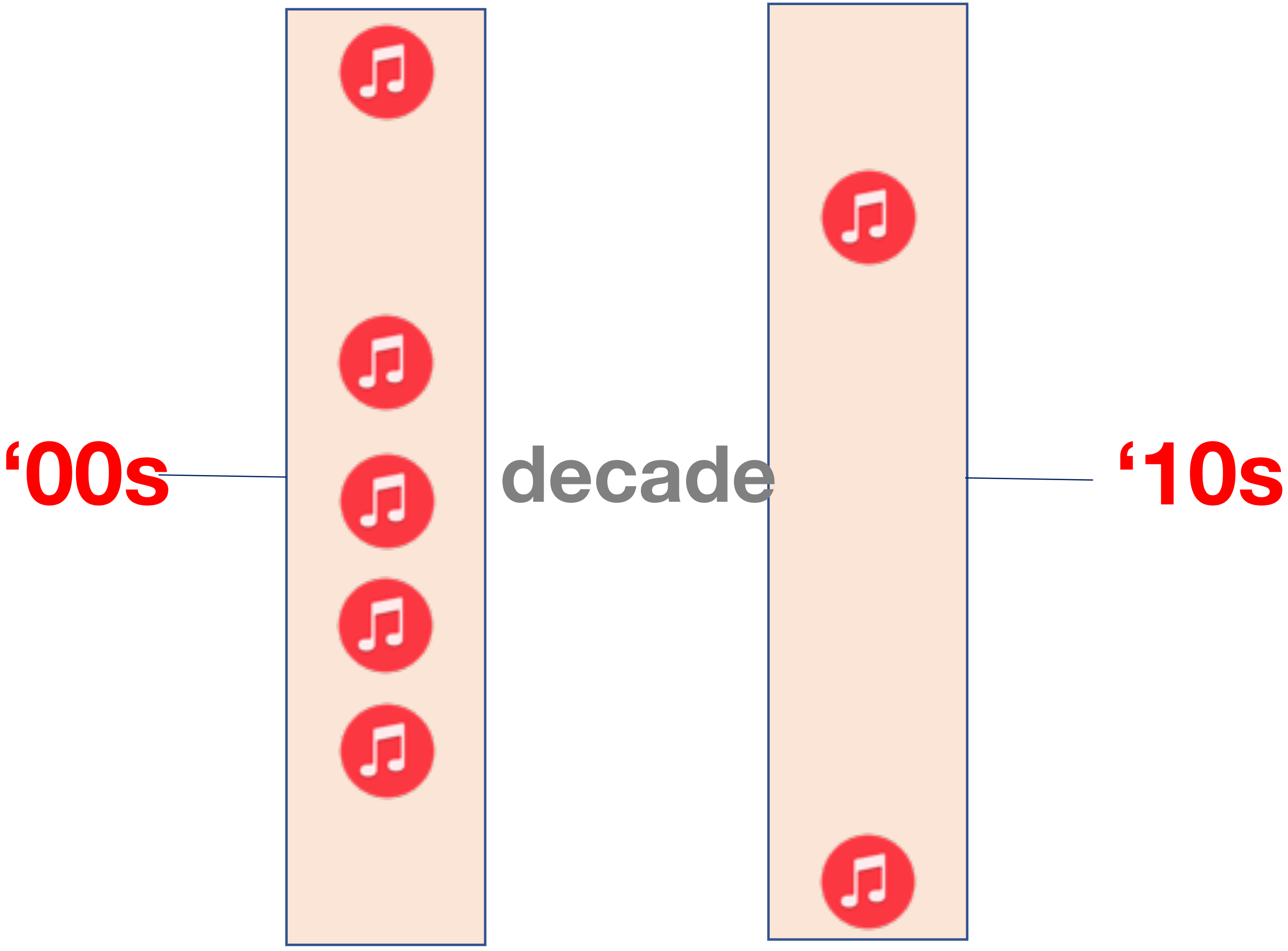
Features	Label

Unsupervised

Clustering



Clustering



Supervised

Regression

Classification

Unsupervised

Clustering

Data

The best data has 3 qualities:

- Clean
- Coverage
- Complete

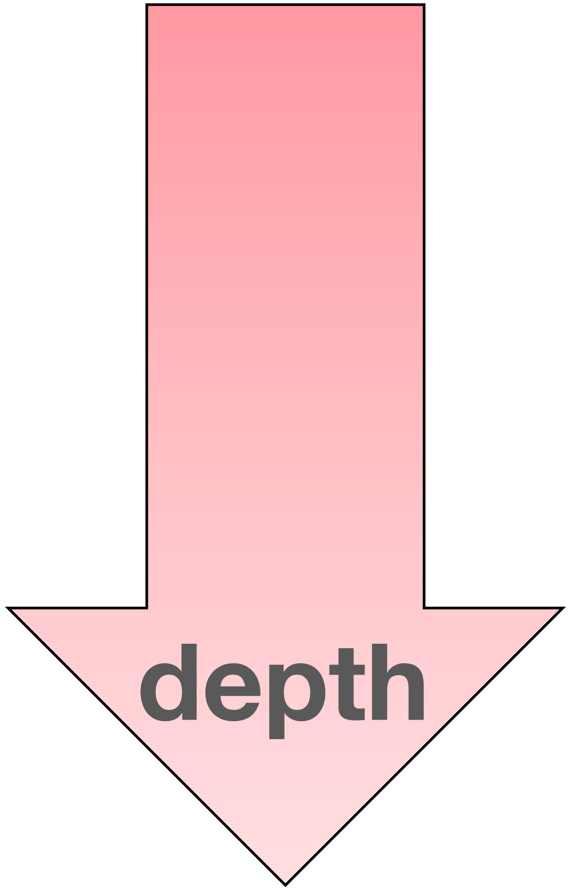
The best data has 3 qualities:

