10-315
Machine Learning
Problem Formulation

Instructor: Pat Virtue

Today

Autoencoder (Aliens) (previous slides)

- Features

ML Problem Formulation

- Task input and output
- Task, Performance, Experience
- Data and notation
- Examples: Iris Classification and Car Price Regression

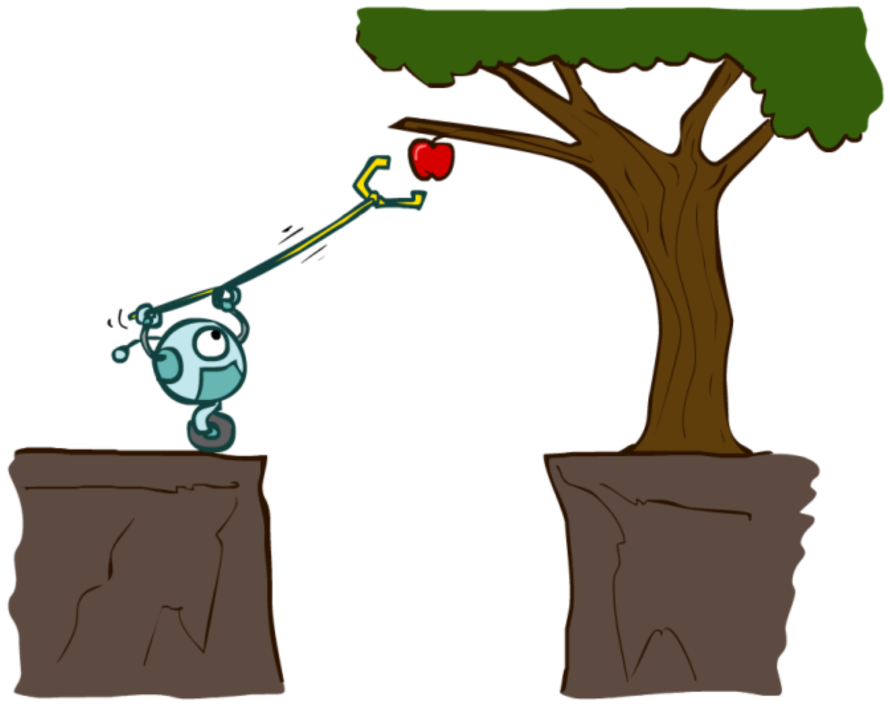
ML Training and Models

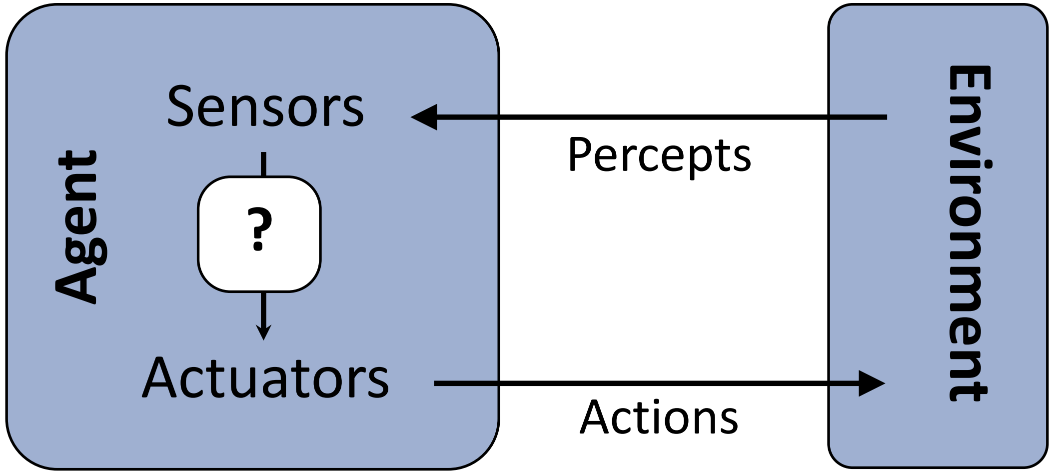
- Linear
- Memorization
- Nearest Neighbor

ML Problem Formulation

Agents

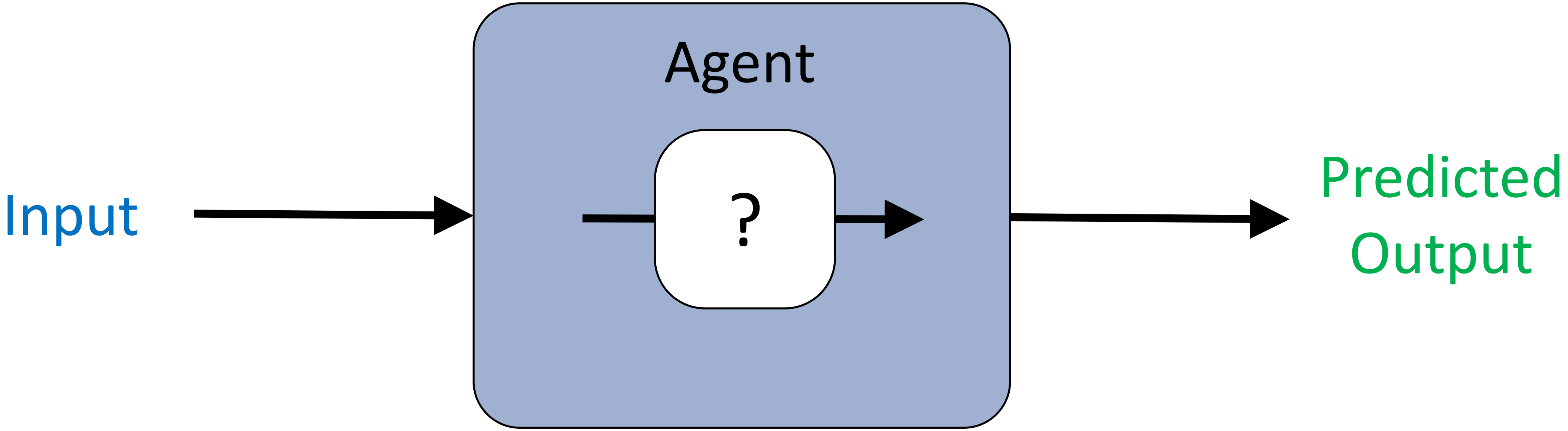
- An **agent** is an entity that *perceives* and *acts*.
- Actions can have an effect on the *environment.*
- The specific *sensors* and *actuators* affect what the agent is capable of *perceiving* and what *actions* it is capable of taking





Slide credit: ai.berkeley.edu

Agent: Simple Input/Output Task



Task Input and Output

Input	Task	Output
Petal measurements	HW1 Iris classification	Category
Time of day	HW1 Traffic prediction	Traffic Volume
Image	HW1 Image classification	Category
Image	Image denoising	Image
Text	Text to image generation	Image
???	HW1 Face generation	Image

Task: Face Generation

https://thispersondoesnotexist.com/

Machine Learning Problem Formulation

Three components <*T,P,E*>:

1. Task, *T*
2. Performance measure, *P*
3. Experience, *E*

Definition of learning:

A computer program **learns** if its performance at tasks in *T*, as measured by *P*, improves with experience *E*

Definition from (Mitchell, 1997)

Machine Learning Problem Formulation

Task

Formalize the task as a mapping from input to output

x y

Experience

Data! Task experience examples will usually be pairs:
(input, measured output)

x y

Performance measure

Objective function that gives a single numerical value representing how well the system performs for a given dataset

