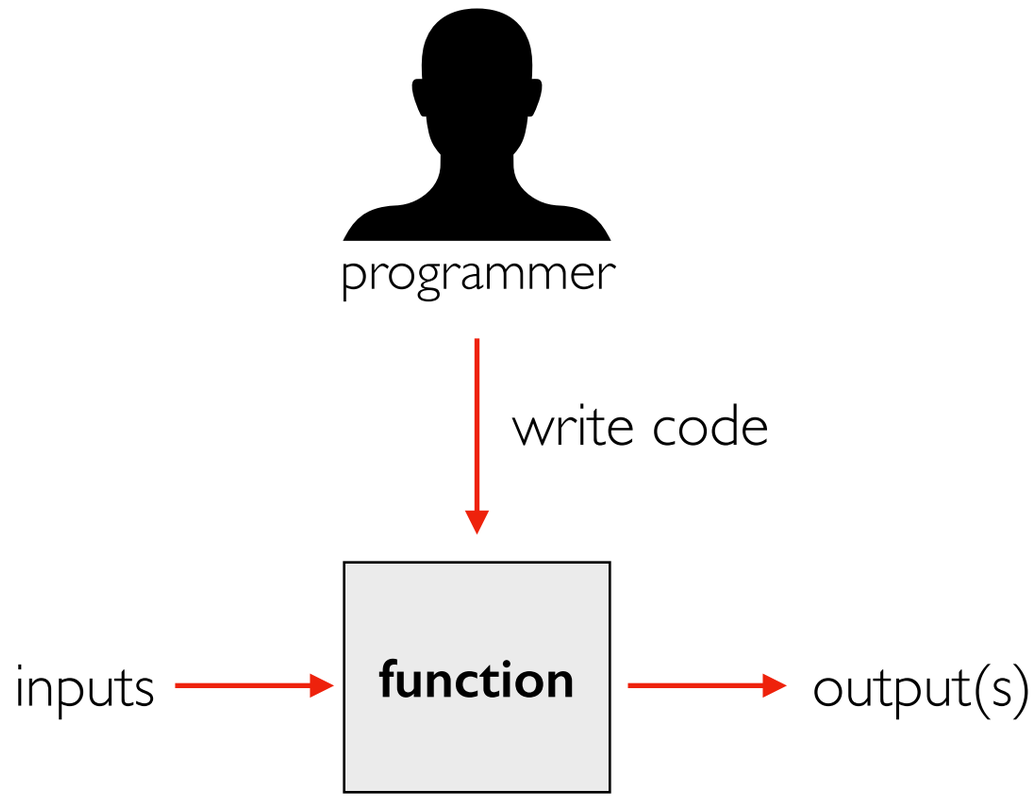
[320] Pre-Machine Learning: Intro

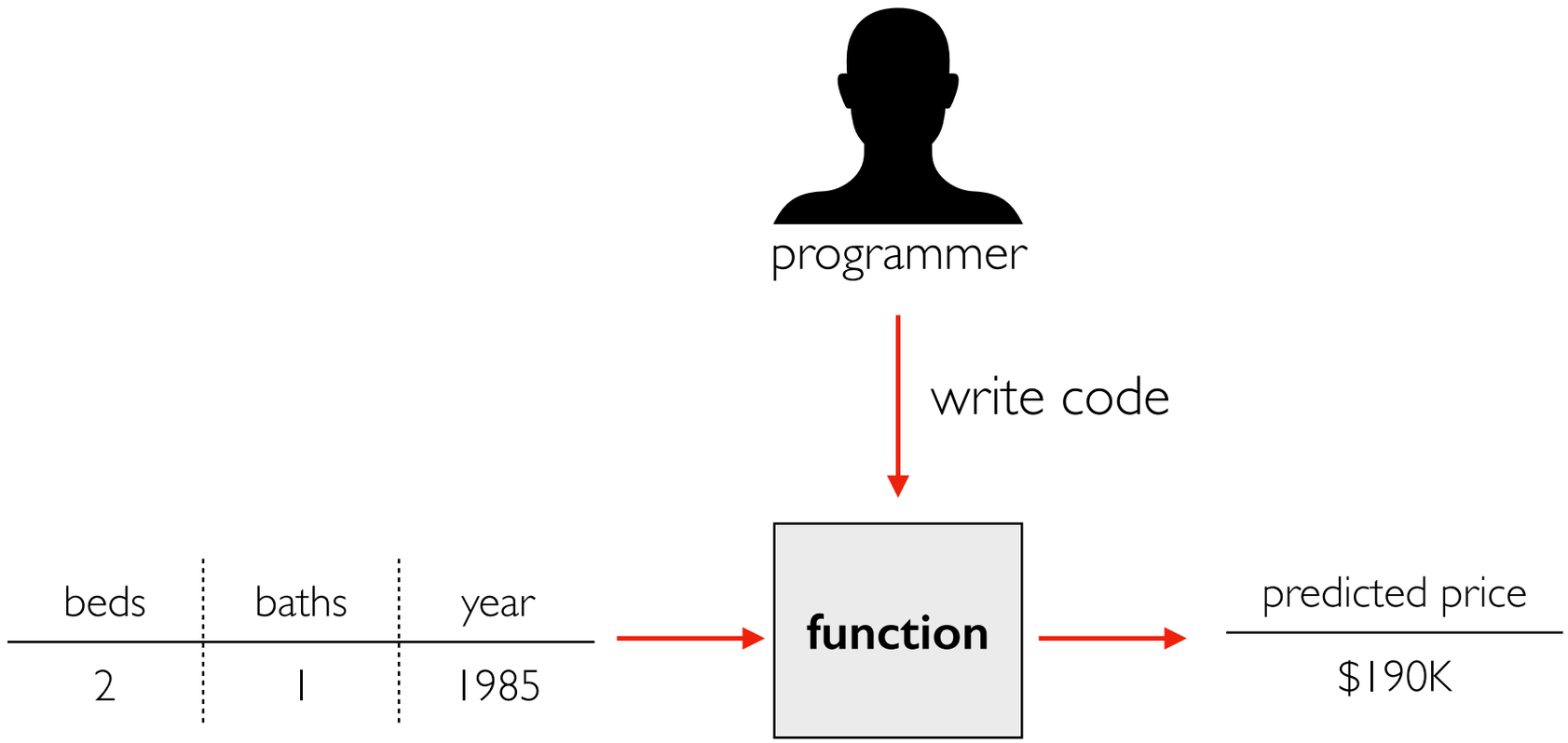
