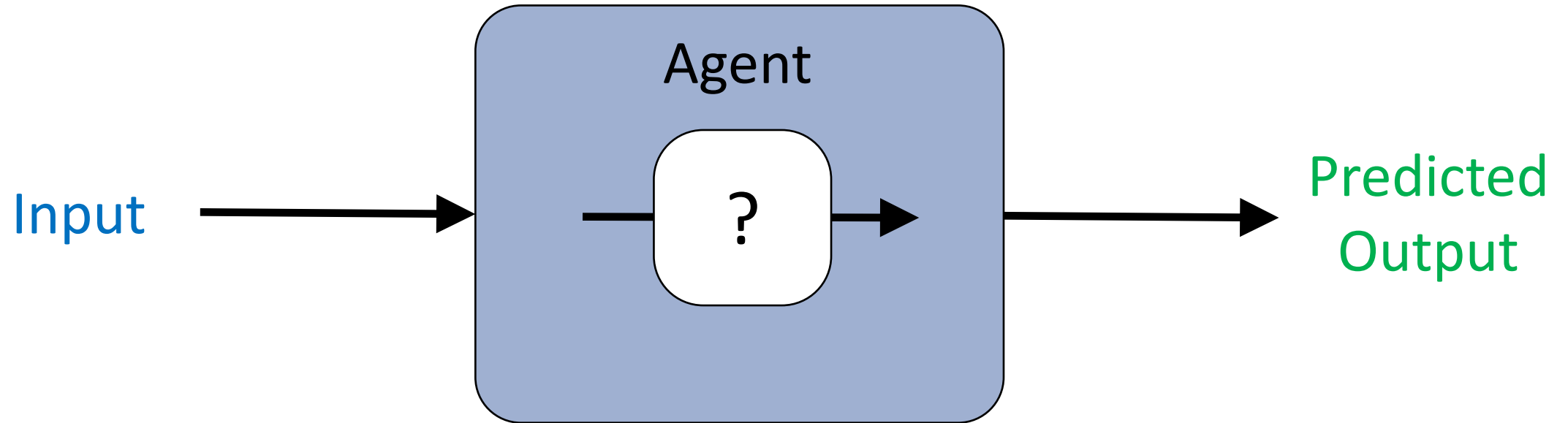


An abstract graphic on the left side of the slide, featuring a sphere-like shape composed of a dense grid of intersecting red, green, and blue lines. The lines are curved and follow the contours of the sphere, creating a complex, woven pattern. The sphere is set against a dark gray background.

10-315 Machine Learning Problem Formulation

Instructor: Pat Virtue

Agent: Simple Input/Output Task



Task Input and Output

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

Today

ML Problem Formulation

- Task input and output
- Task, Performance, Experience
- Data and notation
- Supervised Learning
 - Classification and Regression
- Unsupervised Learning

ML Training and Models

- Nearest Neighbor
- Linear
- Neuron



ML Problem Formulation

Machine Learning Problem Formulation

Three components $\langle T, P, E \rangle$:

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

Machine Learning Problem Formulation

Task

Formalize the task as a mapping from input to output

Notation

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

Experience

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

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

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

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

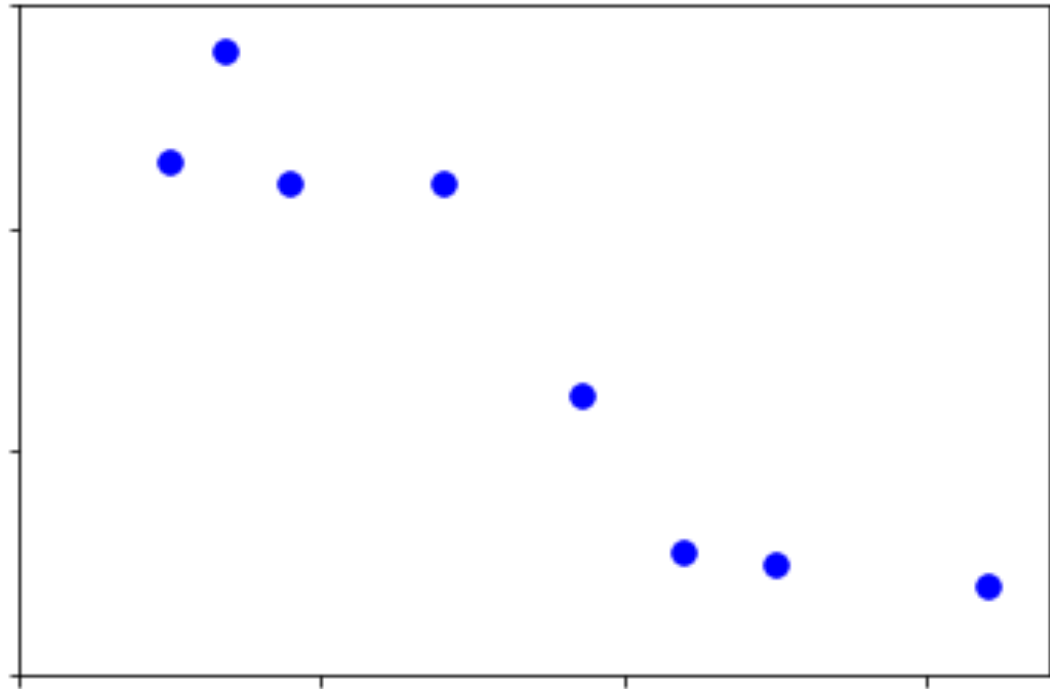
$$\frac{1}{N} \sum_{i=1}^N (y^{(i)} - \hat{y}^{(i)})^2$$