Feature 1	Feature 2	Feature 3	Feature 4
Male	200	1	Yes
Female	316	3	No
F	190	1	No
Male	244		Yes
Male	128	2	Yes
Male		3	Yes
Female	302	2	No

Clean

Feature 1	Feature 2	Feature 3	Feature 4
Male	200	1	Yes
Female	316	3	No
F	190	1	No
Male	244	13	Yes
Male	128	2	Yes
Male		3	Yes
Female	302	2	No

Coverage



Feature 1	Feature 2	Feature 3	Feature 4
Male	200	1	Yes
Female	316	3	No
F	190	1	No
Male	244		Yes
Male	128	2	Yes
Male		3	Yes
Female	302	2	No

Complete

breadth

Feature 1	Feature 2	Feature 3	Feature 4
Male	200	1	Yes
Female	316	3	No
F	190	1	No
Male	244		Yes
Male	128	2	Yes
Male		3	Yes
Female	302	2	No


If ML is a rocket engine,
data is the fuel

Learning the Relationship

Machine Learning

Input	**1**	**2**								
Output	2	4								

$y = 2x$

Input	1	2								
Output	2	4								

Machine Learning

Input	**1**		**3**		**5**		**7**		**9**	
Output	2		10		26		50		82	

$$y = x^2 + 1$$

Input	**1**		**3**		**5**		**7**		**9**	
Output	2		10		26		50		82	

Machine Learning

Input		2		4		6		8		10
Output		4		16		36		64		100

$$y = x^2$$

Input		**2**		**4**		**6**		**8**		**10**
Output		4		16		36		64		100

Machine Learning

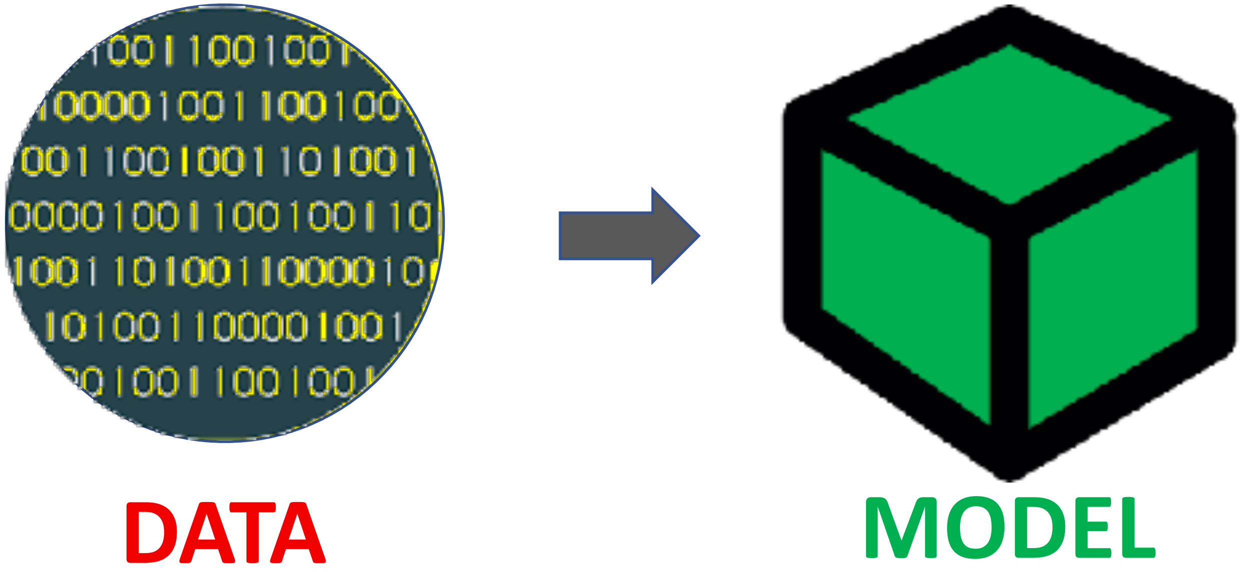
Input	1	2	3	4	5	6	7	8	9	10
Output	2	4	10	16	26	36	50	64	82	100