Tyler Caraza-Harter

Functions/Models

How do we make functions?

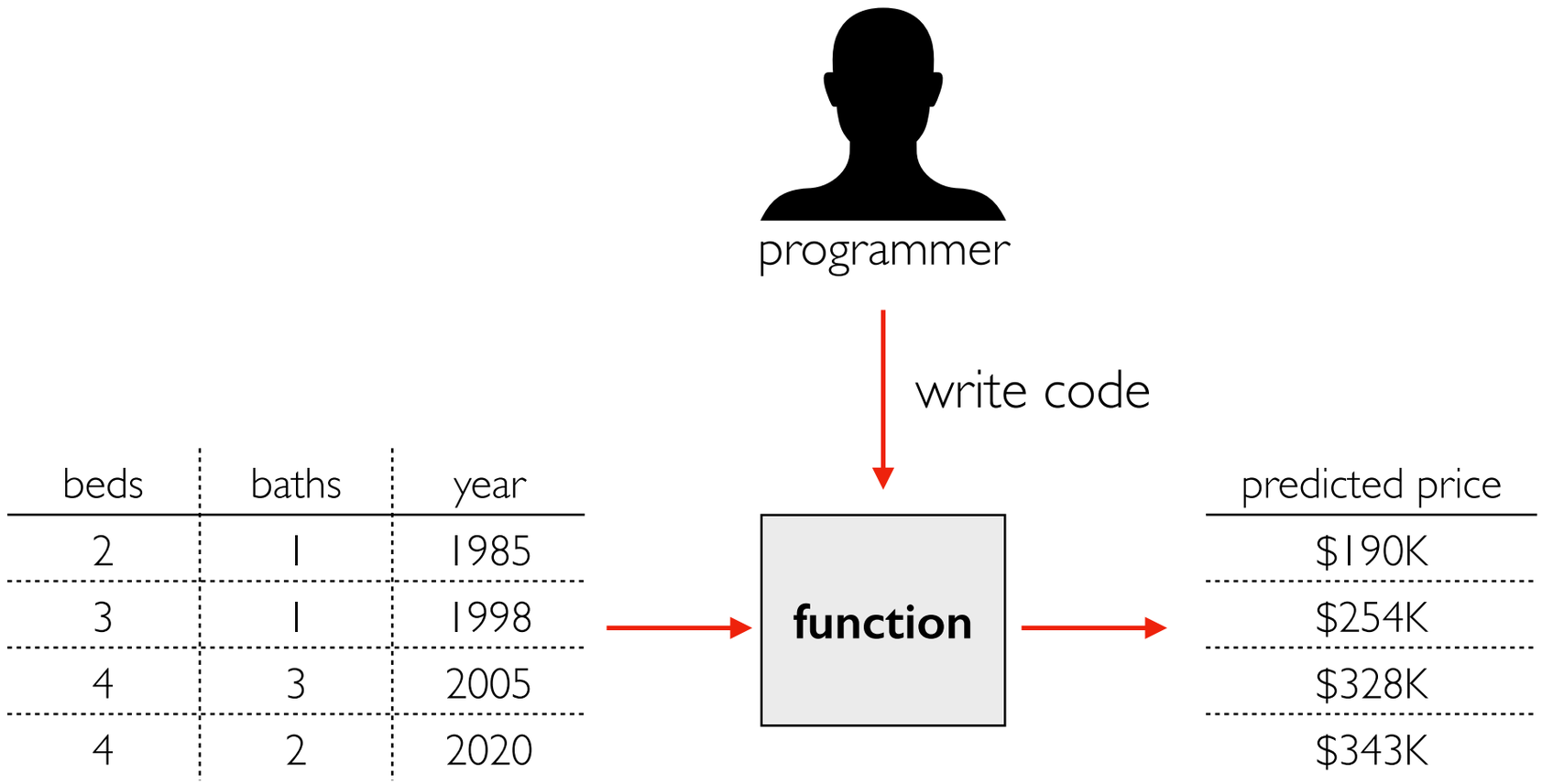


How do we make functions?



