Artificial Neural Networks and Deep Learning

Week 1

Intro to Machine Learning

Canonical example

Canonical example

Canonical example

Dog

fluffy	sad looking	showing teeth	ears down	tail between legs	is retriever	growling
1	1	0	1	0	0	0

Data point

Canonical example

Dog

→

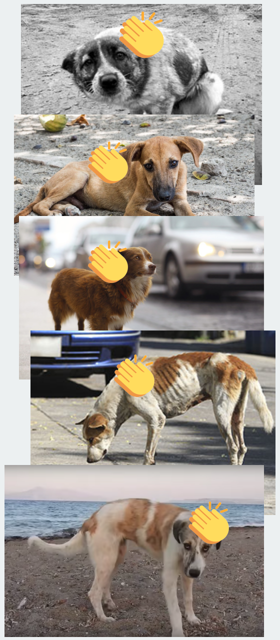
fluffy	sad looking	showing teeth	ears down	tail between legs	is retriever	growling
1	1	0	1	0	0	0

Data point

bites
0

Outcome

Canonical example



Experiments

$$
\begin{array}{ccccccc}
\text{fluffy} & \text{sad looking} & \text{showing teeth} & \text{ears down} & \text{tail between legs} & \text{is retriever} & \text{growling} \\
\end{array}
\qquad
\begin{array}{c}
\text{bites}
\end{array}
$$

$$
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1
\end{bmatrix}
\qquad
\begin{bmatrix}
0 \\ 0 \\ 0 \\ 1 \\ 1
\end{bmatrix}
$$