$$y = x^2 + (x\%2)$$

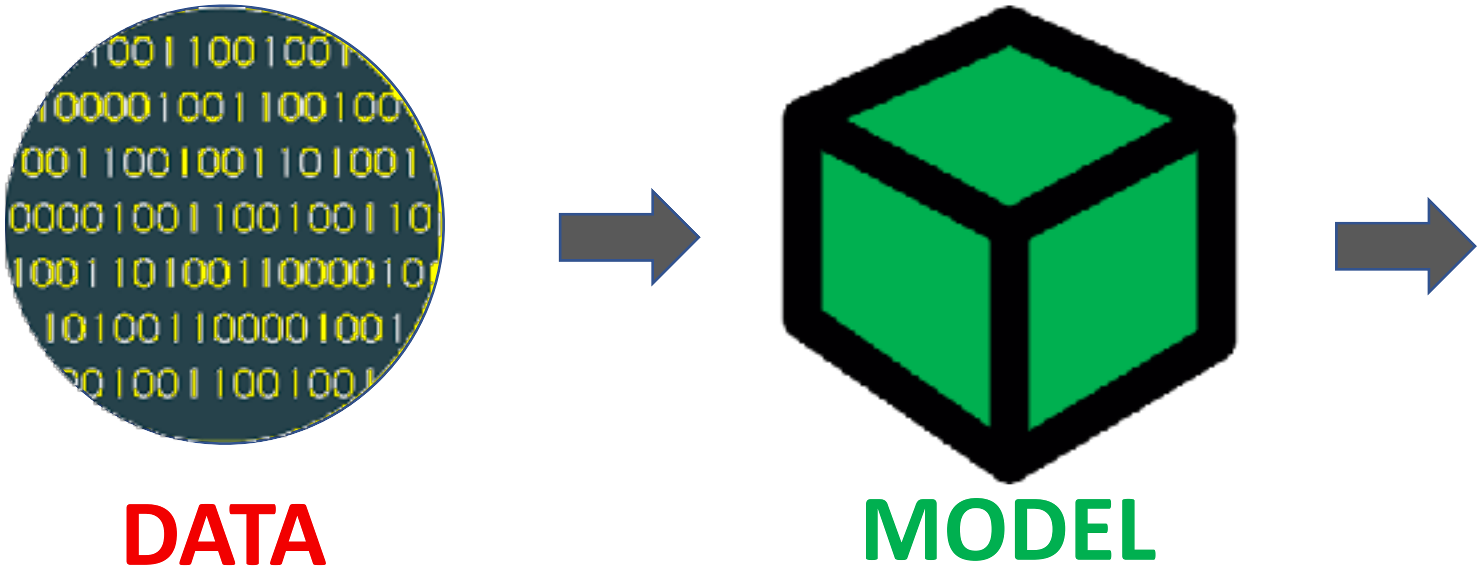
Input	1	2	3	4	5	6	7	8	9	10
Output	2	4	10	16	26	36	50	64	82	100