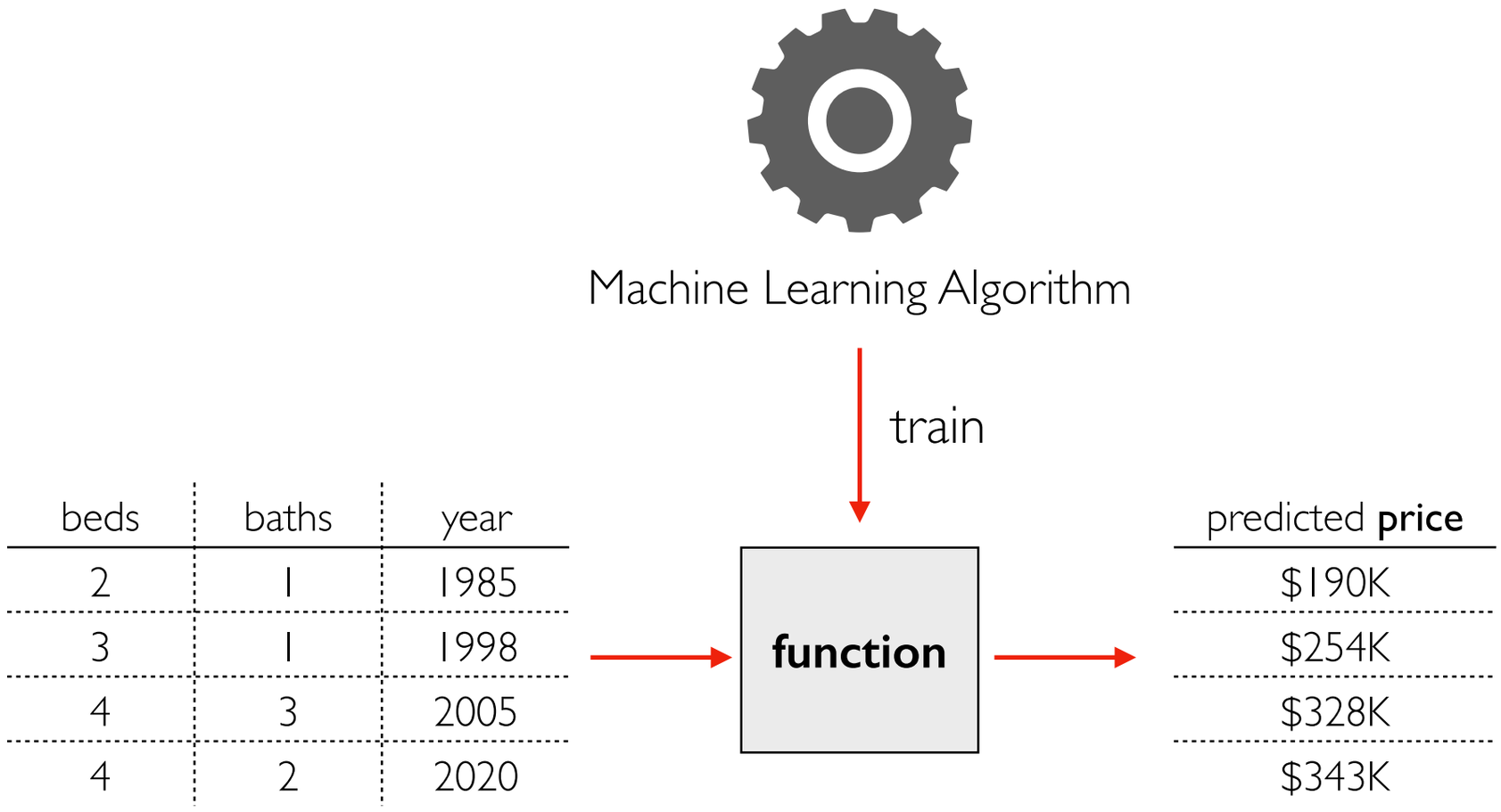
many functions are **models** that can be used to predict

How do we make functions?



many functions are **models** that can be used to predict

How do we make functions?