- Classification: error rate
- Regression: mean squared error

Notation

$$h(x) \rightarrow \hat{y}$$

$$\mathcal{D} = \left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$$

$$\boxed{\frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\left(y^{(i)} \neq \hat{y}^{(i)}\right)}$$

$$\frac{1}{N}\sum_{i=1}^{N} \left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

ML Problem Formulation

Notation alert: Indicator function

$$\mathbb{I}(z) = \mathbf{1}(z) = \begin{cases} 1 & \text{if } z \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

Task

Formalize the task as a mapping from input to output

$$h(x) \rightarrow \hat{y}$$

Experience

Data! Task experience examples will usually be pairs:
(input, measured output)

$$\mathcal{D} = \left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$$

Performance measure

Objective function that gives a single numerical value representing how well the system performs for a given dataset

- Classification: error rate
- Regression: mean squared error

$$\frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\left(y^{(i)} \neq \hat{y}^{(i)}\right)$$

$$\frac{1}{N}\sum_{i=1}^{N} \left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

Experience: Data and Notation

Example Dataset: Fisher Iris Dataset

Fisher (1936) used 150 measurements of flowers from 3 different species: Iris setosa (0), Iris virginica (1), Iris versicolor (2) collected by Anderson (1936)

Species	Sepal Length	Sepal Width	Petal Length	Petal Width
0	4.3	3.0	1.1	0.1
0	4.9	3.6	1.4	0.1
0	5.3	3.7	1.5	0.2
1	4.9	2.4	3.3	1.0
1	5.7	2.8	4.1	1.3
1	6.3	3.3	4.7	1.6
2	5.9	3.0	5.1	1.8

Images and full dataset: https://en.wikipedia.org/wiki/Iris_flower_data_set

Example Dataset: Fisher Iris Dataset

```
from sklearn import datasets

iris = datasets.load_iris()
X = iris.data
y = iris.target
```

Assume samples in data are i.i.d.

Dataset notation

$$\mathcal{D} = \left\{\left(y^{(i)}, \mathbf{x}^{(i)}\right)\right\}_{i=1}^{N}$$

$$= \left\{\left(\boxed{y^{(i)}}, \boxed{x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, x_4^{(i)}}\right)\right\}_{i=1}^{N}$$

Linear algebra can represent all data

$\mathbf{y} \in \{0,1,2\}^N$

$\vec{x}$

$X \in \mathbb{R}^{N \times 4}$ (design matrix)

		X_1	X_2	X_3	X_4
	Species	Sepal Length	Sepal Width	Petal Length	Petal Width
1	0	4.3	3.0	1.1	0.1
2	0	4.9	3.6	1.4	0.1
3	0	5.3	3.7	1.5	0.2
4	1	4.9	2.4	3.3	1.0
5	1	5.7	2.8	4.1	1.3
6	1	6.3	3.3	4.7	1.6
	2	5.9	3.0	5.1	1.8

Images and full dataset: https://en.wikipedia.org/wiki/Iris_flower_data_set

Example Dataset: Fisher Iris Dataset

```
from sklearn import datasets

iris = datasets.load_iris()
X = iris.data
y = iris.target
```

Assume samples in data are i.i.d.

Dataset notation

$$\mathcal{D} = \left\{\left(\boxed{y^{(i)}}, \boxed{\mathbf{x}^{(i)}}\right)\right\}_{i=1}^{N}$$

$$= \left\{\left(\boxed{y^{(i)}}, \boxed{x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, x_4^{(i)}}\right)\right\}_{i=1}^{N}$$

Data point $i = 6$: $\left(y^{(6)}, \mathbf{x}^{(6)}\right)$

Species	Sepal Length	Sepal Width	Petal Length	Petal Width
0	4.3	3.0	1.1	0.1
0	4.9	3.6	1.4	0.1
0	5.3	3.7	1.5	0.2
1	4.9	2.4	3.3	1.0
1	5.7	2.8	4.1	1.3
1	6.3	3.3	4.7	1.6
2	5.9	3.0	5.1	1.8

Images and full dataset: https://en.wikipedia.org/wiki/Iris_flower_data_set

Example Dataset: Fisher Iris Dataset

```
from sklearn import datasets

iris = datasets.load_iris()
X = iris.data
y = iris.target
```

Assume samples in data are i.i.d.

Dataset notation

$$\mathcal{D} = \left\{\left(y^{(i)}, \mathbf{x}^{(i)}\right)\right\}_{i=1}^{N}$$

$$= \left\{\left(y^{(i)}, x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, x_4^{(i)}\right)\right\}_{i=1}^{N}$$

Species	Sepal Length	Sepal Width	Petal Length	Petal Width
0	4.3	3.0	1.1	0.1
0	4.9	3.6	1.4	0.1
0	5.3	3.7	1.5	0.2
1	4.9	2.4	3.3	1.0
1	5.7	2.8	4.1	1.3
1	6.3	3.3	4.7	1.6
2	5.9	3.0	5.1	1.8

Linear algebra can represent all data

$\boxed{\mathbf{y}} \in \{0,1,2\}^N$

$\boxed{X} \in \mathbb{R}^{N \times 4}$ (design matrix)

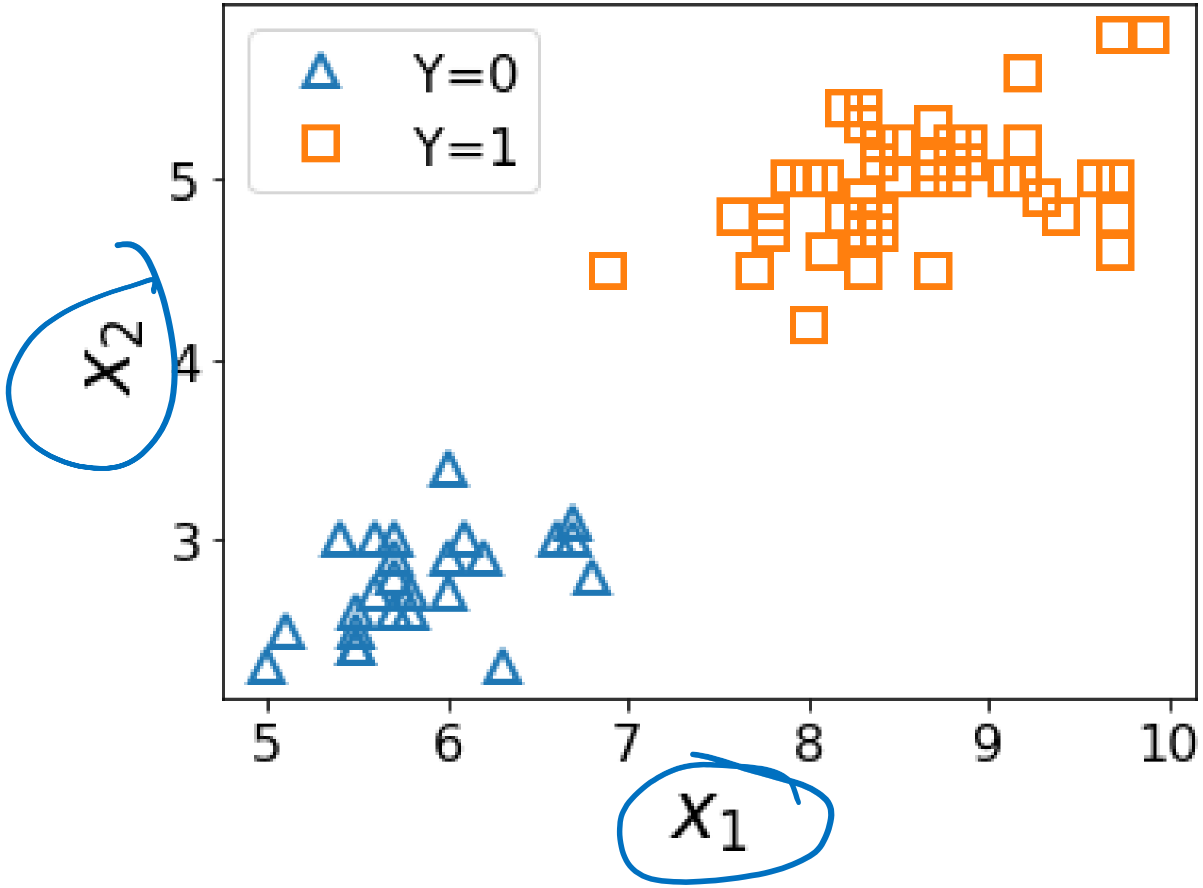
Images and full dataset: https://en.wikipedia.org/wiki/Iris_flower_data_set