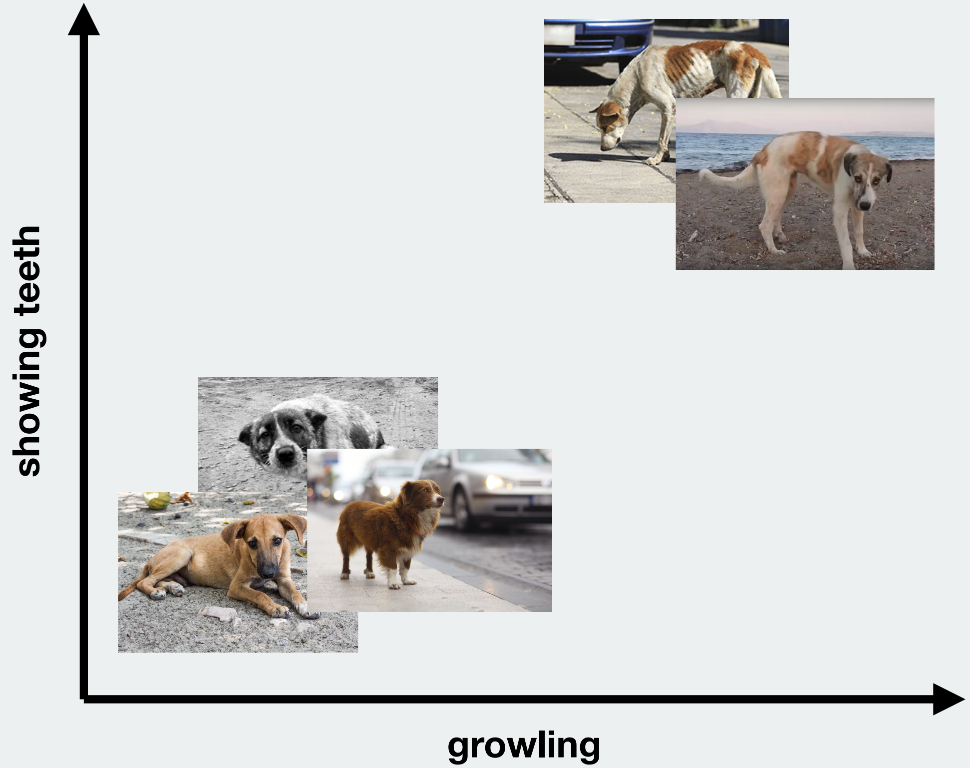
Data points

X

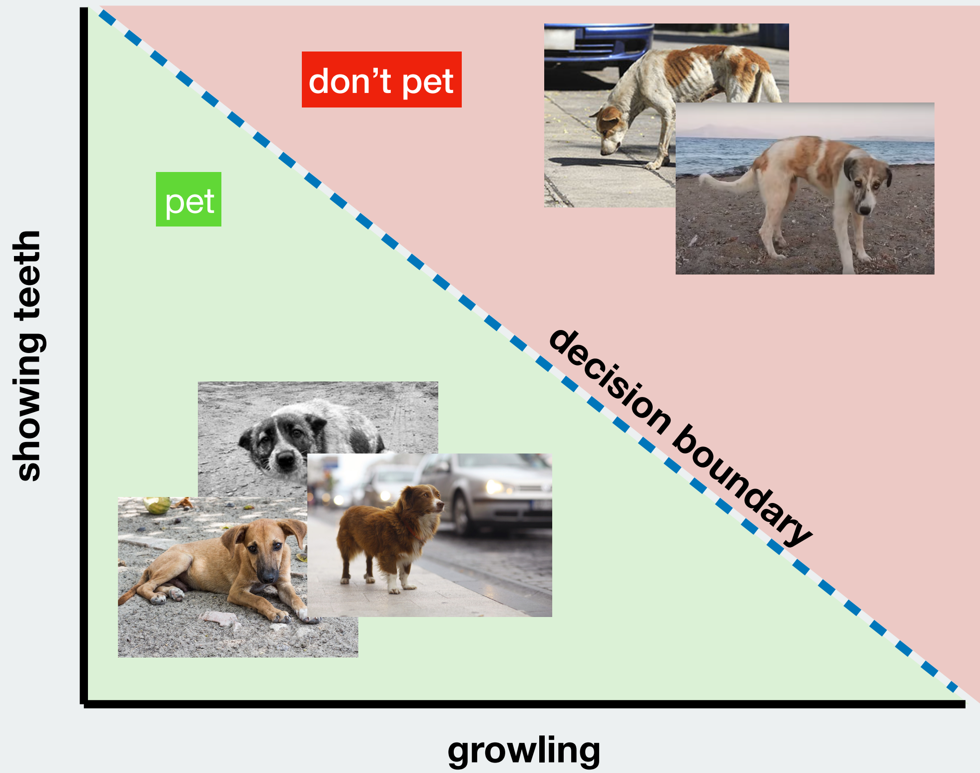
Outcomes

y

Canonical example



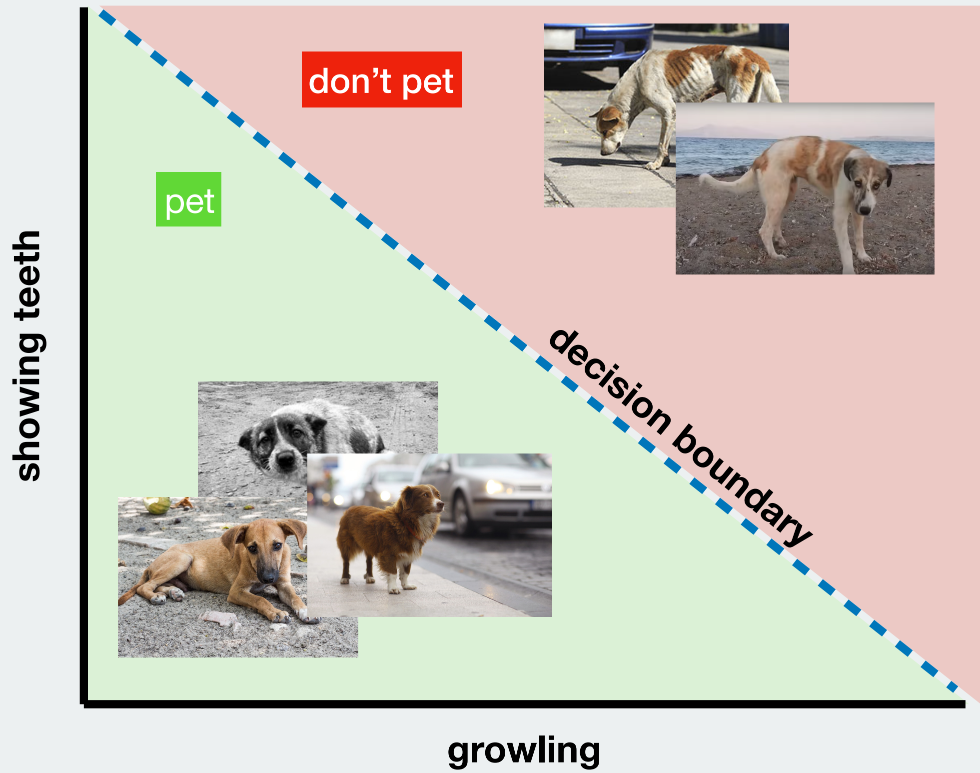
Canonical example



Canonical example

Supervised Machine Learning

When the input data has outcome labels



Types of machine learning

Types of machine learning

Supervised

When you have outcomes 👏

Unsupervised

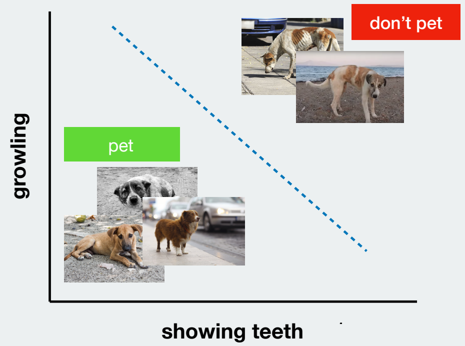
When you don’t have outcomes

Types of machine learning

Supervised

When you have outcomes 👏

- Classification



Unsupervised

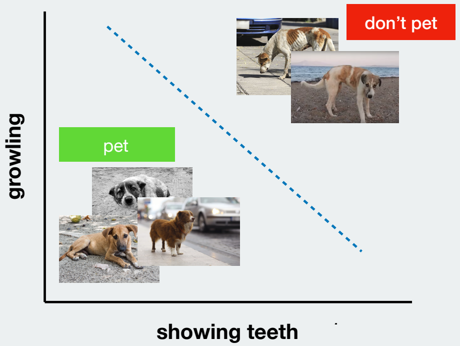
When you don't have outcomes

Types of machine learning

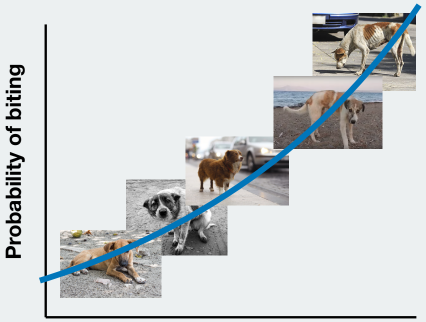
Supervised

When you have outcomes 👏

- Classification



- Regression



Unsupervised

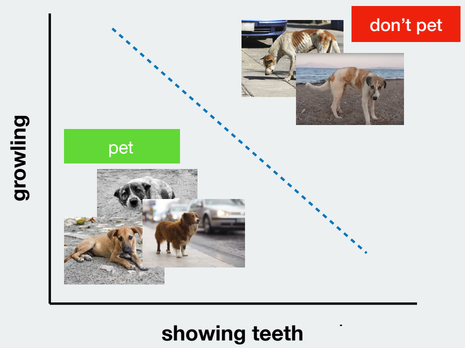
When you don't have outcomes

Types of machine learning