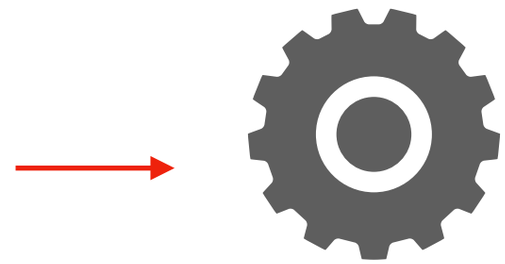


many functions are **models** that can be used to predict

How do we make functions?

training data

beds	baths	year	**price**
1	1	1980	$140K
3	1	1990	$240K
3	4	2004	$295K
4	3	2018	$350K



Machine Learning Algorithm

train

live data

beds	baths	year
2	1	1985
3	1	1998
4	3	2005
4	2	2020

function

predicted **price**
$190K
$254K
$328K
$343K

many functions are **models** that can be used to predict

Kinds of Machine Learning

Main Categories of Machine Learning

learning from data

1

Supervised Machine Learning

*data is **labeled**, we know what we want to predict*

2

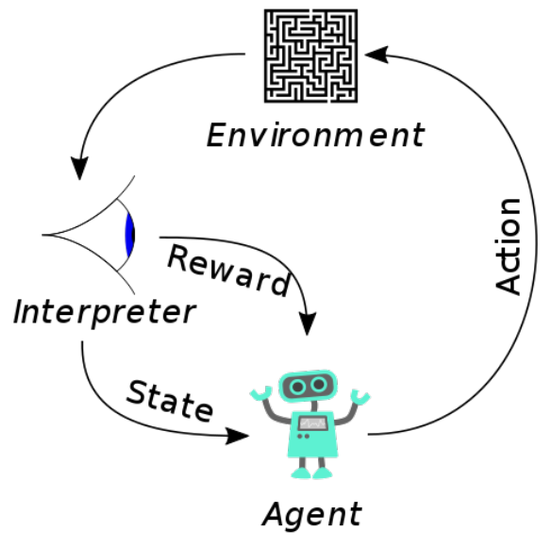
Unsupervised Machine Learning

*data is **unlabeled**, we're just looking for patterns*

3

Reinforcement Learning

not covered in CS 320



https://en.wikipedia.org/wiki/Reinforcement_learning

Main Categories of Machine Learning

learning from data

1

Supervised Machine Learning

*data is **labeled**, we know what we want to predict*

2

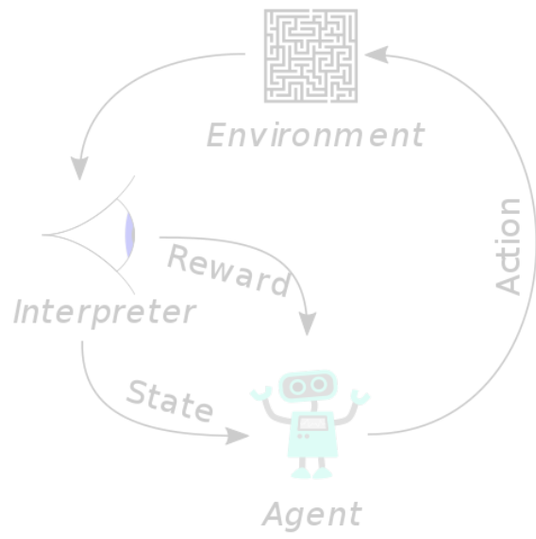
Unsupervised Machine Learning

*data is **unlabeled**, we're just looking for patterns*

3

Reinforcement Learning

not covered in CS 320



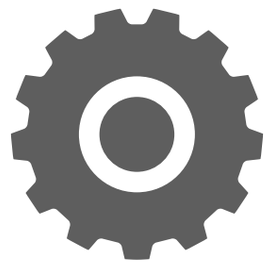
https://en.wikipedia.org/wiki/Reinforcement_learning

Supervised Learning

supervised because data is labeled,
we know what we want to predict

training data

beds	baths	year	**price**
1	1	1980	$140K
3	1	1990	$240K
3	4	2004	$295K
4	3	2018	$350K



Machine Learning Algorithm

train

live data

beds	baths	year
2	1	1985
3	1	1998
4	3	2005
4	2	2020

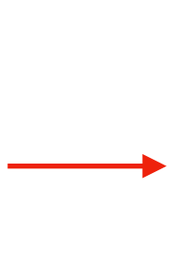
function

predicted **price**
$190K
$254K
$328K
$343K

Supervised Learning: Regression

training data

beds	baths	year	**price**
1	1	1980	$140K
3	1	1990	$240K
3	4	2004	$295K
4	3	2018	$350K



Machine Learning Algorithm

train

live data

beds	baths	year
2	1	1985
3	1	1998
4	3	2005
4	2	2020

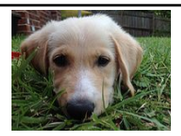
function

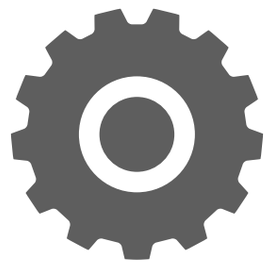
predicted **price**
$190K
$254K
$328K
$343K