Model Training

Model Training

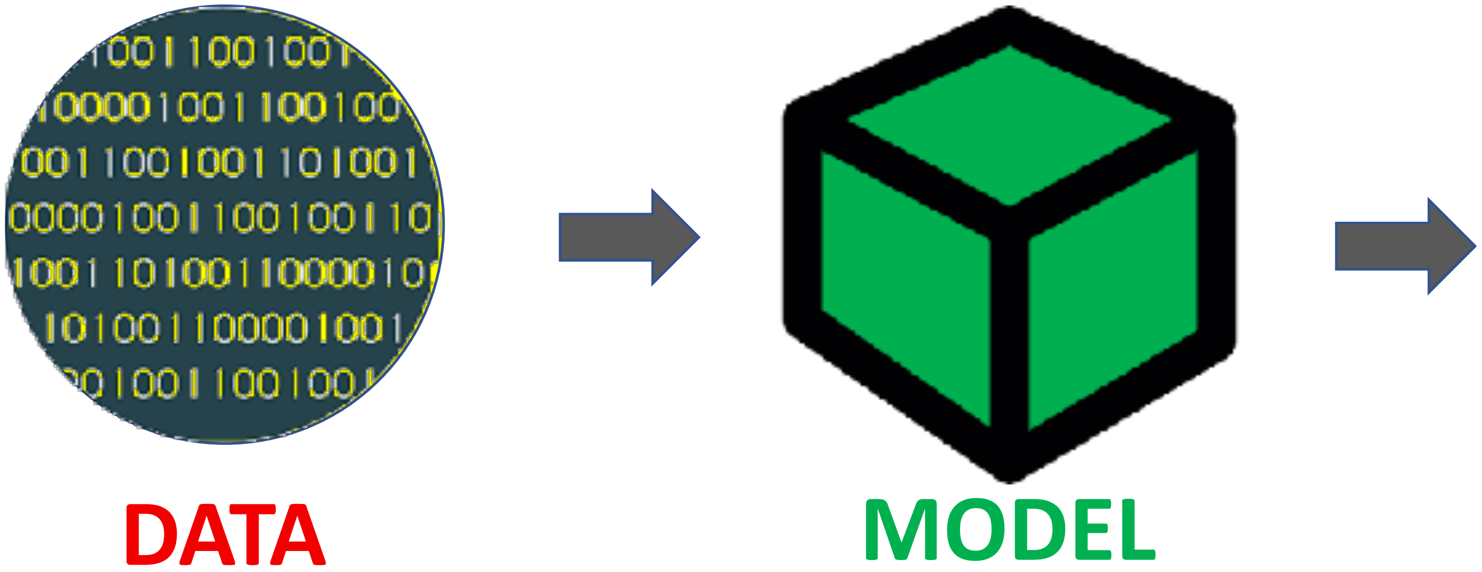


Model Training



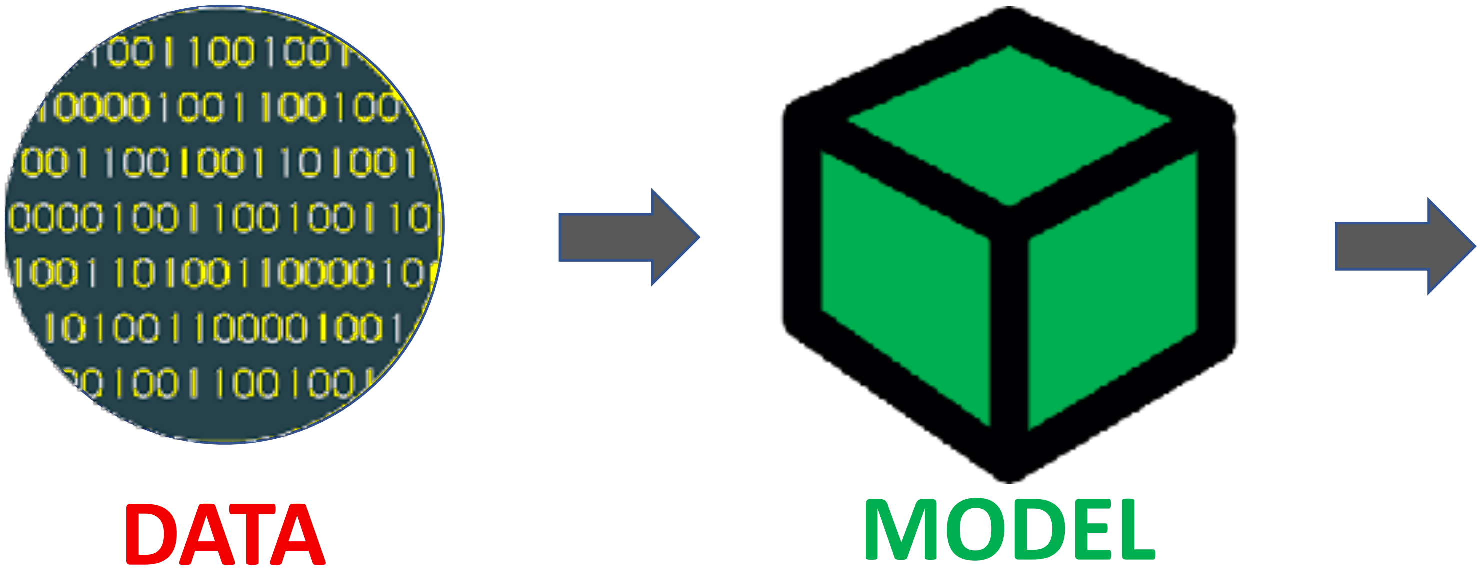
Prediction
0
1
0
0
1
0

Model Training



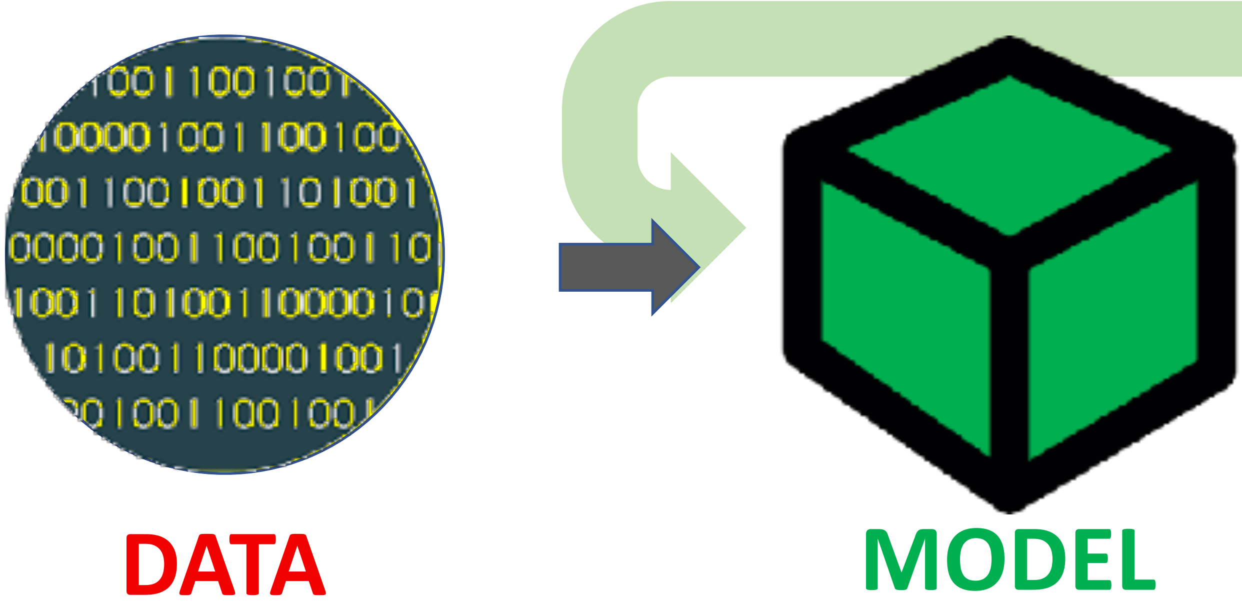
Prediction	Label
0	1
1	1
0	0
0	1
1	0
0	0