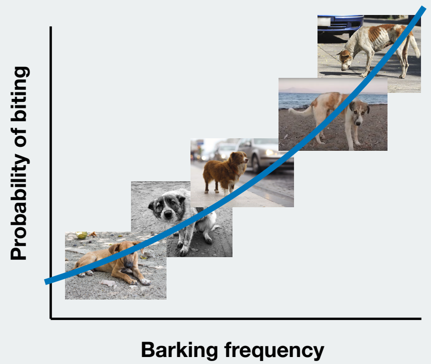
Supervised

When you have outcomes

- Classification



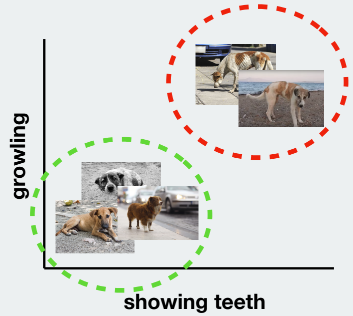
- Regression



Unsupervised

When you don't have outcomes

- Clustering

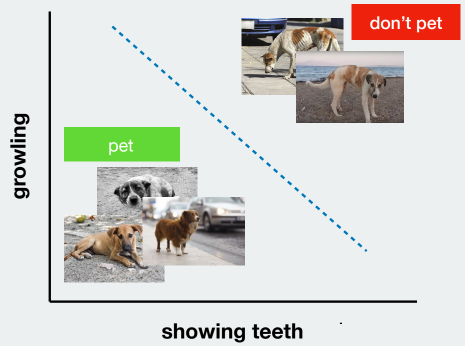


Types of machine learning

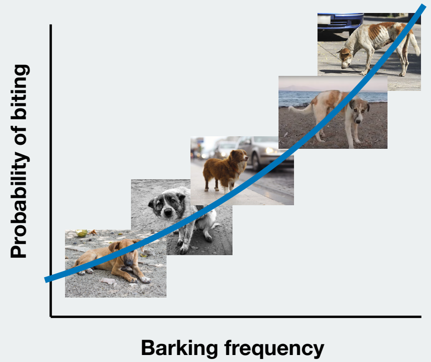
Supervised

When you have outcomes

- Classification



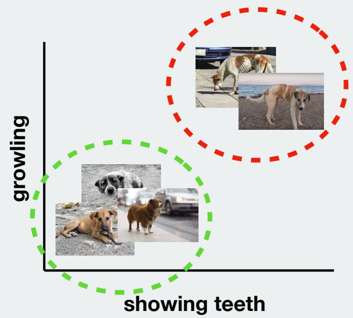
- Regression



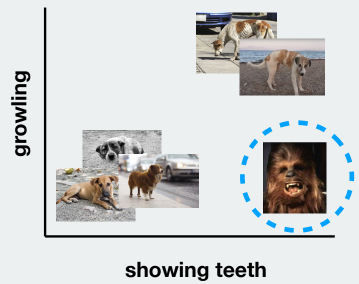
Unsupervised

When you don't have outcomes

- Clustering



- Outlier detection

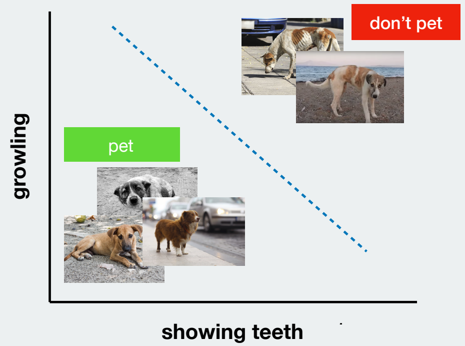


Types of machine learning

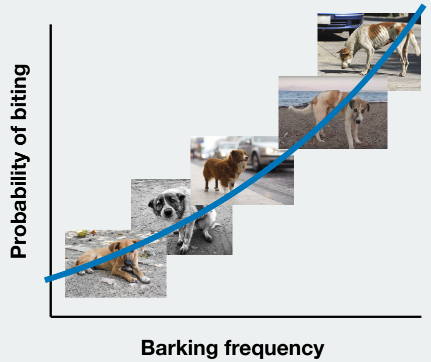
Supervised

When you have outcomes

- Classification



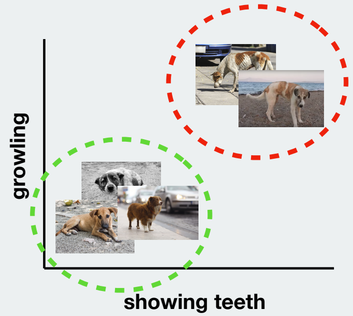
- Regression



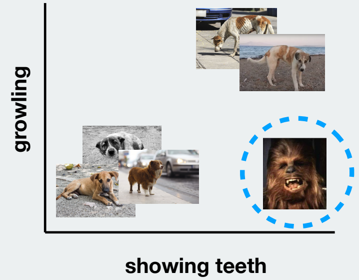
Unsupervised

When you don't have outcomes

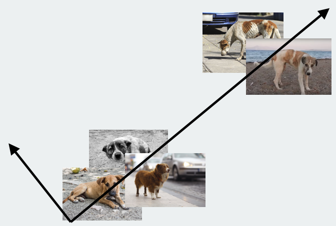