our function in a **regressor** because
it's outputting continuous data

Supervised Learning: Classification

training data

photo	**label**
	dog
	dog
	cat



Machine Learning Algorithm

train

live data

photo


function

classification
dog
dog
cat

our function in a **classifier** because
it's outputting distcrete data

https://en.wikipedia.org/wiki/Dog https://en.wikipedia.org/wiki/Cat

Main Categories of Machine Learning

learning from data

1

Supervised Machine Learning

*data is **labeled**, we know what we want to predict*

2

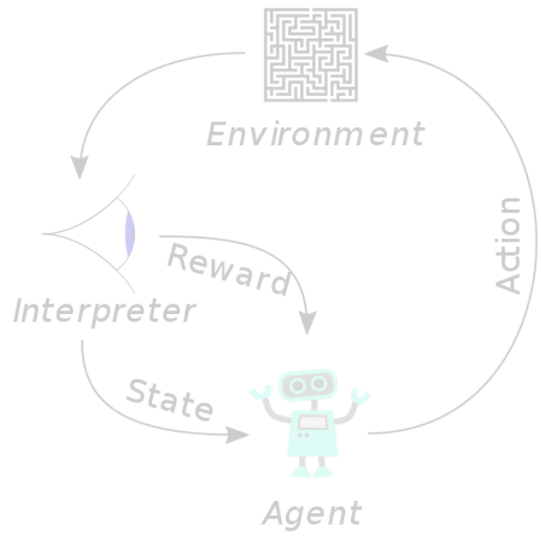
Unsupervised Machine Learning

*data is **unlabeled**, we're just looking for patterns*

3

Reinforcement Learning

not covered in CS 320



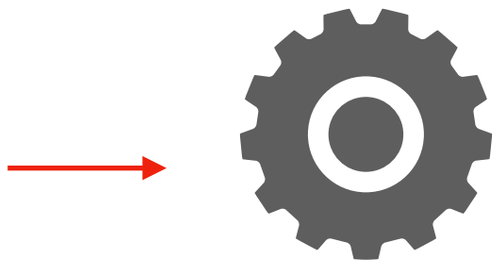
https://en.wikipedia.org/wiki/Reinforcement_learning

Unsupervised Learning

unsupervised because each column is just data, with no special column we're trying to predict

training data

beds	baths	year
1	1	1980
3	1	1990
3	4	2004
4	3	2018



Machine Learning Algorithm

train

live data

beds	baths	year
2	1	1985
3	1	1998
4	3	2005
4	2	2020

function

cluster
cluster 1
cluster 1
cluster 2
cluster 2

unsupervised clustering algorithms try to identify groups of similar data. The algorithm decides the groups.

Sometimes (but often not) they'll correspond to things we describe. E.g., cluster 1: old houses with few bathrooms; cluster 2: new houses with many bathrooms

Foundations

Important Packages

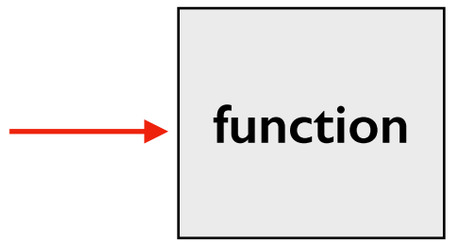
We'll be learning the following to do ML and related calculations efficiently:

1. numpy
2. pytorch
3. scikit-learn

```
pip3 install numpy scikit-learn
pip3 install torch==1.4.0+cpu torchvision==0.5.0+cpu -f https://download.pytorch.org/whl/torch_stable.html
```

Linear Algebra

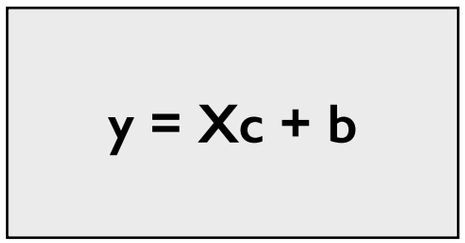
beds	baths	year
2	1	1985
3	1	1998
4	3	2005
4	2	2020



predicted **price**
$190K
$254K
$328K
$343K

with matrices...

$$\begin{bmatrix} 2 & 1 & 1985 \\ 3 & 1 & 1998 \\ 4 & 3 & 2005 \\ 4 & 2 & 2020 \end{bmatrix}$$



$$\begin{bmatrix} 190000 \\ 254000 \\ 328000 \\ 343000 \end{bmatrix}$$

Linear Algebra

beds	baths	year
2	1	1985
3	1	1998
4	3	2005
4	2	2020

→ **function** →

predicted **price**
$190K
$254K
$328K
$343K

with matrices...

$$\begin{bmatrix} 2 & 1 & 1985 \\ 3 & 1 & 1998 \\ 4 & 3 & 2005 \\ 4 & 2 & 2020 \end{bmatrix} \rightarrow \boxed{y = Xc + b} \rightarrow \begin{bmatrix} 190000 \\ 254000 \\ 328000 \\ 343000 \end{bmatrix}$$

X

$$\begin{bmatrix} 2 & 1 & 1985 \\ 3 & 1 & 1998 \\ 4 & 3 & 2005 \\ 4 & 2 & 2020 \end{bmatrix} \cdot \begin{bmatrix} 41.46 \\ 10.36 \\ 1.70 \end{bmatrix} + - 3277.31$$

c

b

dot product

Linear Algebra

beds	baths	year
2	1	1985
3	1	1998
4	3	2005
4	2	2020

→ **function** →

predicted **price**
$190K
$254K
$328K
$343K

with matrices...