Experience: Data and Notation

Example Dataset: Selling My Car

Example

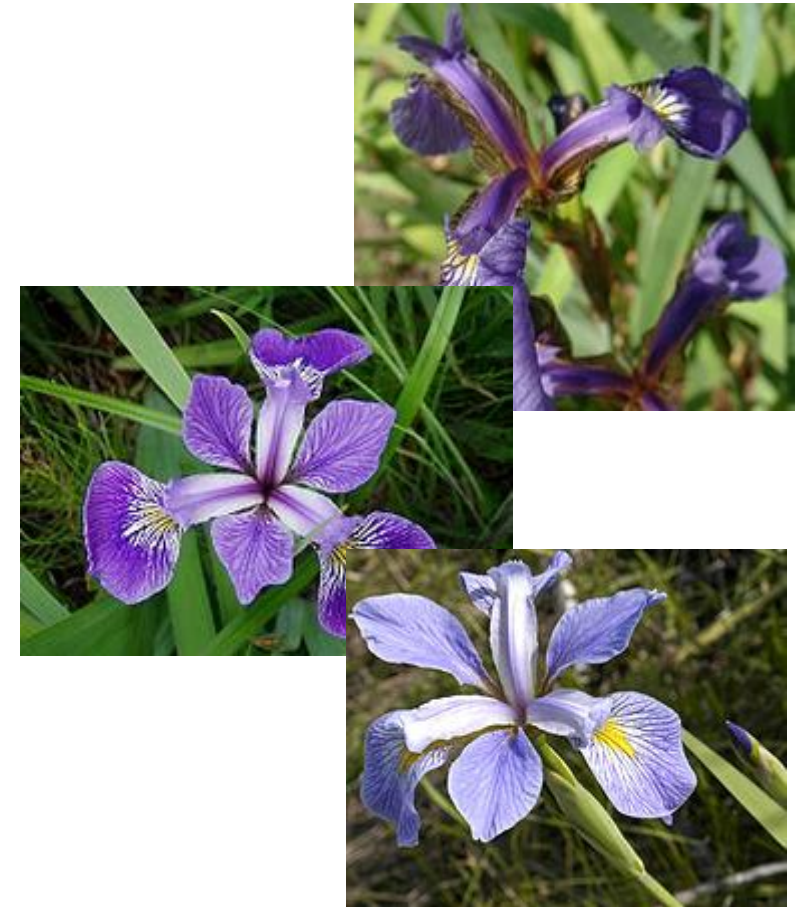
Trying to see how much I should sell my car for. Looking up data from car websites, I find the **mileage** for a set of cars and the selling **price** for each car.



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

Assume samples in data are i.i.d.

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

Dataset notation

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

Linear algebra can represent all data

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

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

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

Example Dataset: Fisher Iris Dataset

Assume samples in data are i.i.d.

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

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 $i = 6$: $(\boxed{y^{(6)}}, \boxed{\mathbf{x}^{(6)}})$

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

Example Dataset: Fisher Iris Dataset

Assume samples in data are i.i.d.

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

Dataset notation

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

Linear algebra can represent all data

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

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

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

ML Data : Supervised vs Unsupervised

Supervised training data:

Pairs of input and output

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

Unsupervised training data:

No output data (i.e., no answers!)

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

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

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

ML Data : Supervised vs Unsupervised

Supervised training data:

Pairs of input and output

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

Unsupervised training data:

No output data (i.e., no answers!)

$$\mathcal{D} = \{(\mathbf{x}^{(i)} \quad)\}_{i=1}^N$$

ML Tasks: Supervised Learning

Supervised learning: Pairs of input and output in training data

$$\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N \quad h(\mathbf{x}) \rightarrow \hat{y}$$