Model Training



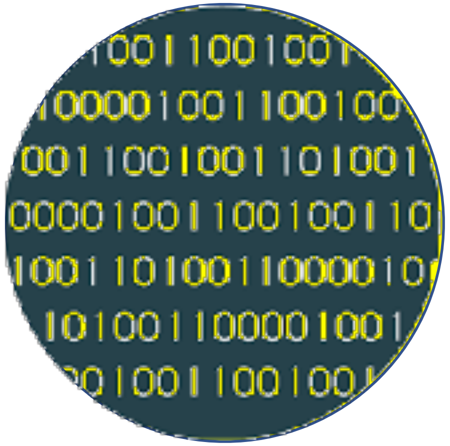
Prediction	Label
0	1
1	1
0	0
0	1
1	0
0	0

Model Training



Prediction	Label
0	1
1	1
0	0
0	1
1	0
0	0

Model Training



DATA

MODEL

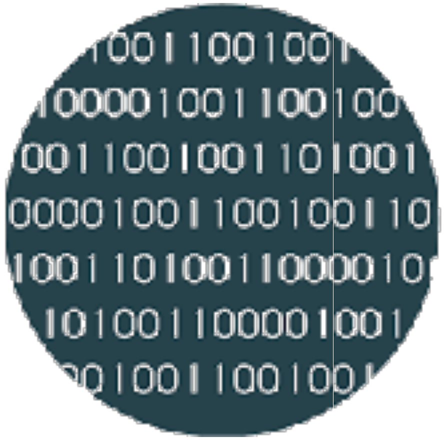
Prediction	Label
1	1
1	1
0	0
1	1
0	0
0	0