- Clustering



- Outlier detection



- Latent variable analysis

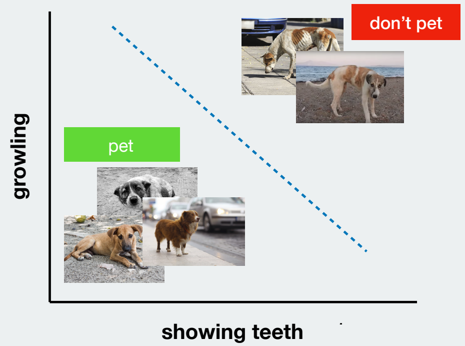


Types of machine learning

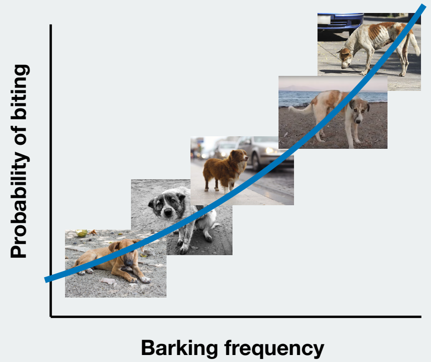
Supervised

When you have outcomes

- Classification



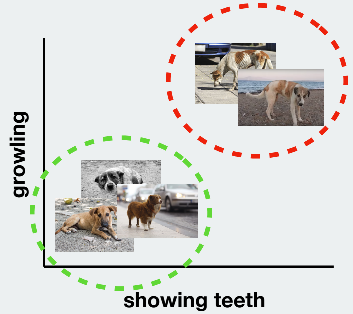
- Regression



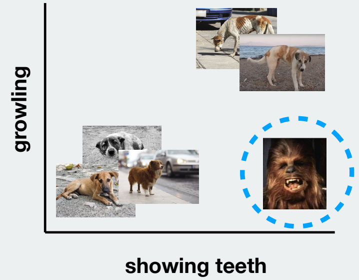
Unsupervised

When you don't have outcomes

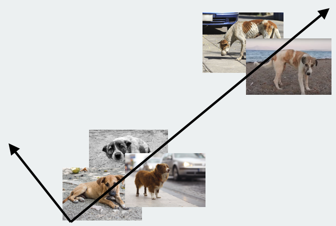
- Clustering



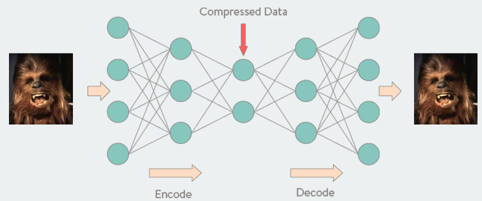
- Outlier detection



- Latent variable analysis

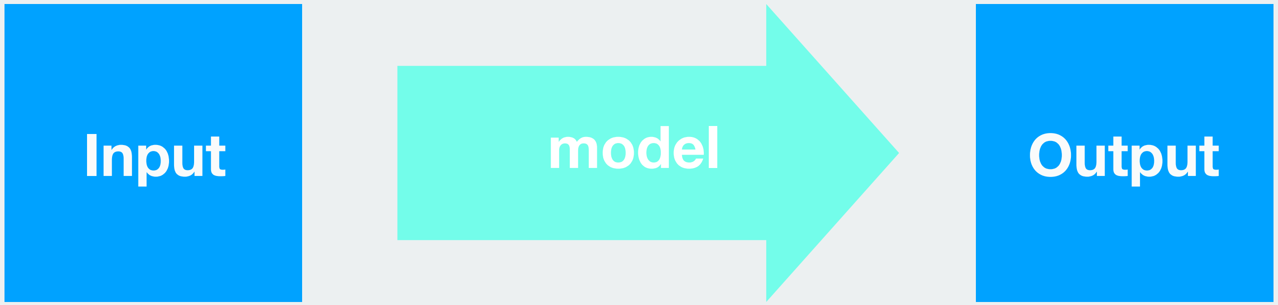


- Auto encoding



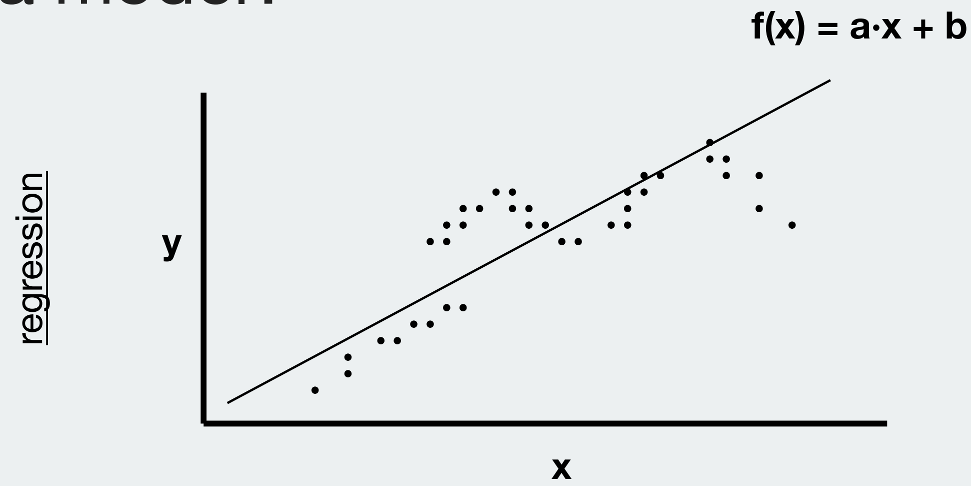
In a nutshell

You want the model that
best fits your data

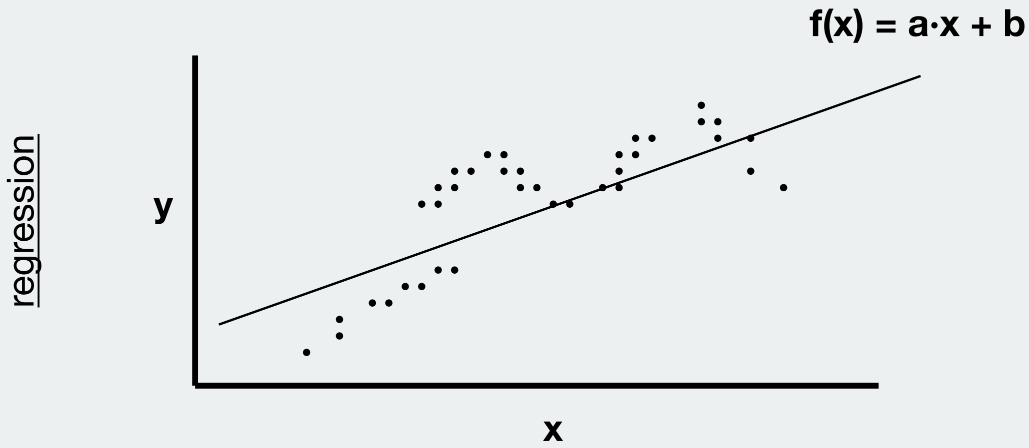


What is a model?

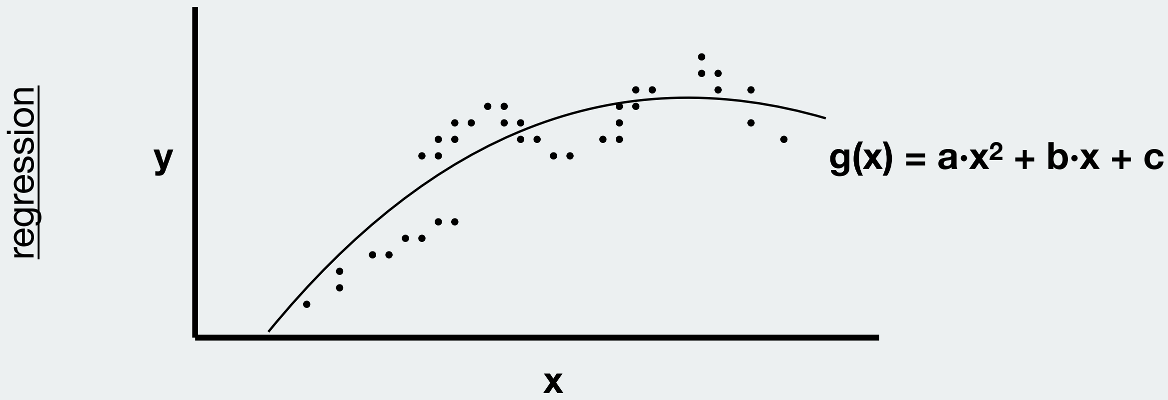
What is a model?



What is a model?

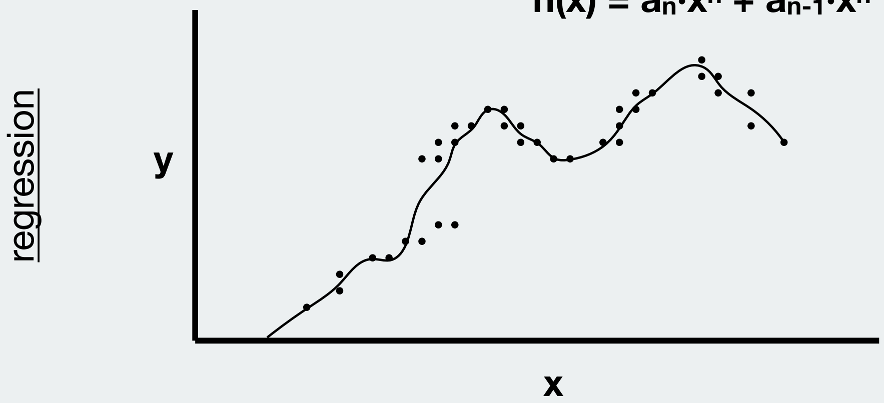


What is a model?