Classification

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

Regression

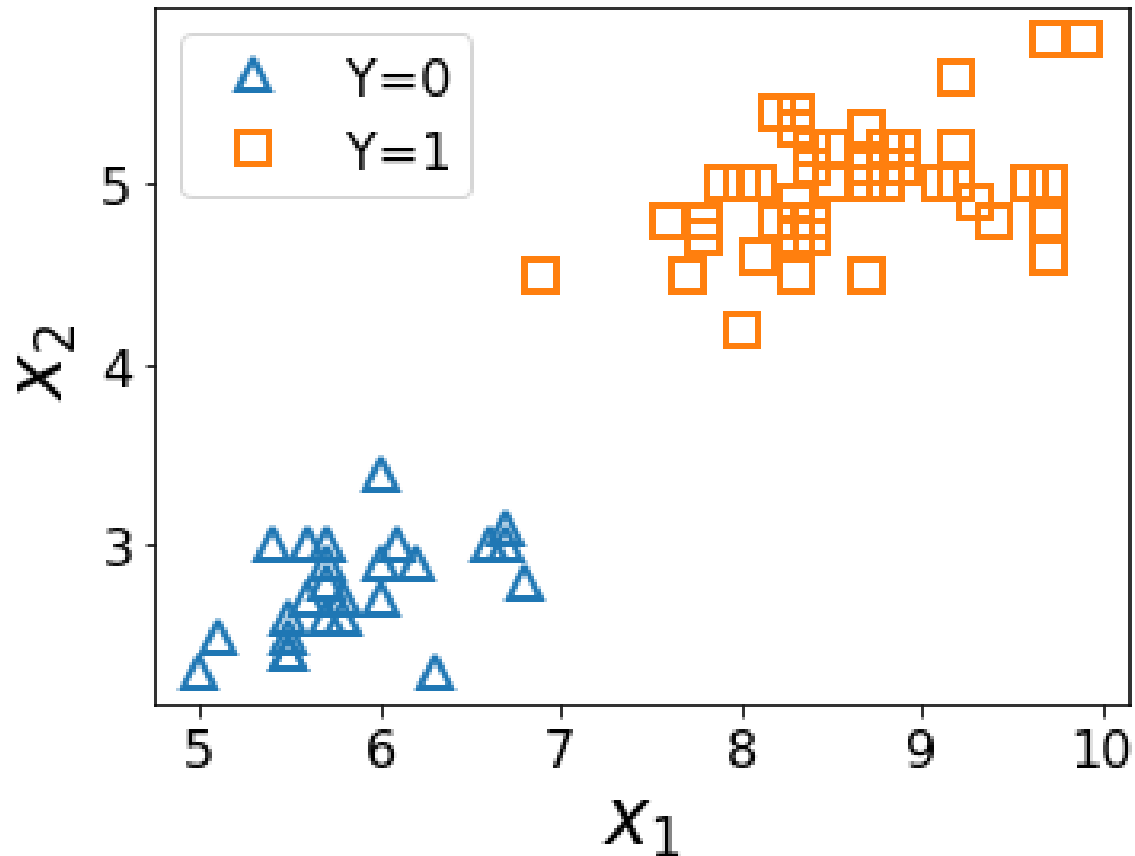
- Output values
- $y \in \mathcal{Y}$, where \mathcal{Y} is usually continuous, order of values has meaning

Task: Classification

ML Task: Classification

Predict species label from first two input measurements

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

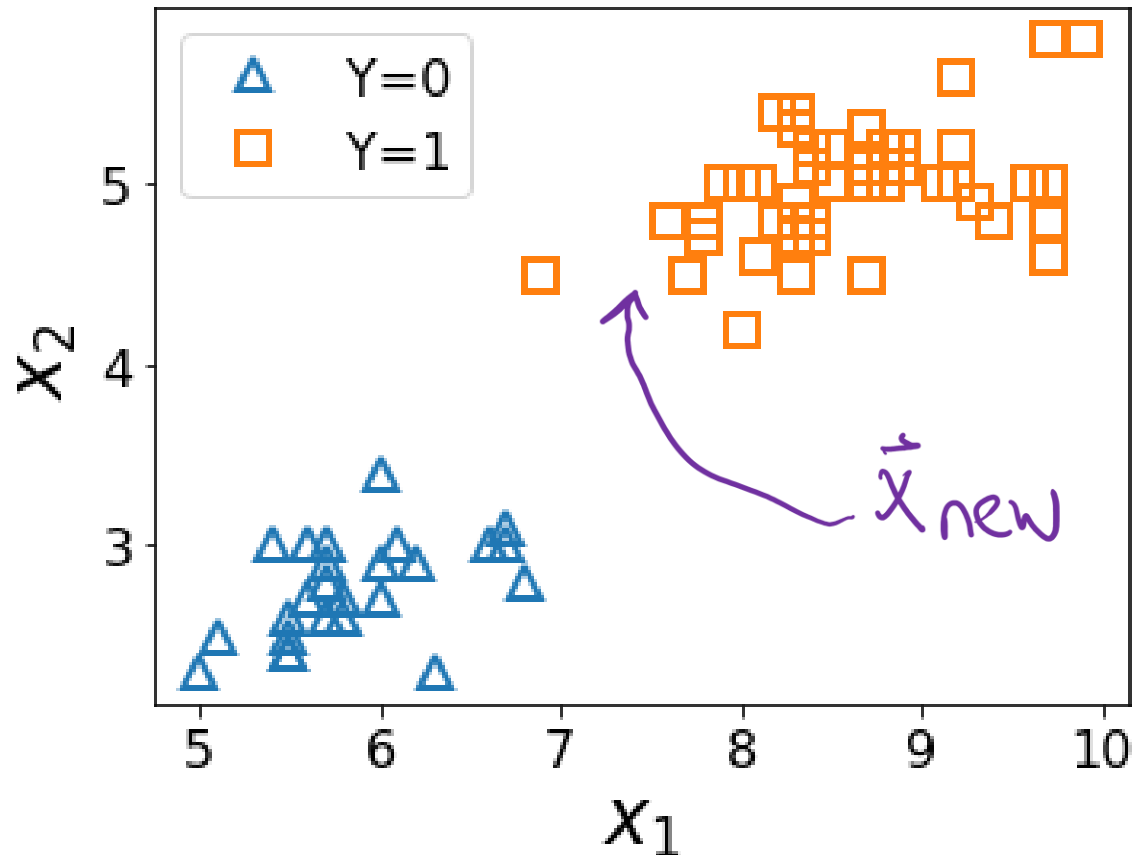


Species	Sepal Length	Sepal Width
0	4.3	3.0
0	4.9	3.6
0	5.3	3.7
1	4.9	2.4
1	5.7	2.8
1	6.3	3.3

ML Task: Classification

Nearest neighbor classification algorithm

Find $\mathbf{x}^{(i)}$ closest to \mathbf{x}_{new} . Then return it's label $y^{(i)}$.