Task: Classification

ML Task: Classification

Predict species label from first two input measurements

$h(\mathbf{x}) \rightarrow \hat{y}$



y x_1 x_2

Species	Sepal Length	Sepal Width
0	4.3	3.0
0	4.9	3.6
0	5.3	3.7
1	4.9	2.4
1	5.7	2.8
1	6.3	3.3

Images and full dataset: https://en.wikipedia.org/wiki/Iris_flower_data_set

Classification

Notation alert: Indicator function

$$\mathbb{I}(z) = \mathbf{1}(z) = \begin{cases} 1 & \text{if } z \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

Iris data example

$$\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}, \text{ where } \mathbf{x}^{(i)} \in \mathbb{R}^4, y^{(i)} \in \{0, 1, 2\}$$

Predict species label from input measurements

$$h(\mathbf{x}) \rightarrow \hat{y}$$

Performance measure?

Classification error rate

- Fraction of times $y \neq \hat{y}$ in a given dataset
- $\frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(y^{(i)} \neq \hat{y}^{(i)})$

Species	Sepal Length	Sepal Width	Petal Length	Petal Width
0	4.3	3.0	1.1	0.1
0	4.9	3.6	1.4	0.1
0	5.3	3.7	1.5	0.2
1	4.9	2.4	3.3	1.0
1	5.7	2.8	4.1	1.3
1	6.3	3.3	4.7	1.6
2	5.9	3.0	5.1	1.8

Images and full dataset: https://en.wikipedia.org/wiki/Iris_flower_data_set

ML Tasks

Supervised learning: Pairs of input and output in training data

$$\mathcal{D} = \left\{\left(\mathbf{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N} \qquad h(\mathbf{x}) \to \hat{y}$$

Classification

- Output labels
- $y \in \mathcal{Y}$, where $\mathcal{Y}$ is discrete and order of values has no meaning

Regression

- Output values
- $y \in \mathcal{Y}$, where $\mathcal{Y}$ is usually continuous, order of values has meaning

Unsupervised Tasks

ML Tasks

Unsupervised learning

$$\mathcal{D} = \left\{ \mathbf{x}^{(i)} \right\}_{i=1}^{N} \quad h(\mathbf{x}) \rightarrow ???$$

- Training data has no output values
- Tasks can vary
- Often used to organize data for future (minimally) supervised learning

Task: Face Generation

https://thispersondoesnotexist.com/





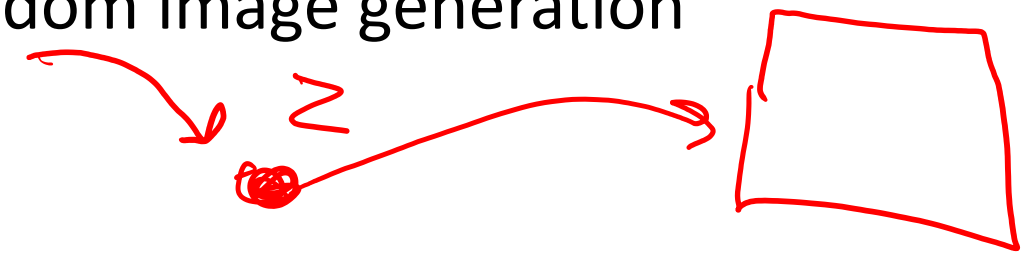
ML Tasks

Unsupervised learning

$$\mathcal{D} = \{\ \mathbf{x}^{(i)}\ \}_{i=1}^{N} \qquad h(\mathbf{x}) \rightarrow ???$$

- Training data has no output values
- Tasks can vary
- Often used to organize data for future (minimally) supervised learning

Example: Unsupervised autoencoder → Random image generation



$$\mathbf{x} \rightarrow \boxed{h(\mathbf{x})} \rightarrow \hat{\mathbf{x}}$$

$$\mathbf{x} \rightarrow \boxed{\boxed{f(\mathbf{x})} \rightarrow \mathbf{z} \rightarrow \boxed{g(\mathbf{z})}} \rightarrow \hat{\mathbf{x}} \qquad\qquad \mathbf{z} \rightarrow \boxed{g(\mathbf{z})} \rightarrow \hat{\mathbf{x}}$$

ML Tasks

Unsupervised learning

$$\mathcal{D} = \{ \mathbf{x}^{(i)} \}_{i=1}^{N} \qquad h(\mathbf{x}) \rightarrow ???$$

- Training data has no output values
- Tasks can vary
- Often used to organize data for future (minimally) supervised learning

Example: Text Generation

$$x_4 \; x_5 \; x_6 \longrightarrow \boxed{\text{Predict}} \longrightarrow x_7$$

Vocab pause

Task

- Prediction
- Inference
- Hypothesis function h(x)
- Classification
- Regression

Experience/Data

Input

- Input feature
- Measurement
- Attribute

Output

- Target
- Class/category/label
- True output
- Measured output
- Predicted output