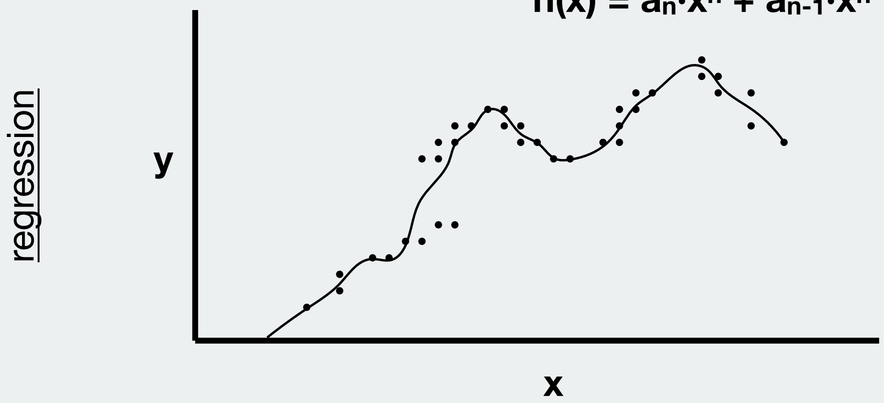
What is a model?

$$h(x) = a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \dots + a_0 \cdot x^0$$

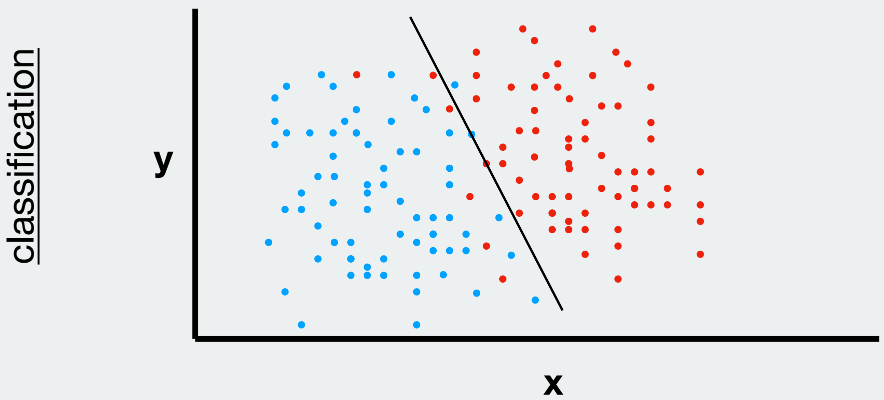


What is a model?

$$h(x) = a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \ldots + a_0 \cdot x^0$$



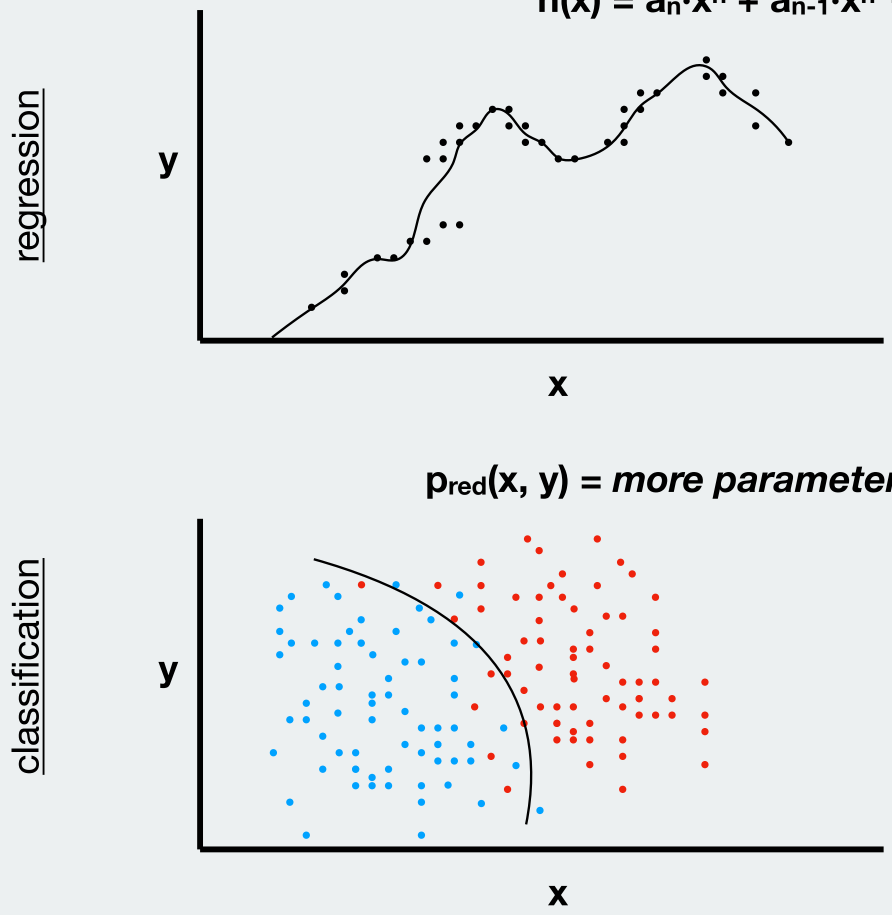
$p_{red}(x, y) =$ ***few parameters***



What is a model?