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

\vec{x}_{new}

Classification

Iris data example

Notation alert: Indicator function

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

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

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

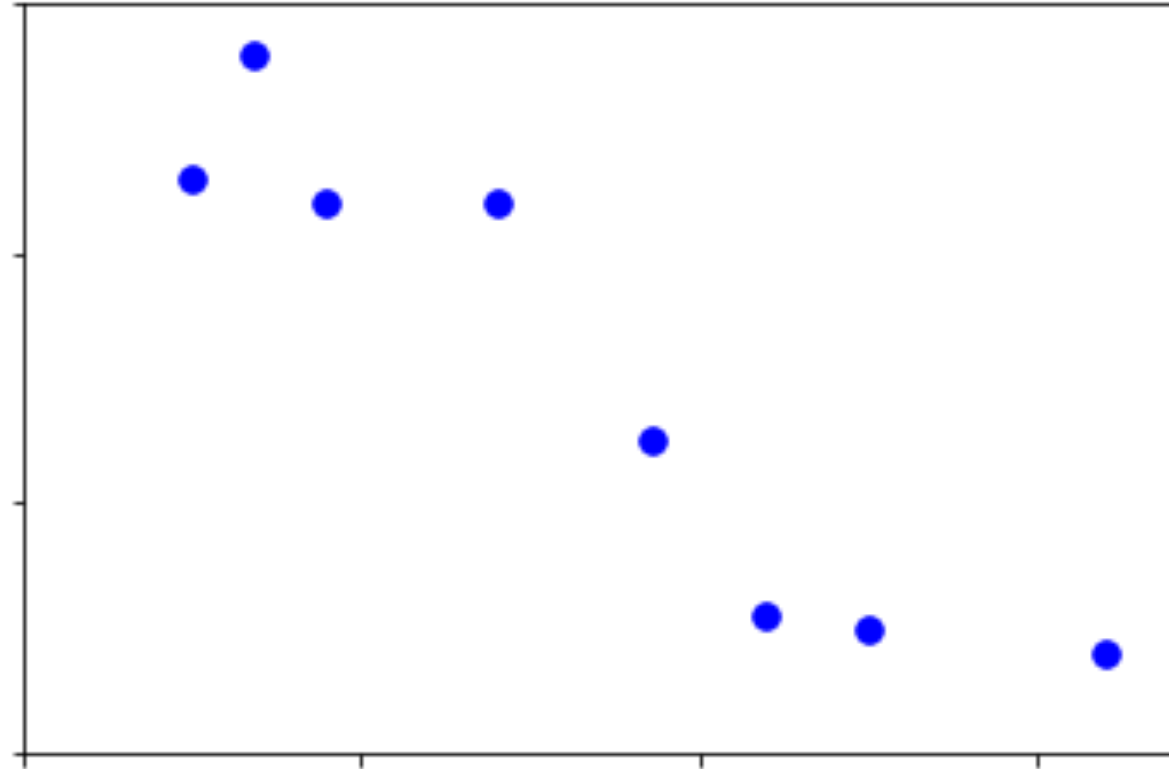
Species	Sepal Length	Sepal Width	Petal Length	Petal Width
0	4.3	3.0	1.1	0.1
0	4.9	3.6	1.4	0.1
0	5.3	3.7	1.5	0.2
1	4.9	2.4	3.3	1.0
1	5.7	2.8	4.1	1.3
1	6.3	3.3	4.7	1.6
2	5.9	3.0	5.1	1.8

ML Task: 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.