Model Training

TRAINED MODEL

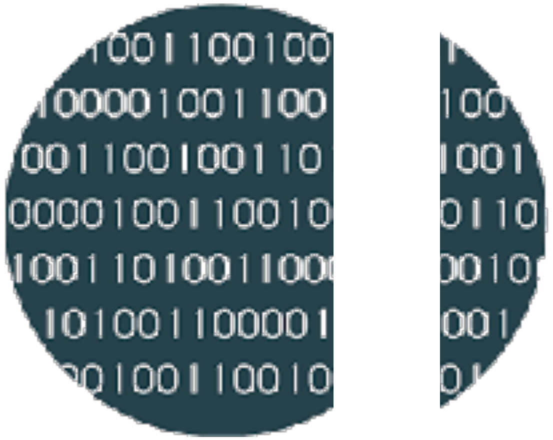
Model Training

Evaluate the Model



Evaluate the Model

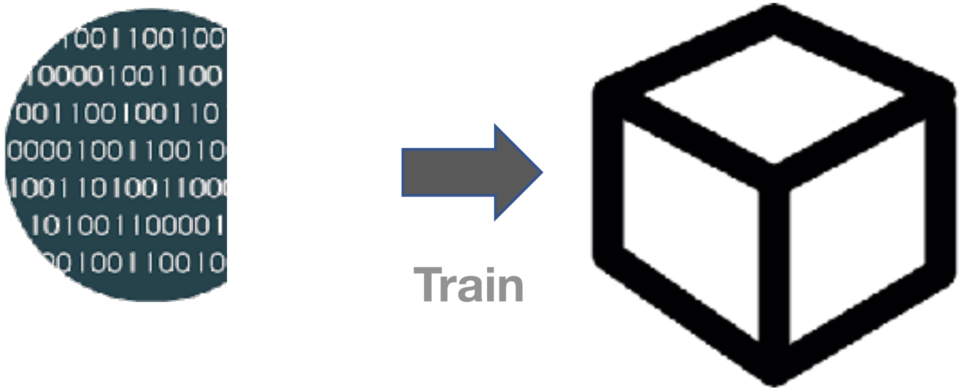
Training Data

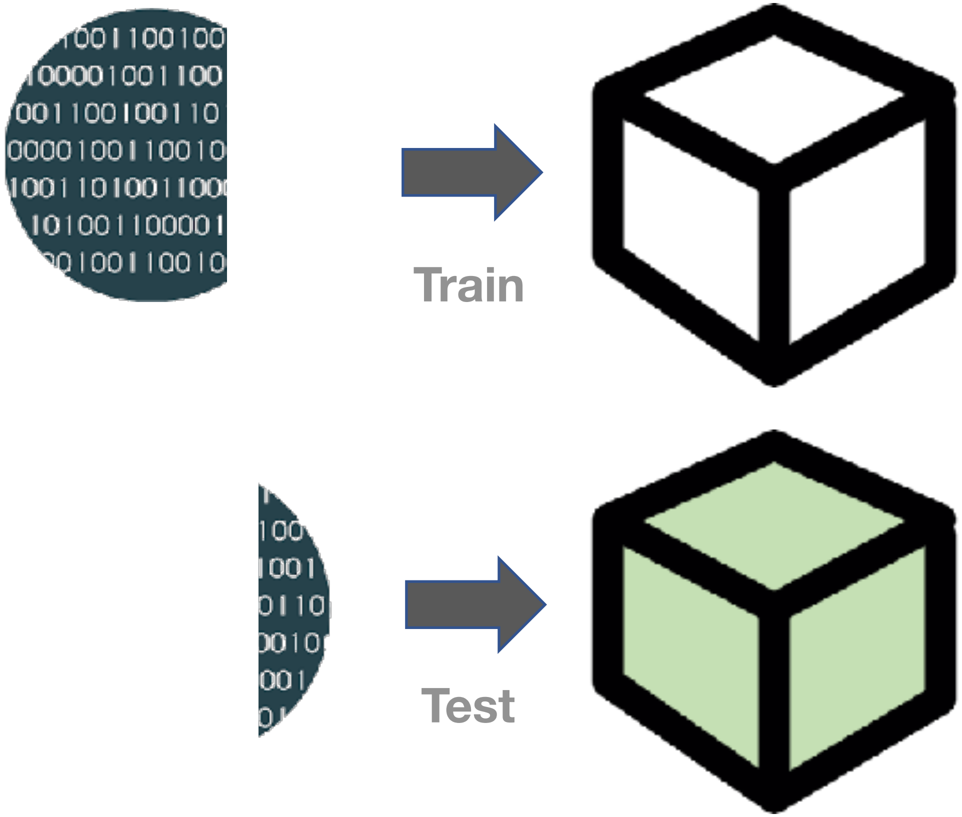


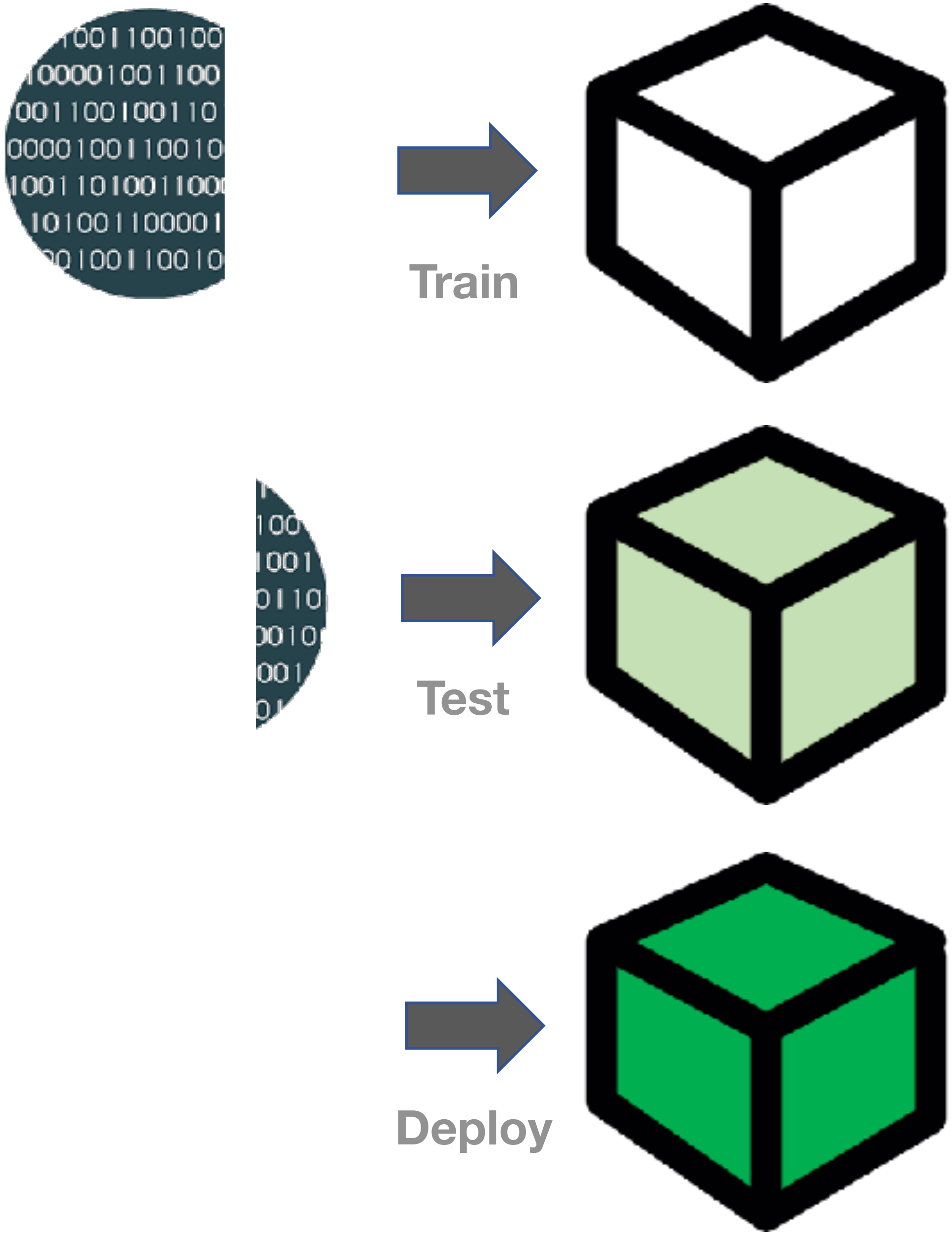
Test Data

Evaluate the Model

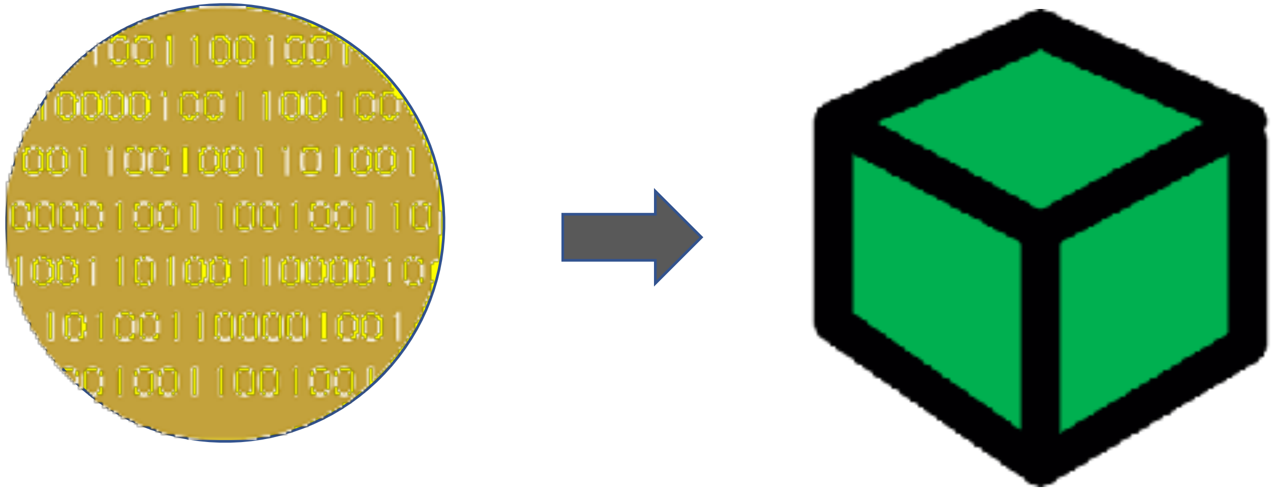
Training Data


Train


Train
Test