$$\begin{bmatrix} 2 & 1 & 1985 \\ 3 & 1 & 1998 \\ 4 & 3 & 2005 \\ 4 & 2 & 2020 \end{bmatrix} \rightarrow \boxed{y = Xc + b} \rightarrow \begin{bmatrix} 190000 \\ 254000 \\ 328000 \\ 343000 \end{bmatrix}$$

X c b

$$\begin{bmatrix} 2 & 1 & 1985 \\ 3 & 1 & 1998 \\ 4 & 3 & 2005 \\ 4 & 2 & 2020 \end{bmatrix} \cdot \begin{bmatrix} 41.46 \\ 10.36 \\ 1.70 \end{bmatrix} + - 3277.31$$

dot product

x c x c x c b

$$2 * 41.46 + 1 * 10.36 + 1985 * 1.7 - 3277.31$$

$$= 190000$$

Linear Algebra

beds	baths	year
2	1	1985
3	1	1998
4	3	2005
4	2	2020

→ **function** →

predicted **price**
$190K
$254K
$328K
$343K

with matrices...

$$\begin{bmatrix} 2 & 1 & 1985 \\ 3 & 1 & 1998 \\ 4 & 3 & 2005 \\ 4 & 2 & 2020 \end{bmatrix} \rightarrow \boxed{y = Xc + b} \rightarrow \begin{bmatrix} 190000 \\ 254000 \\ 328000 \\ 343000 \end{bmatrix}$$

X c b

$$\begin{bmatrix} 2 & 1 & 1985 \\ 3 & 1 & 1998 \\ 4 & 3 & 2005 \\ 4 & 2 & 2020 \end{bmatrix} \cdot \begin{bmatrix} 41.46 \\ 10.36 \\ 1.70 \end{bmatrix} + - 3277.31$$

dot product

x c x c x c b

$$3 * 41.46 + 1 * 10.36 + 1998 * 1.7 - 3277.31$$

$$= 254000$$

Linear Algebra

beds	baths	year
2	1	1985
3	1	1998
4	3	2005
4	2	2020

→ **function** →

predicted **price**
$190K
$254K
$328K
$343K

with matrices...

$$\begin{bmatrix} 2 & 1 & 1985 \\ 3 & 1 & 1998 \\ 4 & 3 & 2005 \\ 4 & 2 & 2020 \end{bmatrix} \rightarrow \boxed{y = Xc + b} \rightarrow \begin{bmatrix} 190000 \\ 254000 \\ 328000 \\ 343000 \end{bmatrix}$$

X

$$\begin{bmatrix} 2 & 1 & 1985 \\ 3 & 1 & 1998 \\ 4 & 3 & 2005 \\ 4 & 2 & 2020 \end{bmatrix}$$

c

$$\cdot \begin{bmatrix} 41.46 \\ 10.36 \\ 1.70 \end{bmatrix}$$

b

$$+ - 3277.31$$

dot product

```
import numpy as np
X = df.values
y = np.dot(X, c) + b
```

Linear Algebra

beds	baths	year
2	1	1985
3	1	1998
4	3	2005
4	2	2020

→ **function** →

predicted **price**
$190K
$254K
$328K
$343K

with matrices...

note! Some resources will use **A** instead of **X** and **x** instead of **c**

$$\begin{bmatrix} 2 & 1 & 1985 \\ 3 & 1 & 1998 \\ 4 & 3 & 2005 \\ 4 & 2 & 2020 \end{bmatrix} \rightarrow \boxed{y = Xc + b} \rightarrow \begin{bmatrix} 190000 \\ 254000 \\ 328000 \\ 343000 \end{bmatrix}$$

X

$$\begin{bmatrix} 2 & 1 & 1985 \\ 3 & 1 & 1998 \\ 4 & 3 & 2005 \\ 4 & 2 & 2020 \end{bmatrix} \cdot \begin{bmatrix} 41.46 \\ 10.36 \\ 1.70 \end{bmatrix} + - 3277.31$$

c

b

dot product

```
import numpy as np
X = df.values
y = np.dot(X, c) + b
```

Linear Algebra

```
y = x ** 2
```

not linear

```
y = 3*c0 + -2*c1 + 0.5*c2 + ... + 10*c49
```

linear

with matrices...

$$\begin{bmatrix} 2 & 1 & 1985 \\ 3 & 1 & 1998 \\ 4 & 3 & 2005 \\ 4 & 2 & 2020 \end{bmatrix} \rightarrow \boxed{y = Xc + b} \rightarrow \begin{bmatrix} 190000 \\ 254000 \\ 328000 \\ 343000 \end{bmatrix}$$

X

$$\begin{bmatrix} 2 & 1 & 1985 \\ 3 & 1 & 1998 \\ 4 & 3 & 2005 \\ 4 & 2 & 2020 \end{bmatrix}$$