Unsupervised Tasks

ML Tasks

Unsupervised learning

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

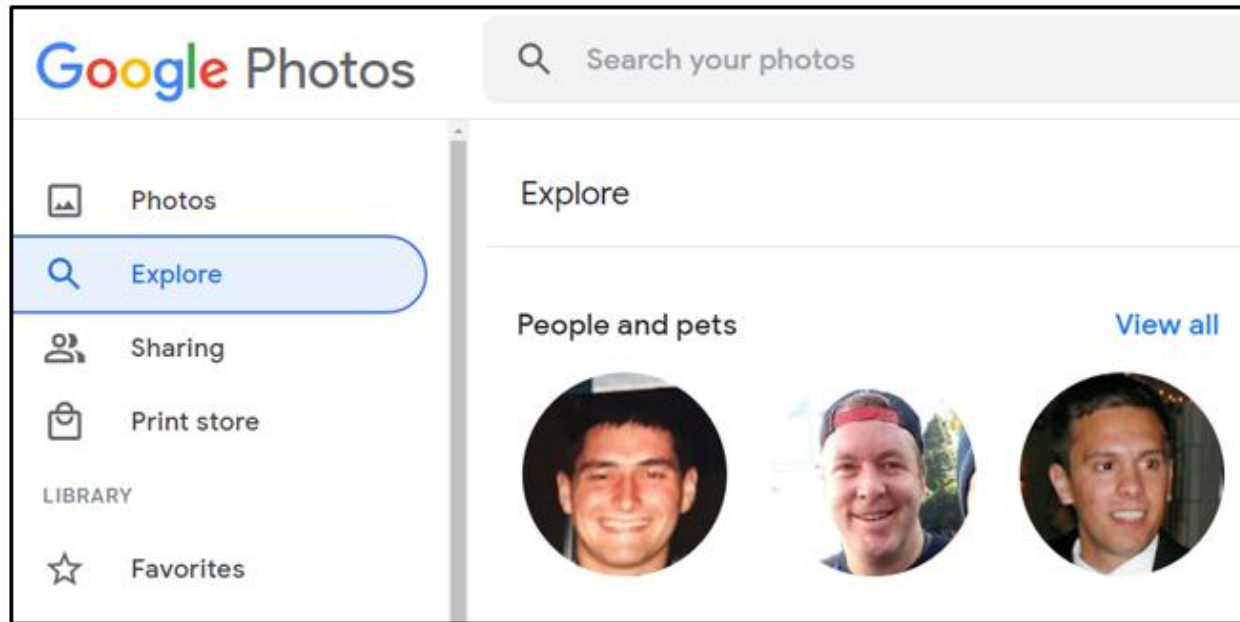
- Training data has no output values
- Tasks can vary
- Often used to organize data for future (minimally) supervised learning

Task: Unsupervised Face Generation

<https://thispersondoesnotexist.com/>






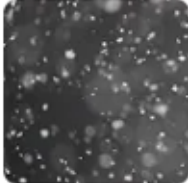











Tasks: Unsupervised Clustering (Photos)



Tasks: Unsupervised Clustering (News)

Google News

 Nor'easter dumps heavy snow and cuts off power to hundreds of thousands across Northeast as many roads remain impassable 4 hours ago · Gene Norman, Artemis Moshtaghian & ... 	 400000 In Maine, New Hampshire and Vermont Without Power After Snowstorm 8 minutes ago · Remy Tumin 
 The Daily Weather Update from FOX Weather: Nor'easter continues dumping snow on New England 56 minutes ago 	 NH forecast video: A few showers as spring nor'easter moves away 2 hours ago · Kevin Skarupa 
 Engineers Pinpoint Cause of Voyager 1 Issue, Are Working on Solution – Voyager 17 hours ago 	 NASA optimistic about resolving Voyager 1 computer problem Mar 27 · Jeff Foust 
Voyager 1's Data Transmission Issue Traced to Memory Corruption, Fix in Progress 1 hour ago 	 Voyager 1 stops communicating with Earth 4 days ago · Ashley Strickland 

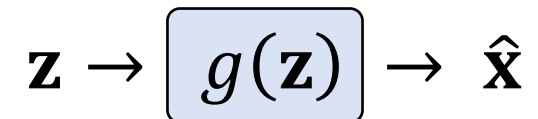
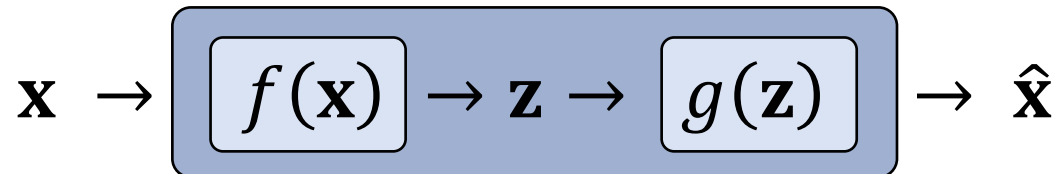
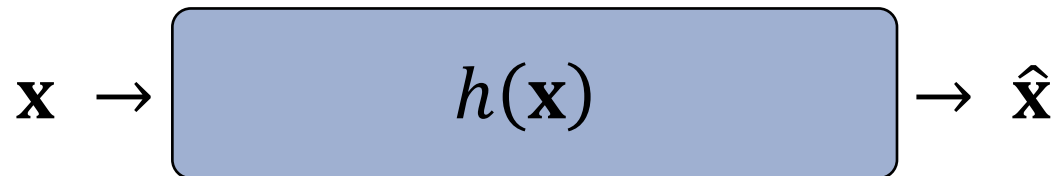
ML Tasks

Unsupervised learning

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

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

Example: Unsupervised autoencoder \rightarrow Random image generation



ML Tasks

Unsupervised learning

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

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

Example: Text Generation

Vocab pause

Task

- Prediction
- Inference
- Hypothesis function
- 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