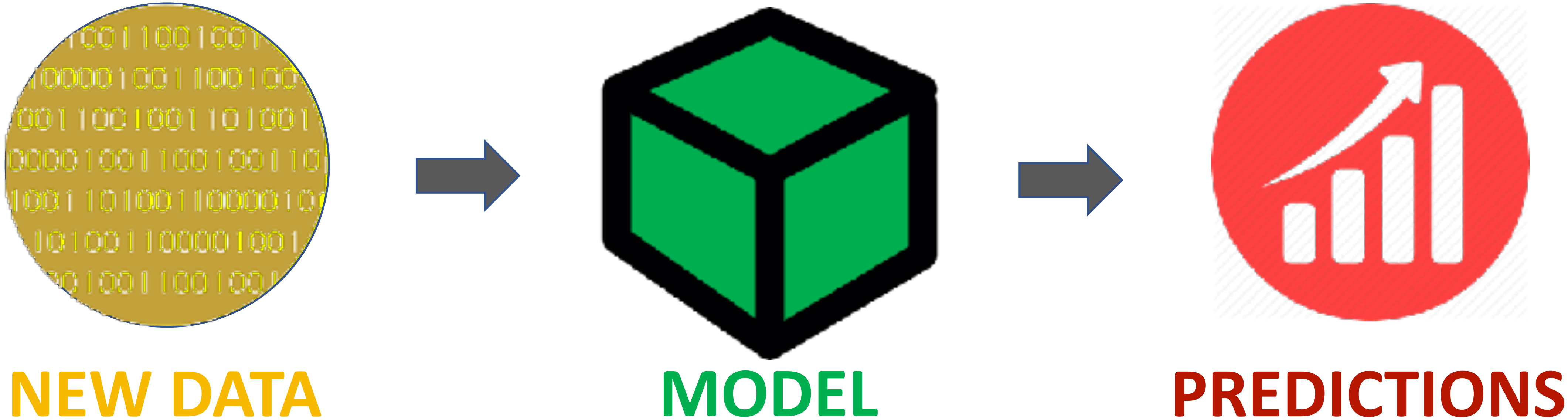

Train
Test
Deploy


DATA
ALGORITHM
MODEL



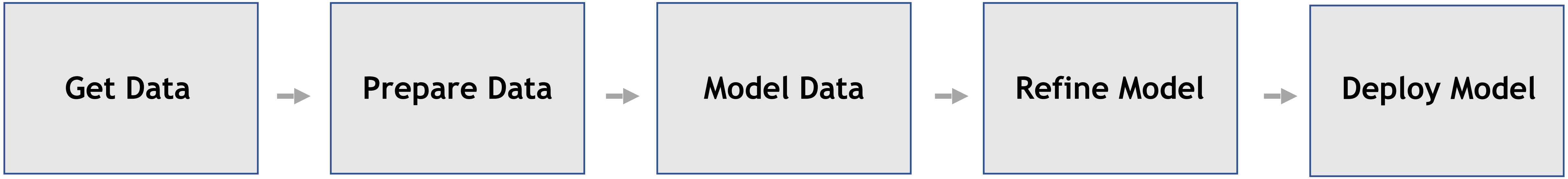
NEW DATA

MODEL



NEW DATA
MODEL
PREDICTIONS

ML Process



ML Process

Get Data → Prepare Data → Model Data → Refine Model → Deploy Model

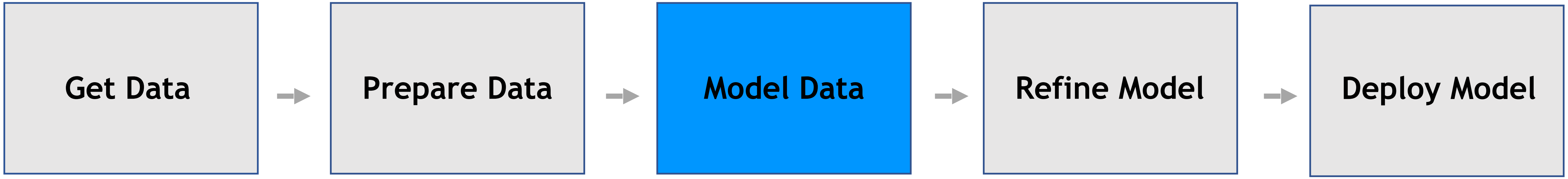
What data should you use?
Is it labeled?

ML Process