Supervised

Unsupervised

Performance Measure

- Objective function

Classification

- Error rate
- Accuracy rate

Regression

- Mean squared error

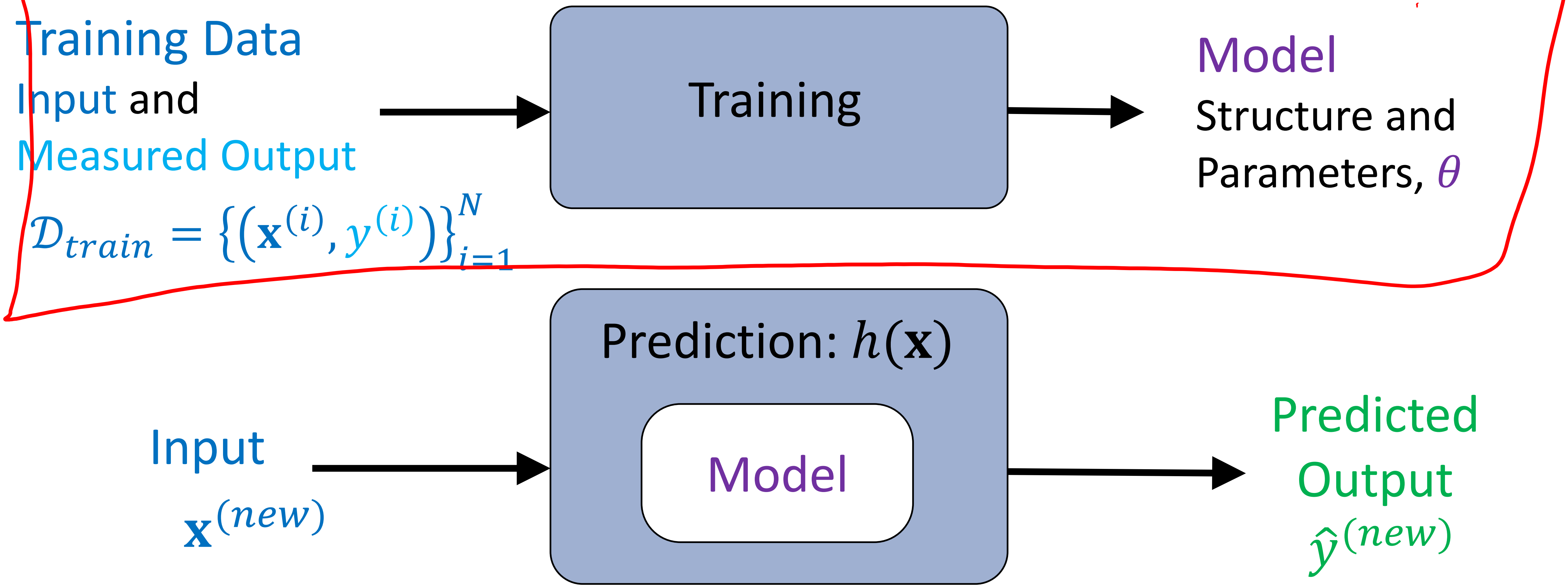
Training

- Model
- Model structure
- Model parameters

Training and ML Models

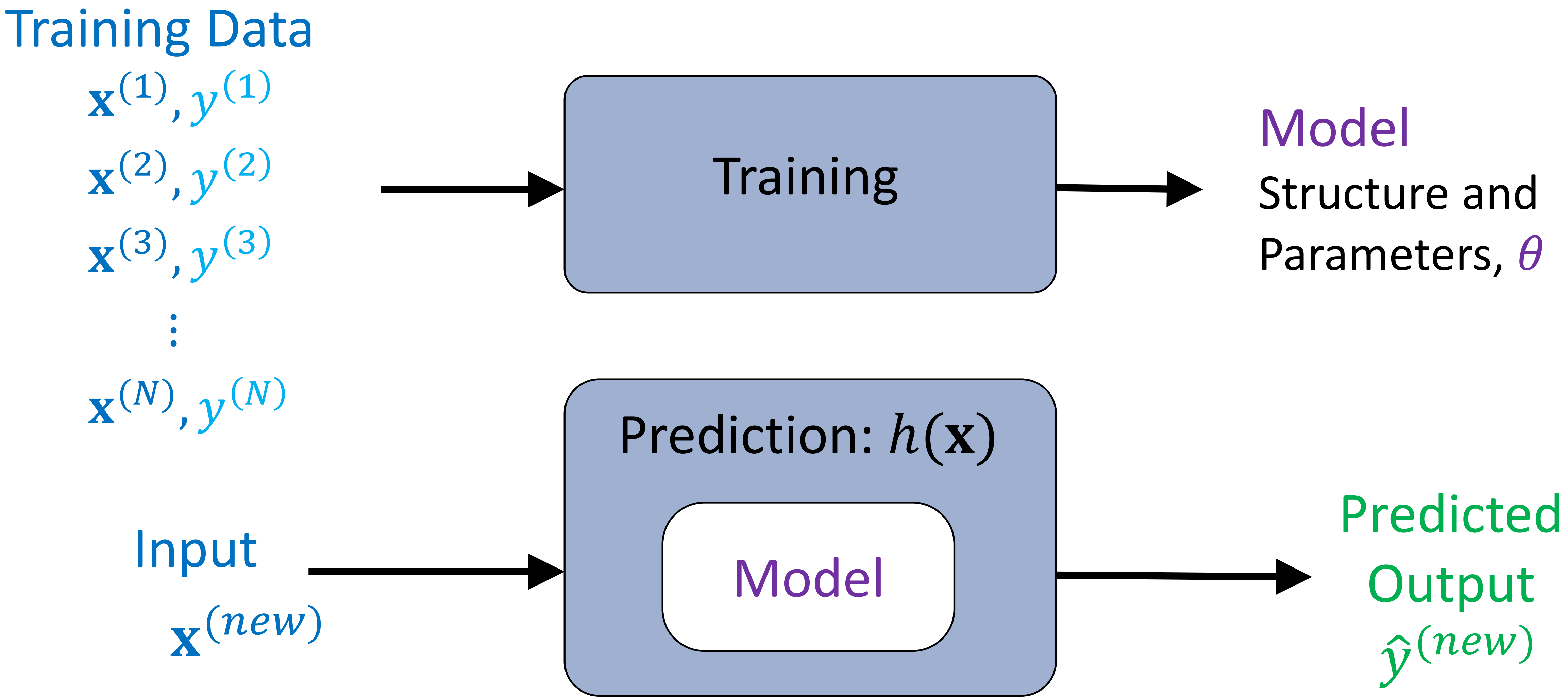
Machine Learning

Using (training) data to learn a model that we'll later use for prediction



Machine Learning

Using (training) data to learn a model that we'll later use for prediction

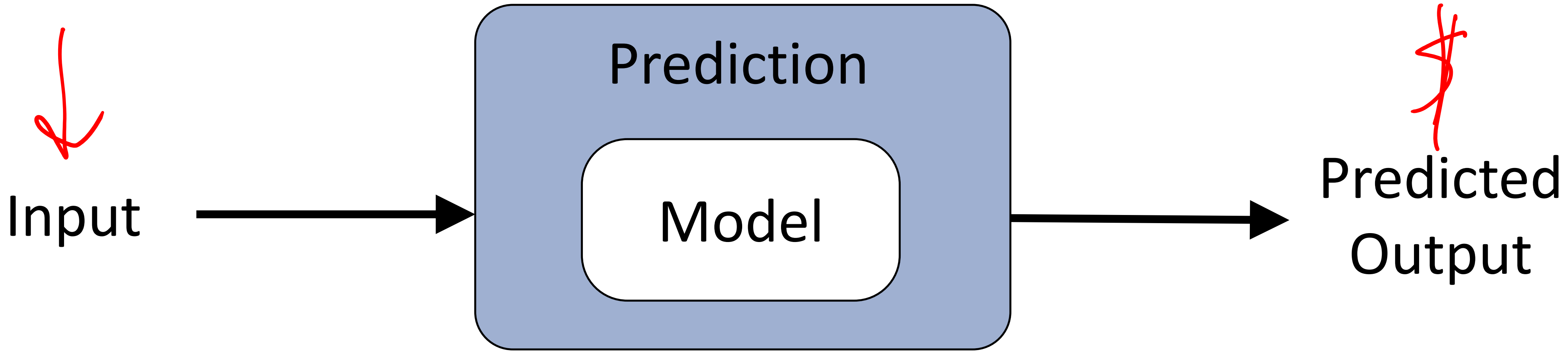


Task: Car Price Prediction

Regression: learning a model to predict a numerical output (but not numbers that just represent categories, that would be classification)