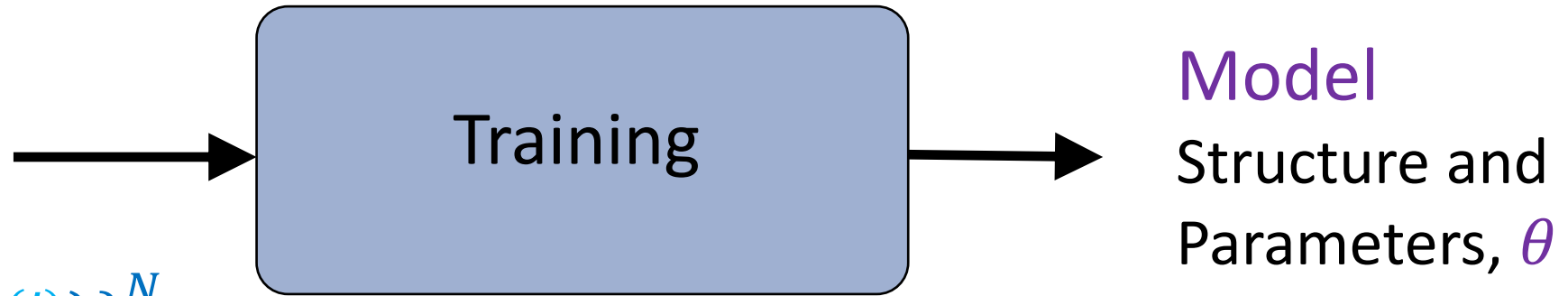
Training and ML Models

Machine Learning

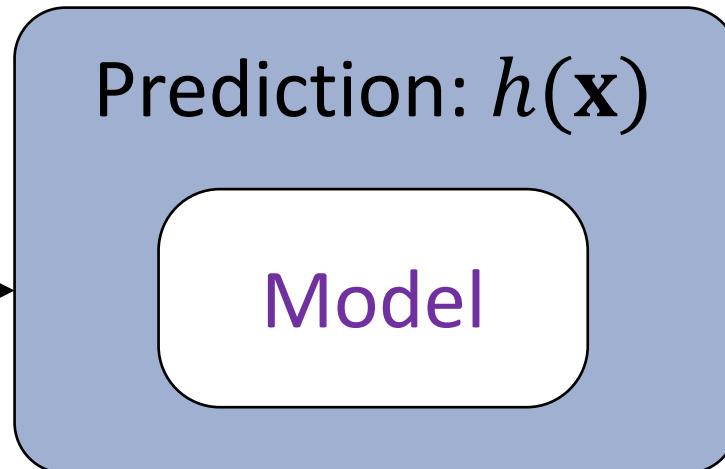
Using (training) data to learn a model that we'll later use for prediction