$\cdot$ c $\begin{bmatrix} 41.46 \\ 10.36 \\ 1.70 \end{bmatrix}$ + b -3277.31

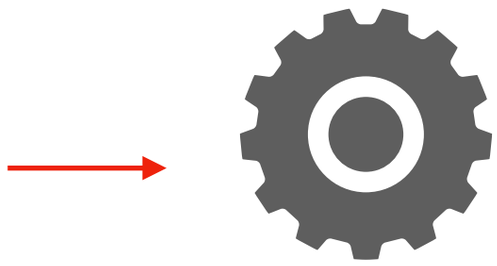
dot product

```
import numpy as np
X = df.values
y = np.dot(X, c) + b
```

Calculus

training data

beds	baths	year	**price**
1	1	1980	$140K
3	1	1990	$240K
3	4	2004	$295K



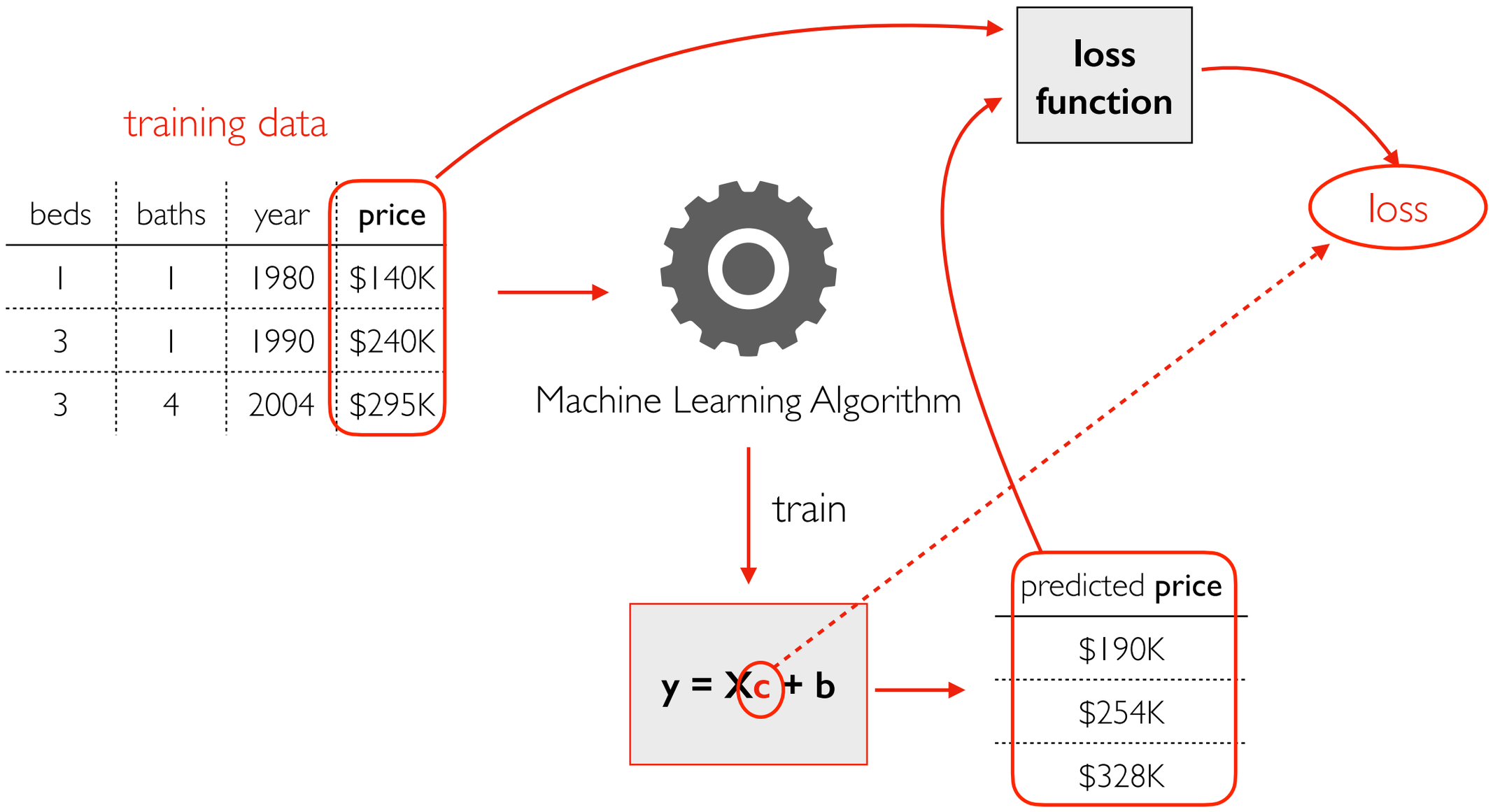
Machine Learning Algorithm

train

y = Xc + b

predicted **price**
$190K
$254K
$328K

Calculus



how do we optimize **c** to minimize **loss**?
Important concepts: derivative, gradient

Parallelism

Parallelism

- doing multiple things at the same time
- requires multiple cores

GPUs (graphics processing units)