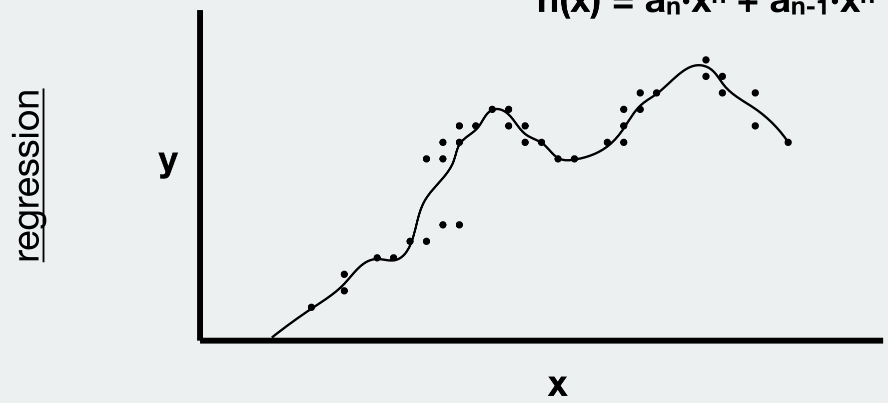
$$h(x) = a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \ldots + a_0 \cdot x^0$$

$p_{red}(x, y)$ = ***more parameters***

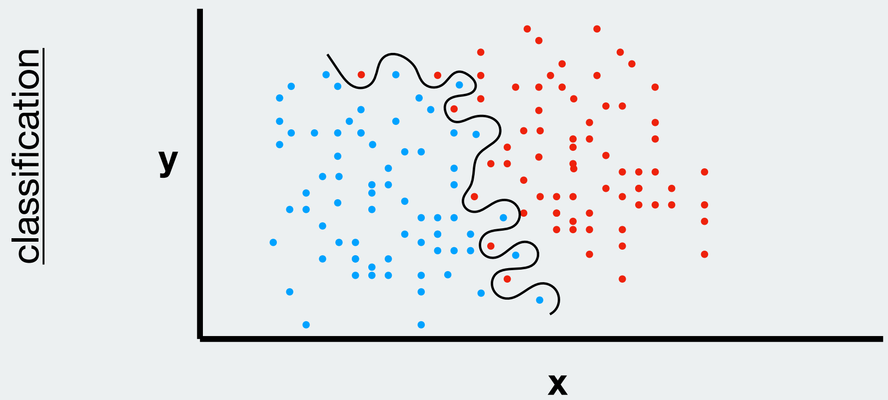


What is a model?

$$h(x) = a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \dots + a_0 \cdot x^0$$



$$p_{red}(x, y) = \textit{many parameters}$$



ML = Finding model parameters

Model evaluation

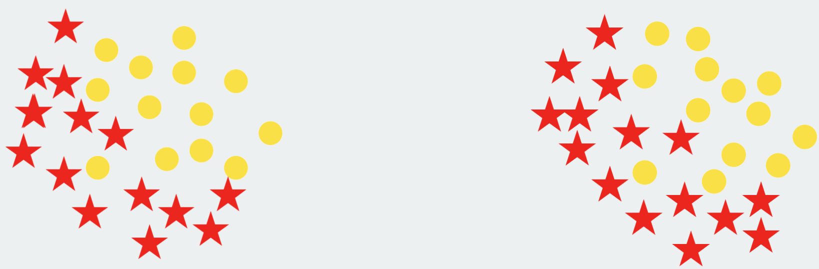
Model evaluation

Data

Model evaluation

Training data	Test data

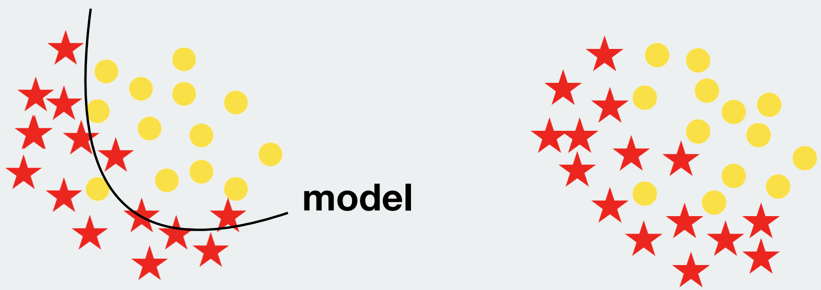
Data



Model evaluation

Training data	Test data

Data



training error: 3 / 27 = **0.11**

Model evaluation

Training data	Test data

Data

training error: 3 / 27 = **0.11**

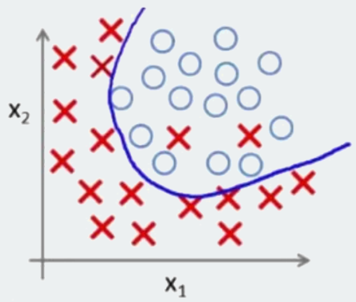
test error: 5 / 27 = **0.18**

Underfitting and overfitting

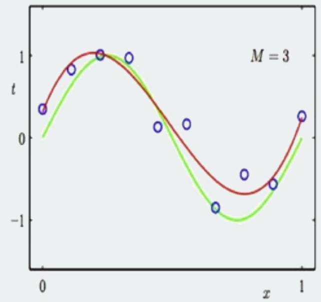
What is a model?