Example

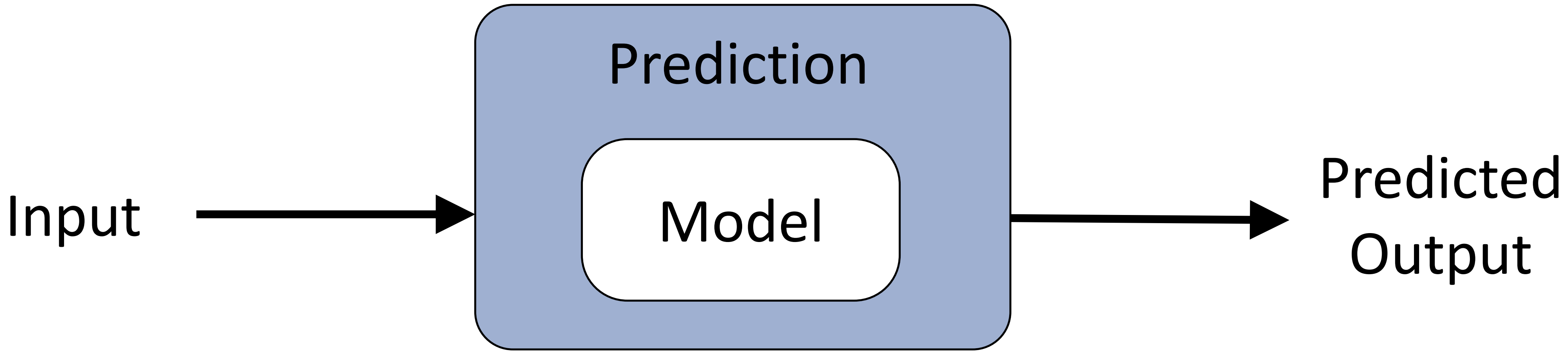
Trying to see how much I should sell my car for.



Task: Car Price Prediction

Regression: learning a model to predict a numerical output (but not numbers that just represent categories, that would be classification)

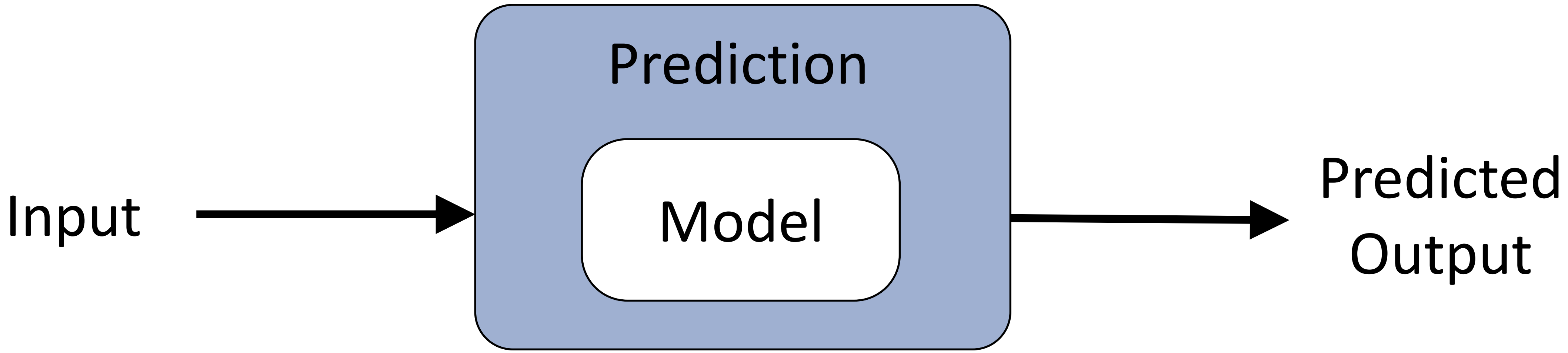
What input features should we use?



Poll 2

Regression: learning a model to predict a numerical output (but not numbers that just represent categories, that would be classification)

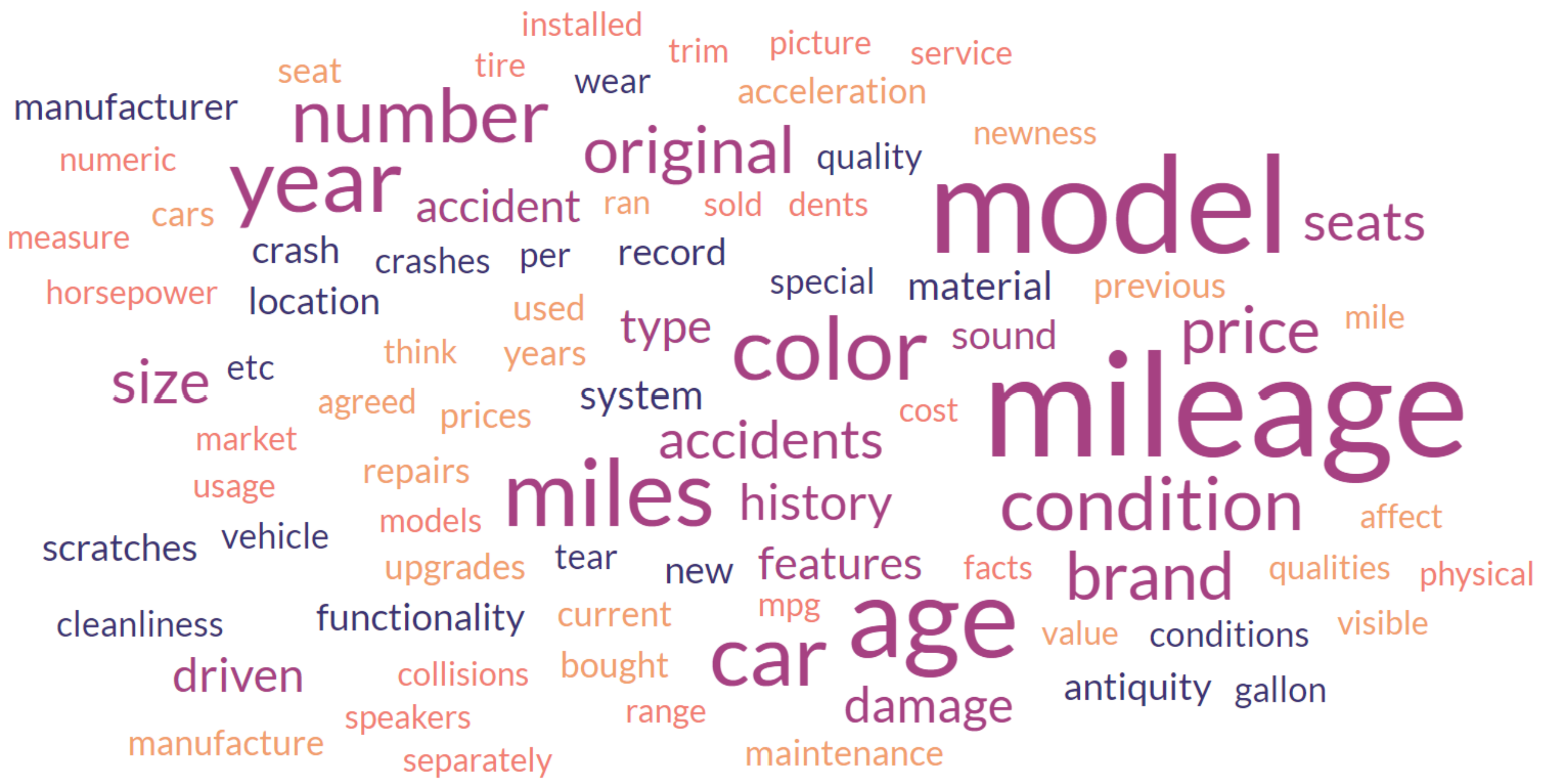
What input features should we use?



Poll 2

Regression: learning a model to predict a numerical output (but not numbers that just represent categories, that would be classification)

What input features should we use?