Get Data → **Prepare Data** → Model Data → Refine Model → Deploy Model

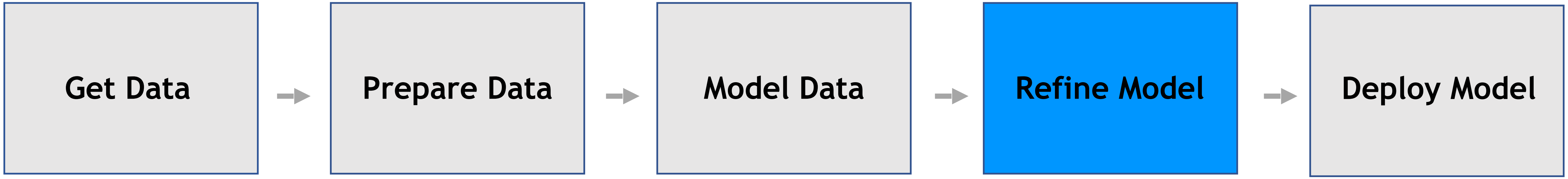
Is your data **complete**, **clean**, does it have **coverage**?

ML Process



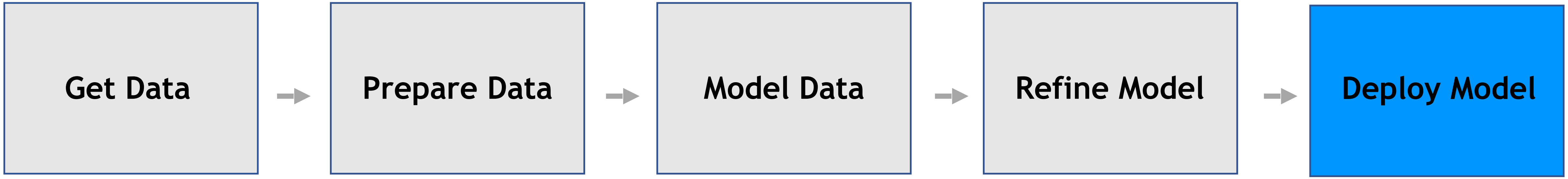
Which algorithms should you use?

ML Process



What level of performance is sufficient?

ML Process



Make predictions.

How do machines learn?

How do machines learn?

By example.