Underfitting and overfitting

Classification



Regression



Small train error

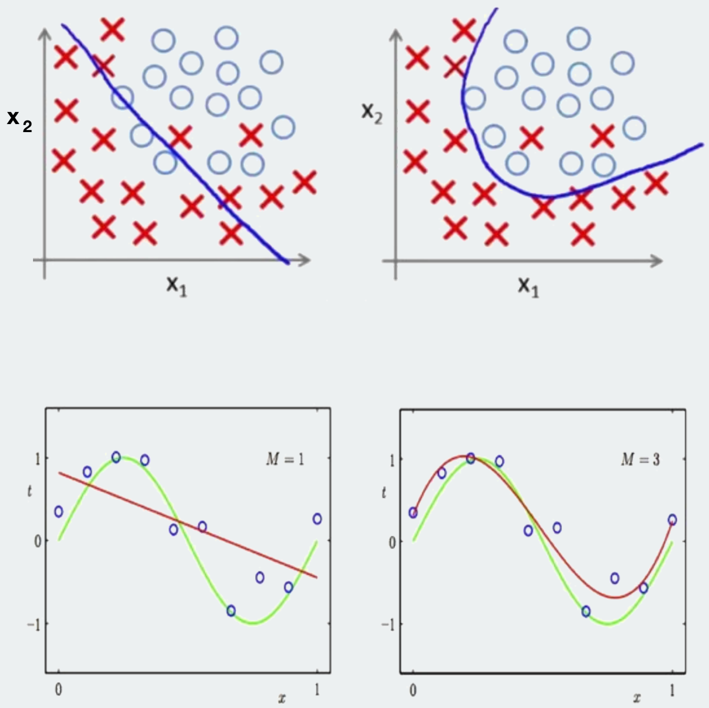
Small test error

Underfitting and overfitting

Underfitting

Classification

Regression



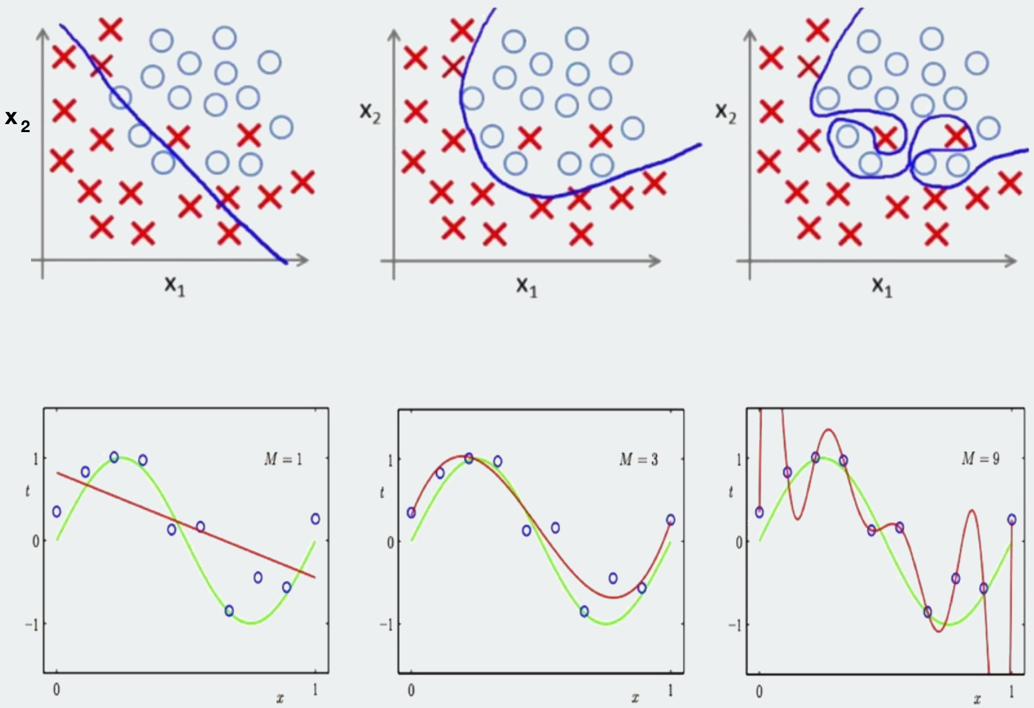
High train error	**Small train error**
High test error	**Small test error**

Underfitting and overfitting

Underfitting **Overfitting**

Classification

Regression



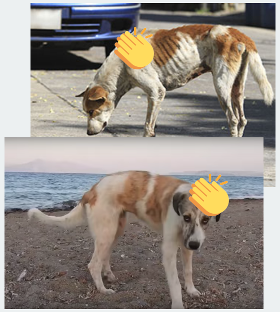
High train error | **Small train error** | **Zero train error**

High test error | **Small test error** | **High test error**

Model performance

Model performance

...



Many many experiments

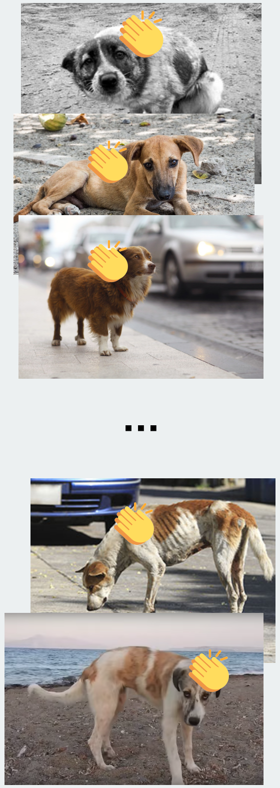
bites

0
0
0
0
0
0
0
0
0
⋮
1
1

Outcome

Number of points: 1.000.000
Total positive (bite): 20.000

Model performance



...

Many many experiments

bites

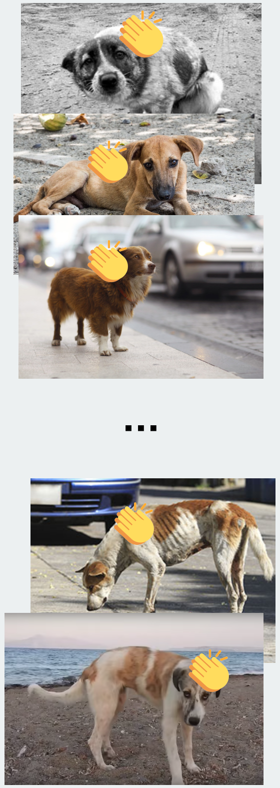
$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 1 \end{bmatrix}$$

Outcome

Number of points: 1.000.000
Total positive (bite): 20.000

What's the simplest model you can think of?

Model performance



...

Many many experiments

bites

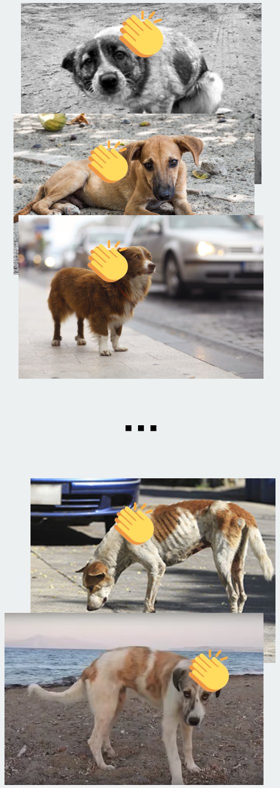
$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 1 \end{bmatrix}$$

Outcome

Number of points: 1.000.000
Total positive (bite): 20.000

baseline: always predict the dominant class
98% accuracy

Model performance



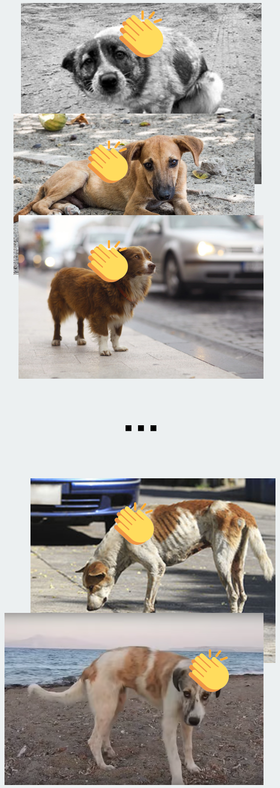
...

Many many experiments

$$\begin{array}{cc} \text{bites} & \text{bites} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \\ \text{Outcome} & \text{Prediction} \end{array}$$

Dumb model: *Only dogs named 'Frank' bite*

Model performance



...