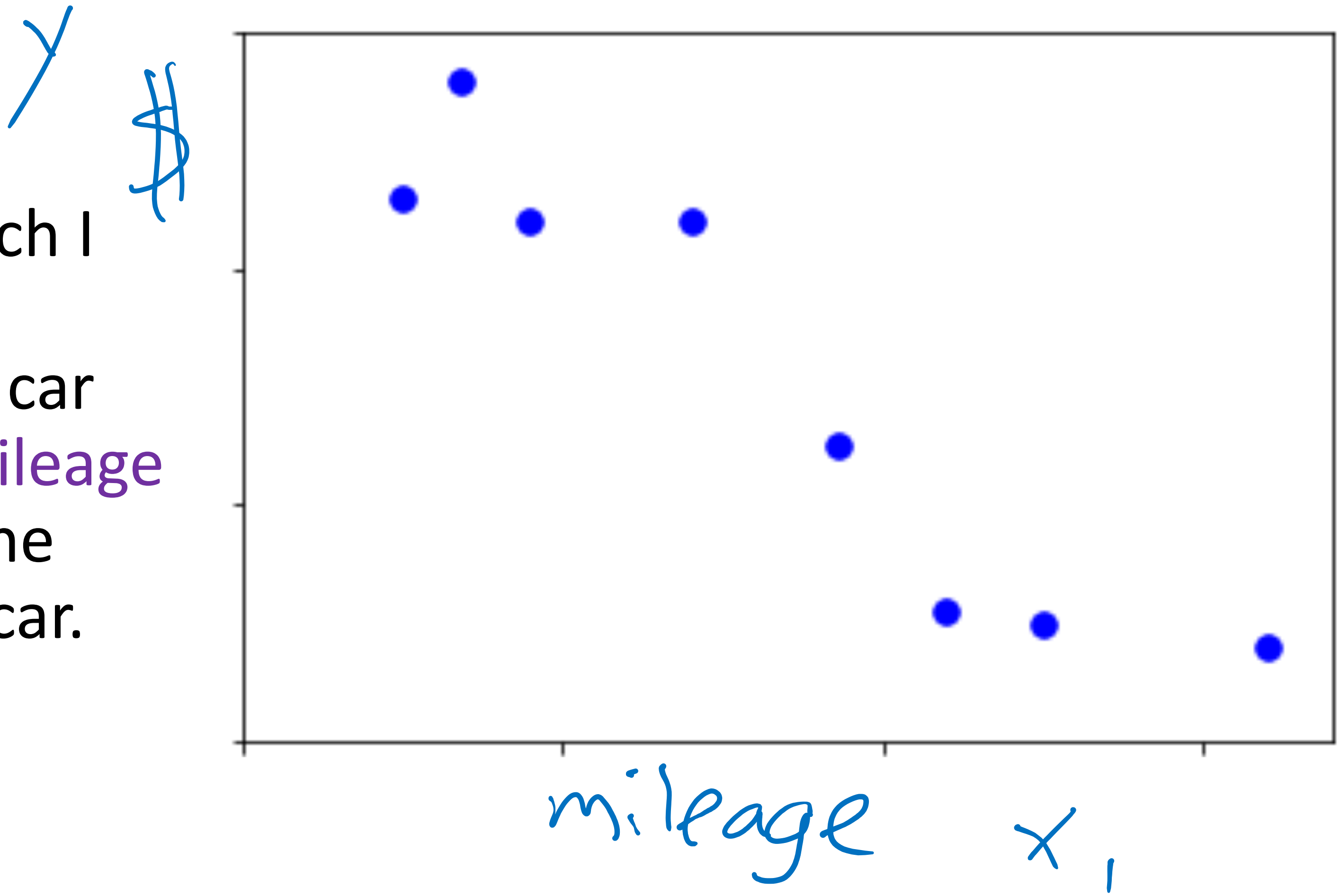


Regression

Regression: learning a model to predict a numerical output (but not numbers that just represent categories, that would be classification)

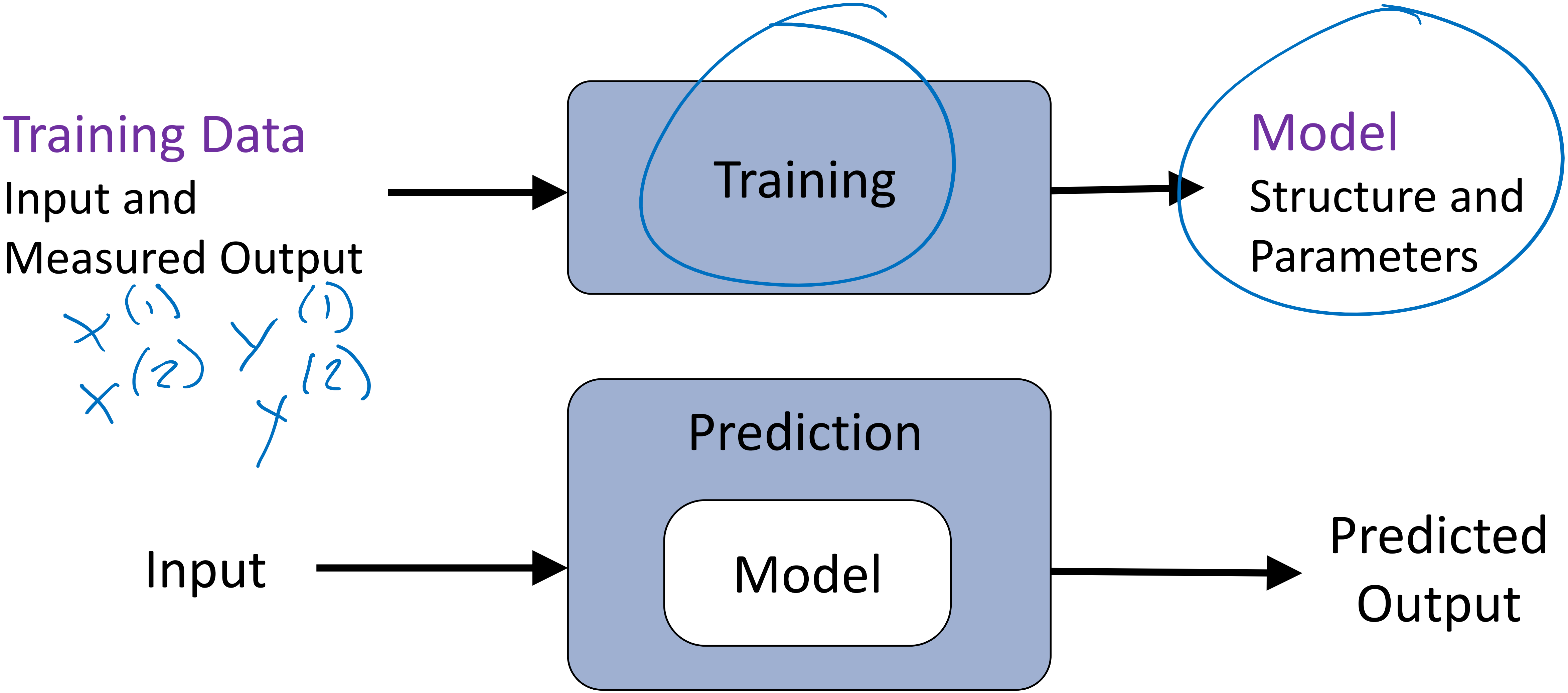
Example

Trying to see how much I should sell my car for. Looking up data from car websites, I find the mileage for a set of cars and the selling price for each car.