Training Data
Input and
Measured Output

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



Input
 $\mathbf{x}^{(new)}$



Predicted
Output
 $\hat{y}^{(new)}$

Machine Learning

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

Training Data

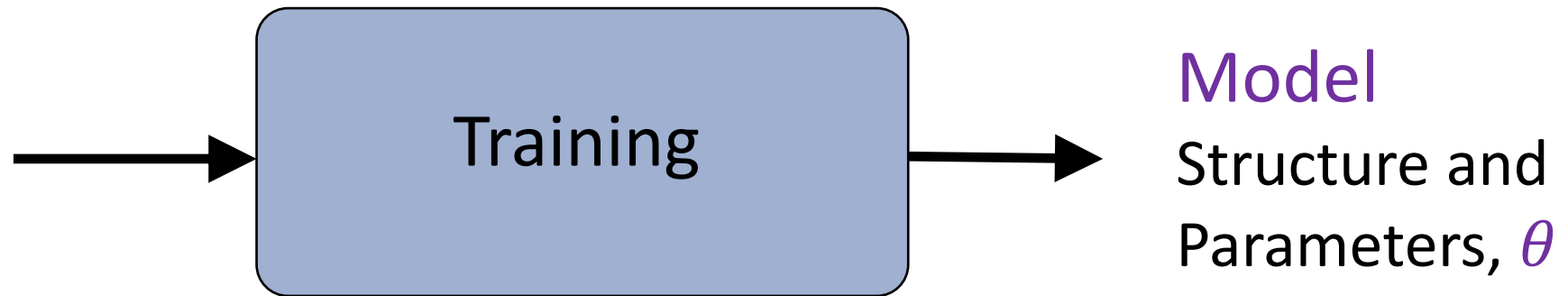
$\mathbf{x}^{(1)}, y^{(1)}$

$\mathbf{x}^{(2)}, y^{(2)}$

$\mathbf{x}^{(3)}, y^{(3)}$

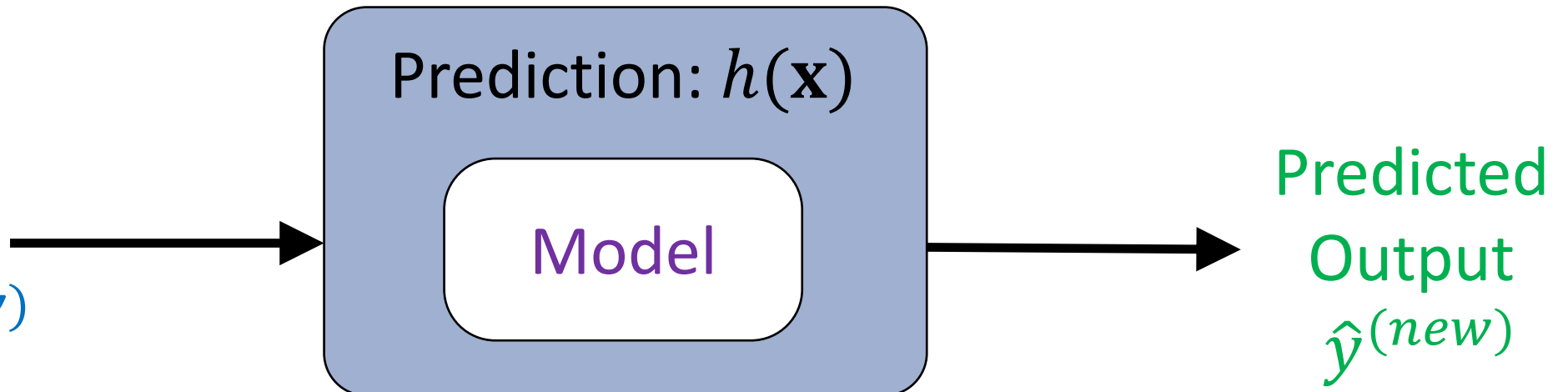
\vdots

$\mathbf{x}^{(N)}, y^{(N)}$



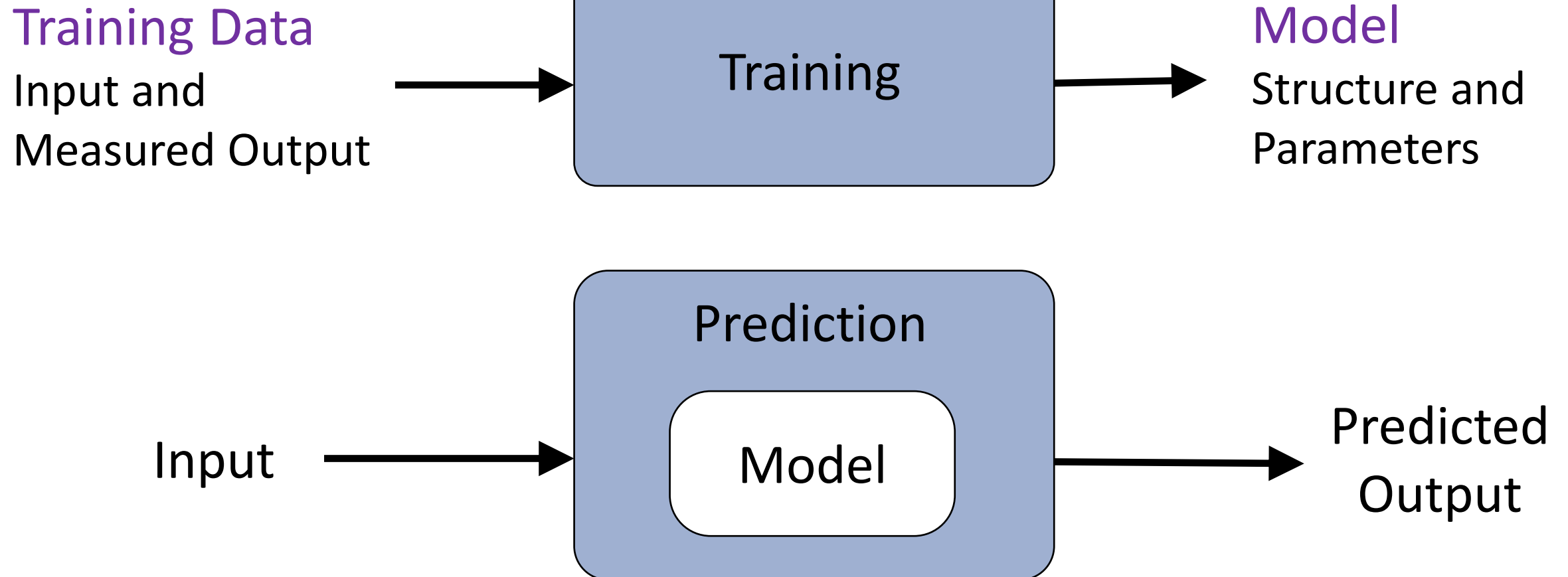
Input

$\mathbf{x}^{(new)}$



Machine Learning

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



Classification

Iris data example

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

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