Many many experiments

bites

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 1 \end{bmatrix}$$

Outcome

bites

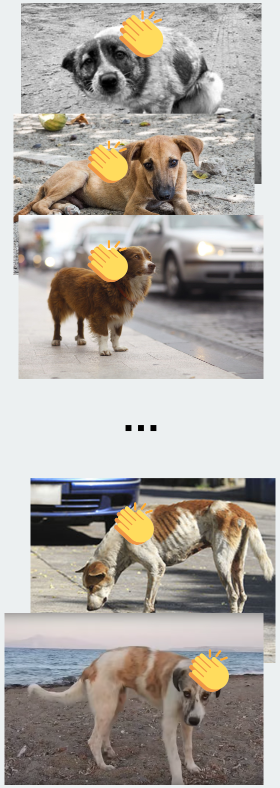
$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

Prediction

Dumb model: *Only dogs named 'Frank' bite*

Dogs named Frank:	1%
Dogs that bite:	2%
Frank & bites:	0.02%

Model performance



...

Many many experiments

$$\begin{array}{cc} \textbf{bites} & \textbf{bites} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \\ \textbf{Outcome} & \textbf{Prediction} \end{array}$$

Dumb model: *Only dogs named 'Frank' bite*

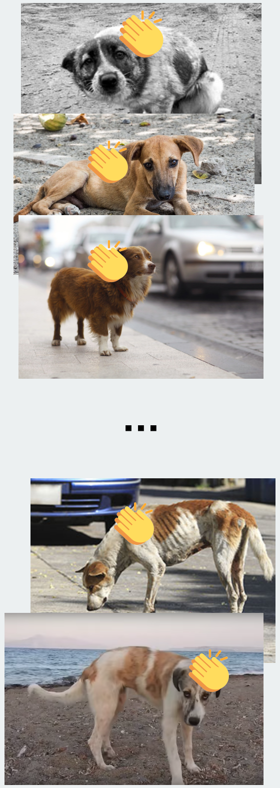
Dogs named Frank:	1%
Dogs that bite:	2%
Frank & bites:	0.02%

Accuracy:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{200 + 970200}{1000000} = 97\%$$

high!

Model performance



Many many experiments

$$\text{bites}\begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\0\\0\\\vdots\\1\\1\end{bmatrix}\ \text{Outcome} \qquad \text{bites}\begin{bmatrix}0\\0\\1\\0\\0\\0\\0\\0\\0\\\vdots\\0\\0\end{bmatrix}\ \text{Prediction}$$

Dumb model: *Only dogs named 'Frank' bite*

Dogs named Frank:	1%
Dogs that bite:	2%
Frank & bites:	0.02%

Accuracy:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{200 + 970200}{1000000} = 97\%$$

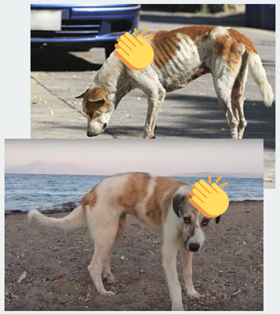
high!

Confusion matrix:

Prediction	Outcome: Bite	Outcome: No bite
Bite	TP: 200	FP: 9800
No bite	FN: 19800	TN: 970200

Model performance

...



Many many experiments

$$\underset{\textbf{Outcome}}{\overset{\textbf{bites}}{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 1 \end{bmatrix}}} \quad \underset{\textbf{Prediction}}{\overset{\textbf{bites}}{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}}}$$

Dumb model: *Only dogs named 'Frank' bite*

Dogs named Frank:	1%
Dogs that bite:	2%
Frank & bites:	0.02%

Accuracy:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{200 + 970200}{1000000} = 97\%$$

high!

Confusion matrix:

		Outcome	
		Bite	No bite
Prediction	Bite	TP: 200	FP: 9800
	No bite	FN: 19800	TN: 970200

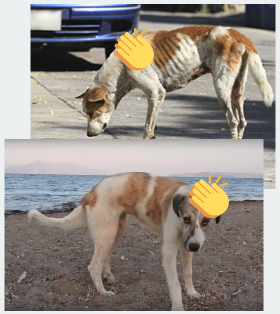
Precision & recall

$$precision = \frac{TP}{TP + FP} = \frac{200}{200 + 9800} = 2\%$$

$$recall = \frac{TP}{TP + FN} = \frac{200}{200 + 19800} = 1\%$$

Model performance

...



Many many experiments

$$\underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 1 \end{bmatrix}}_{\text{Outcome}}^{\text{bites}} \quad \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}}_{\text{Prediction}}^{\text{bites}}$$

Dumb model: *Only dogs named 'Frank' bite*

Dogs named Frank:	1%
Dogs that bite:	2%
Frank & bites:	0.02%

Accuracy:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{200 + 970200}{1000000} = 97\%$$

high!

Confusion matrix:

		Outcome: Bite	Outcome: No bite
Prediction	Bite	TP: 200	FP: 9800
	No bite	FN: 19800	TN: 970200

Precision & recall

$$precision = \frac{TP}{TP + FP} = \frac{200}{200 + 9800} = 2\%$$

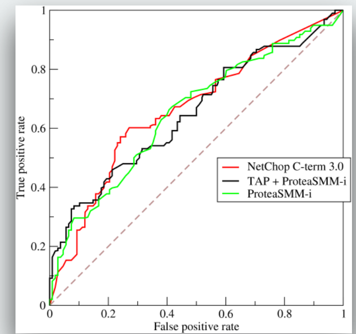
$$recall = \frac{TP}{TP + FN} = \frac{200}{200 + 19800} = 1\%$$

F1-score (harmonic mean of precision and recall):

$$F1 = \frac{2pr}{p + r} = \frac{2 \cdot 0.02 \cdot 0.01}{0.02 + 0.01} = .013$$

low!

Performance metrics summary

Metric	Description	Note	
Accuracy	$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$ "Fraction of classifications that are correct"	Bad when classes are not balanced. Should always be compared with a balance baseline.	Pretty easy to understand
Precision	$$precision = \frac{TP}{TP + FP}$$ "What fraction of the dogs accused of biting, would actually bite?"	Reveals a single useful aspect of a models performance	
Recall	$$recall = \frac{TP}{TP + FN}$$ "What fraction of dogs that would bite, were accused of biting?"	Reveals a single useful aspect of a models performance	
F1 score	$$F1 = \frac{2pr}{p + r}$$ "Harmonic average of precision and recall"	Mathematical construct = no intuitive interpretation	"Honest" measure of performance
Receiver operating characteristic (ROC)	 "Detection rate vs. false alarm rate for varying decision thresholds"	Visual performance metric that gives an good impression of performance. If area under curve is > 0.5, classifier is better than random.	