Machine Learning

Using (training) data to learn a model that we'll later use for prediction



Regression Model

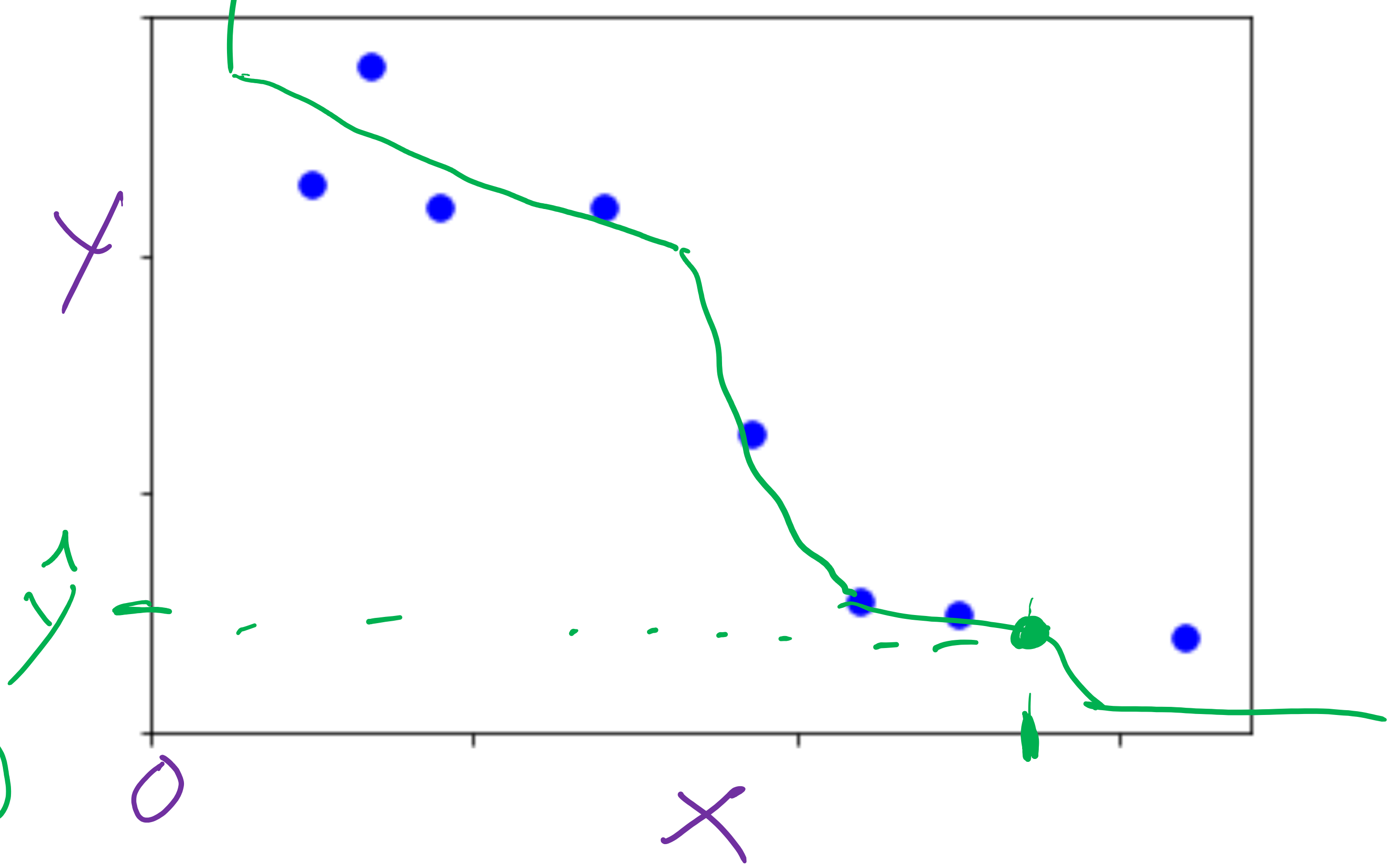
Regression: learning a model to predict a numerical output (but not numbers that just represent categories, that would be classification)

Model?

Human

```
def train(Life)
    → brain

def predict(x_new)
    return brain(x_new)
```



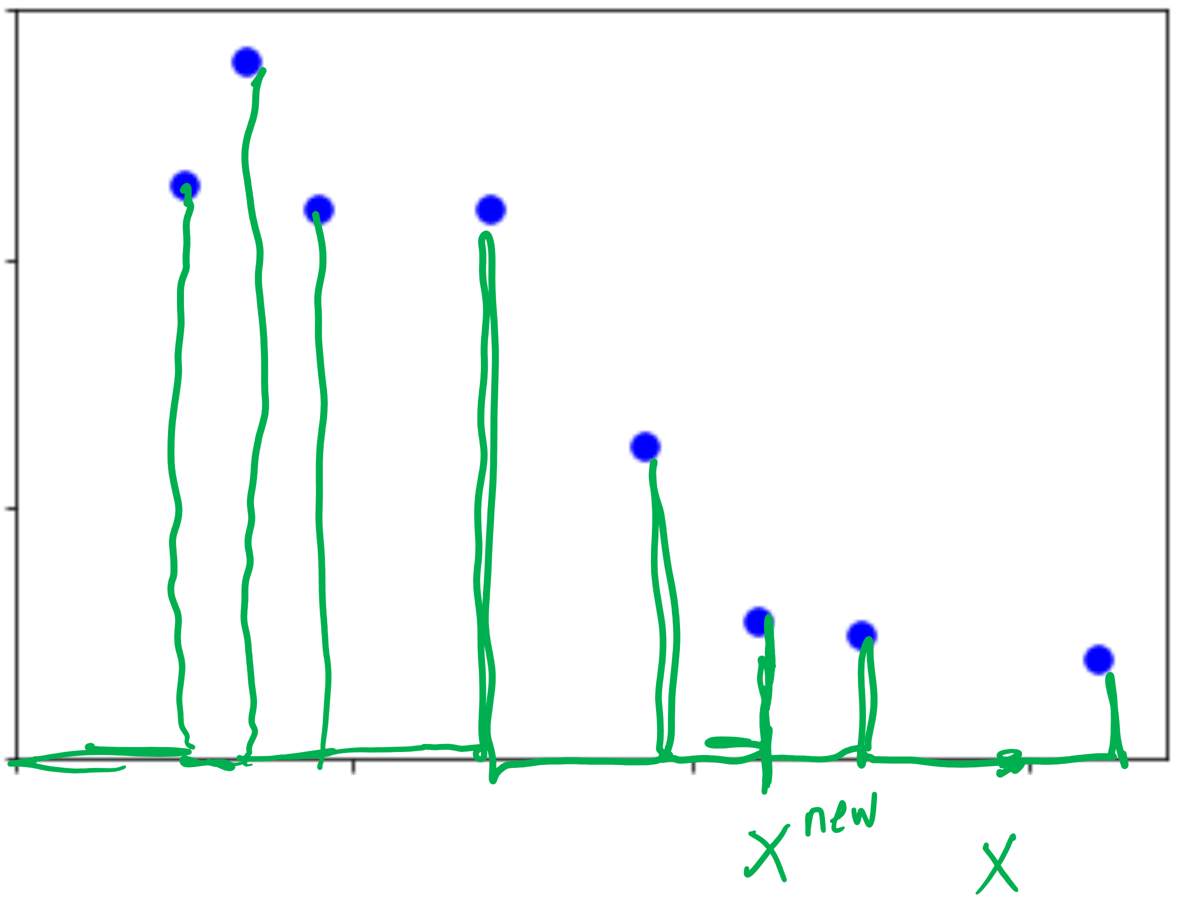
Regression Model

Regression: learning a model to predict a numerical output (but not numbers that just represent categories, that would be classification)

Model: Memorization

```
def train(D)
    self.D = D

def predict(x_new)
    for x, y in self.D
        if x_new == x:
            return y
    return 0
```