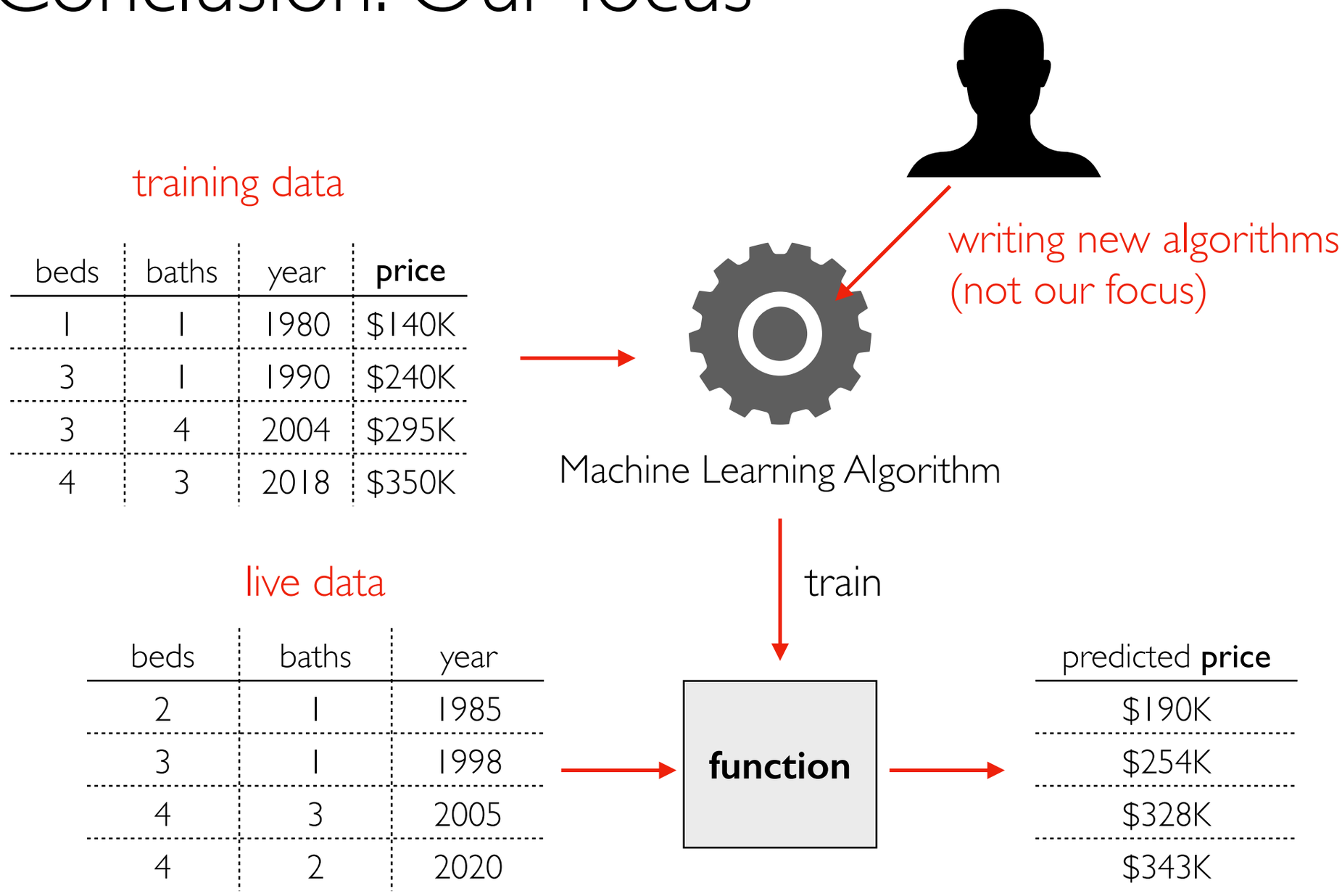
- graphics involves many of the same operation
- better to have many weaker cores working at once than fewer faster cores
- modern GPUs may have 1000s of cores (in contrast to 10s for CPUs)

Scientific Computing

- GPUs can greatly speed up key ML operations
  - multiplying matrices
  - computing gradients
- We'll learn `pytorch` for this...

Practitioners

Conclusion: Our focus



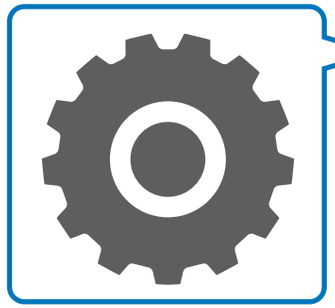
many functions are **models** that can be used to predict

Conclusion: Our focus

how can we clean this up?

training data

beds	baths	year	**price**
1	1	1980	$140K
3	1	1990	$240K
3	4	2004	$295K
4	3	2018	$350K



which algorithm (from sklearn?) should we pick, and how should we configure it?

Machine Learning Algorithm

is it working well? (evaluation)

train

live data

beds	baths	year
2	1	1985
3	1	1998
4	3	2005
4	2	2020

function

predicted **price**
$190K
$254K
$328K
$343K

many functions are **models** that can be used to predict