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

Classification

Iris data example

Notation alert: Indicator function

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

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

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

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

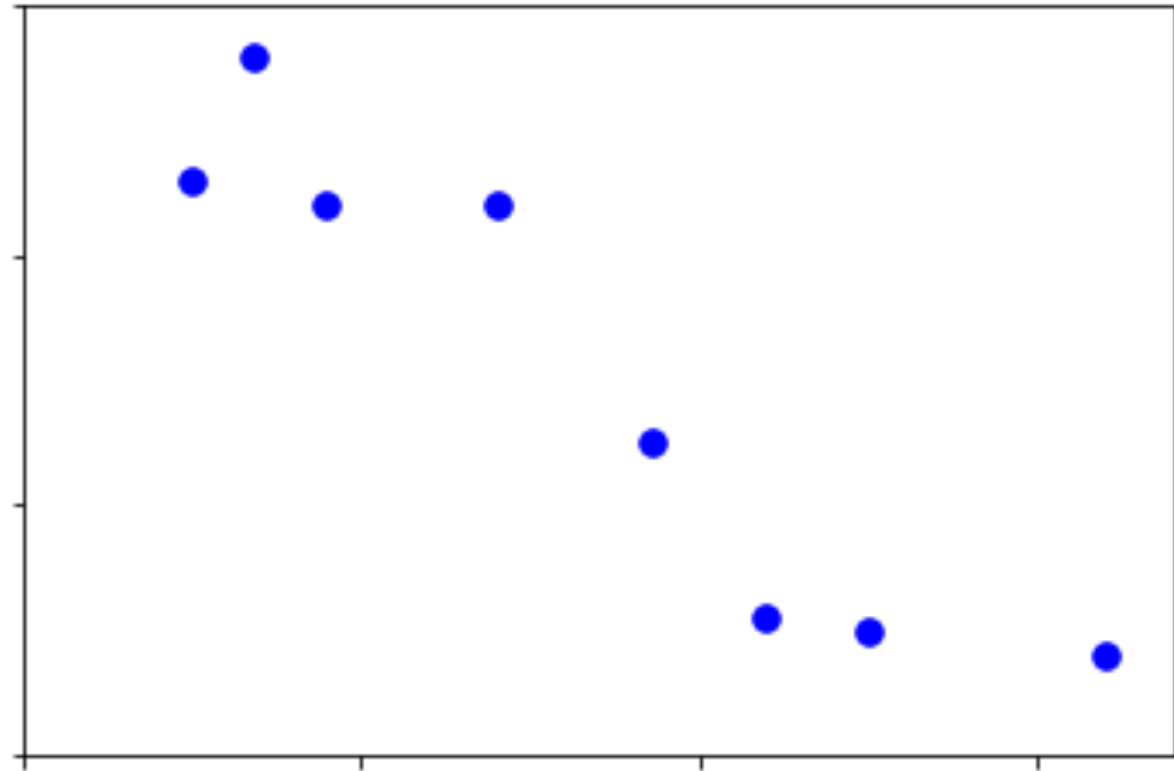
Regression Model

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

Model: Linear

Structure:

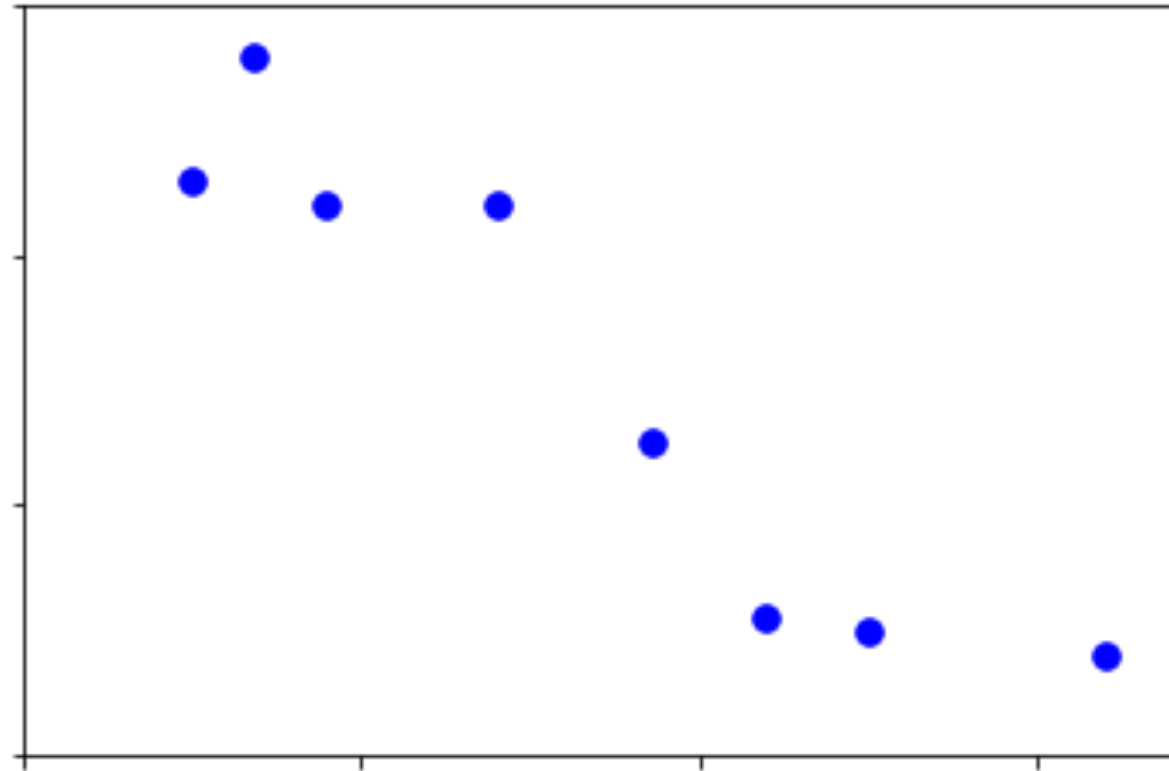
Parameters:



Regression Model

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

How do we know which model and model parameters are best?



Regression Model

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

How do we know which model and model parameters are best?