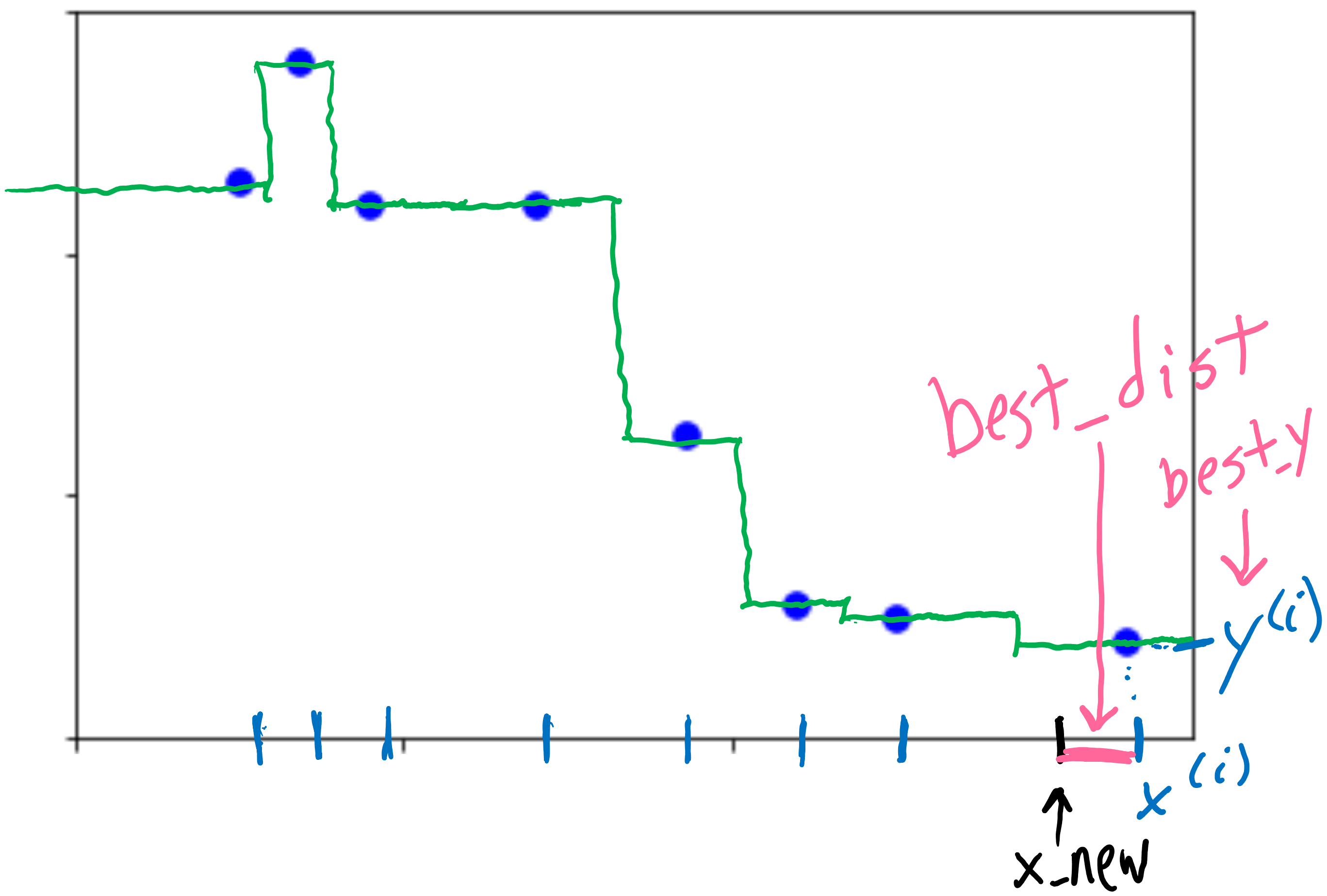
Regression Model

Regression: learning a model to predict a numerical output (but not numbers that just represent categories, that would be classification)

Model: Nearest neighbor

```
def train(D)
    self.D = D

def predict(x_new)
    for x, y in self.D
        if dist(x, x_new) ≤ best_dist
            best_y = y
    return best_y
```

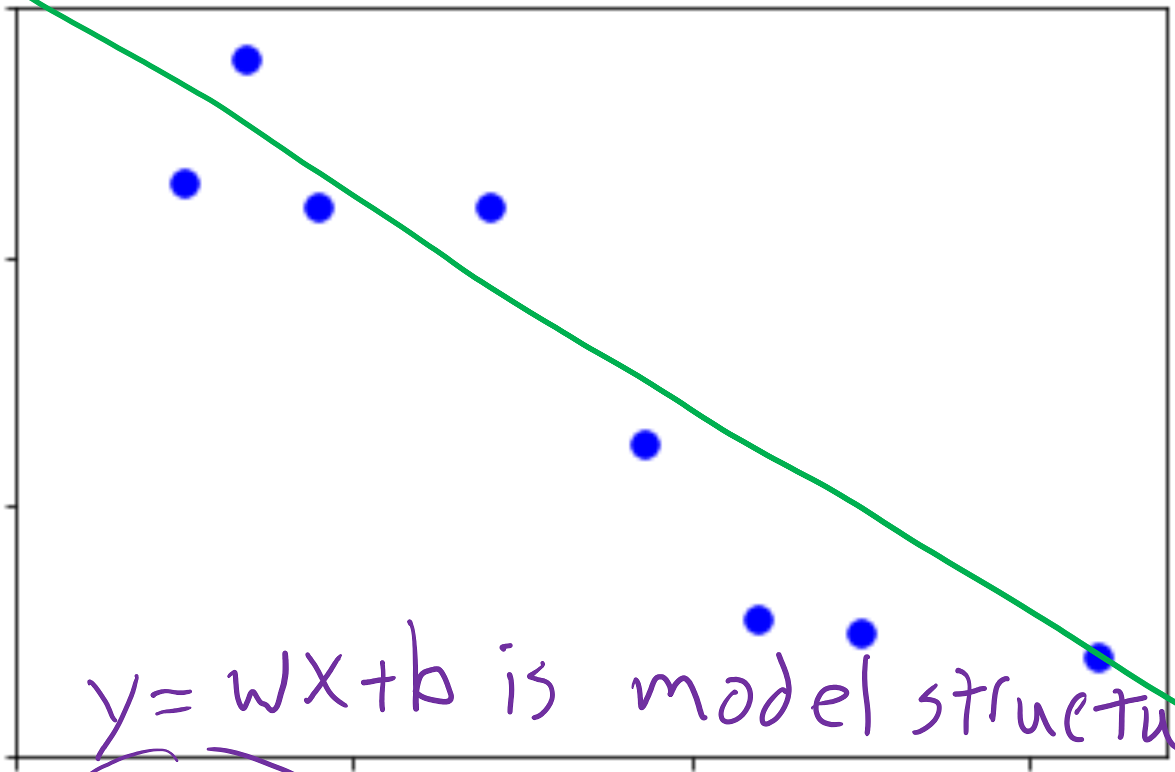


Regression Model

Regression: learning a model to predict a numerical output (but not numbers that just represent categories, that would be classification)

Model: Linear

def train($\mathcal{D}$)
    Find best slope, w
    and intercept, b



def predict(x_new)
    return w*x_new + b

$y = wx + b$ is model structure

$\theta \rightarrow$ w, b are model parameters

Regression Model

Regression: learning a model to predict a numerical output (but not numbers that just represent categories, that would be classification)

Model?

Neural nets?