

# Introduction to Machine Learning



# Module 2: Machine Learning Deep Dive



# Course overview:

- ✓ Module 1: Introduction to Data Science & ML
- ✓ Module 1: Data Wrangling and Exploratory Data Analysis
- ☐ Module 3: Introduction to Machine Learning
- ☐ Module 4: NLP

Now let's turn to the data we will be using...



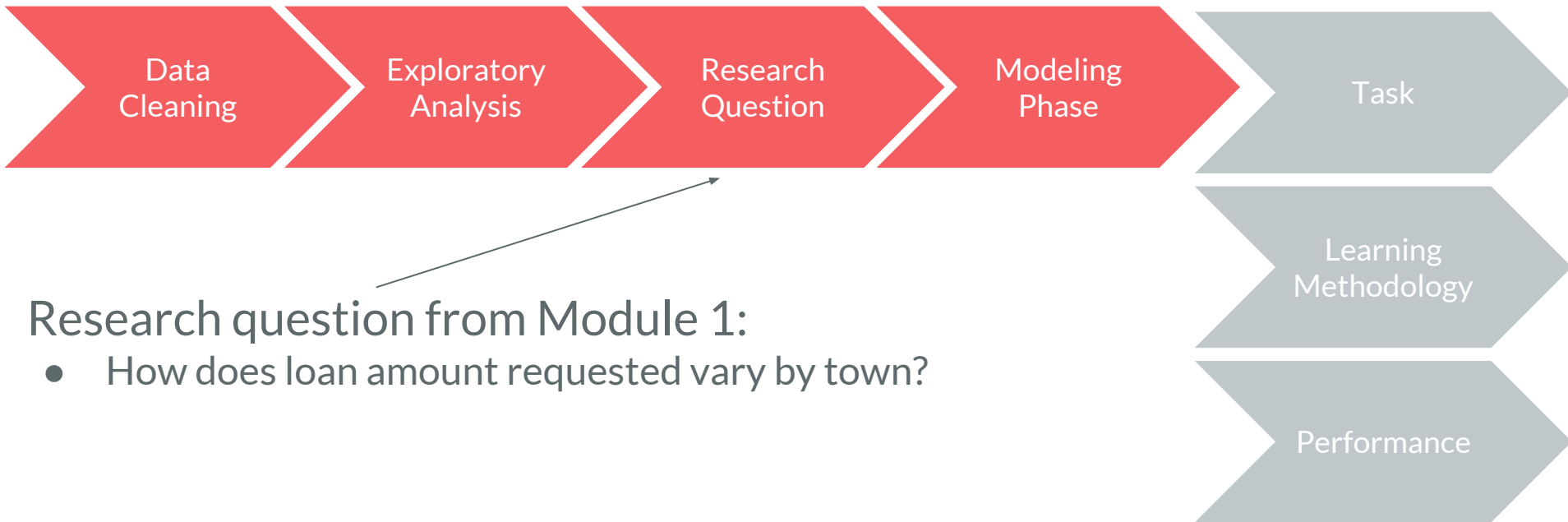
# Module Checklist

- ❑ Model Development
  - ❑ Defining the machine learning task
  - ❑ Supervised vs. unsupervised learning methods
  - ❑ Measuring performance of your model
- ❑ Model Validation



Modeling Phase

Now we have our research question, we are able to start modeling!



Now we have our research question, we are able to start modeling!

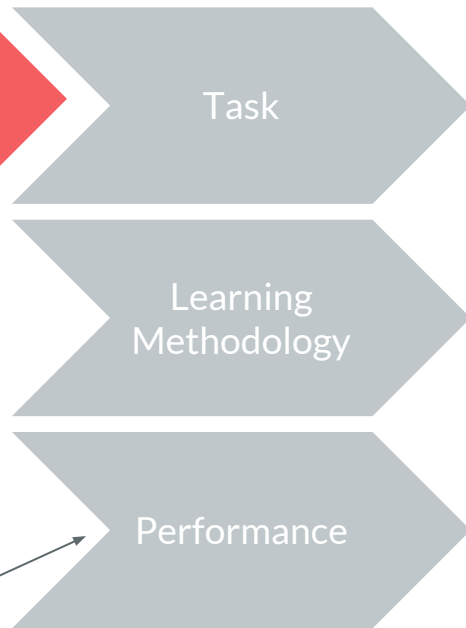


In this module we introduce the first two steps of modeling:

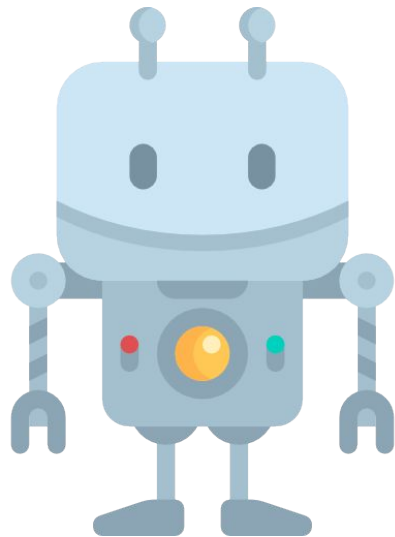
- Defining the machine learning task
- Understanding how the machine learns

We are here!

We will discuss model performance in the next module



Let's start at the basics. Why do we want to build a model?

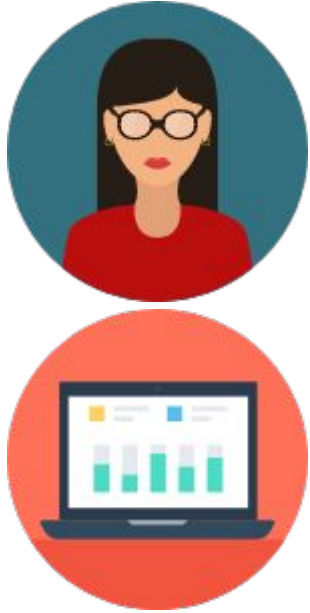


Machine learning allows us to tackle tasks that are too difficult to code all possible approaches to on our own.

**By allowing machines to learn from experience**, we avoid the need for humans to specify all the knowledge a computer needs.



## Human Intuition



Based on our experience of the world, we have an understanding of relationships between features

## Machine Learning Model



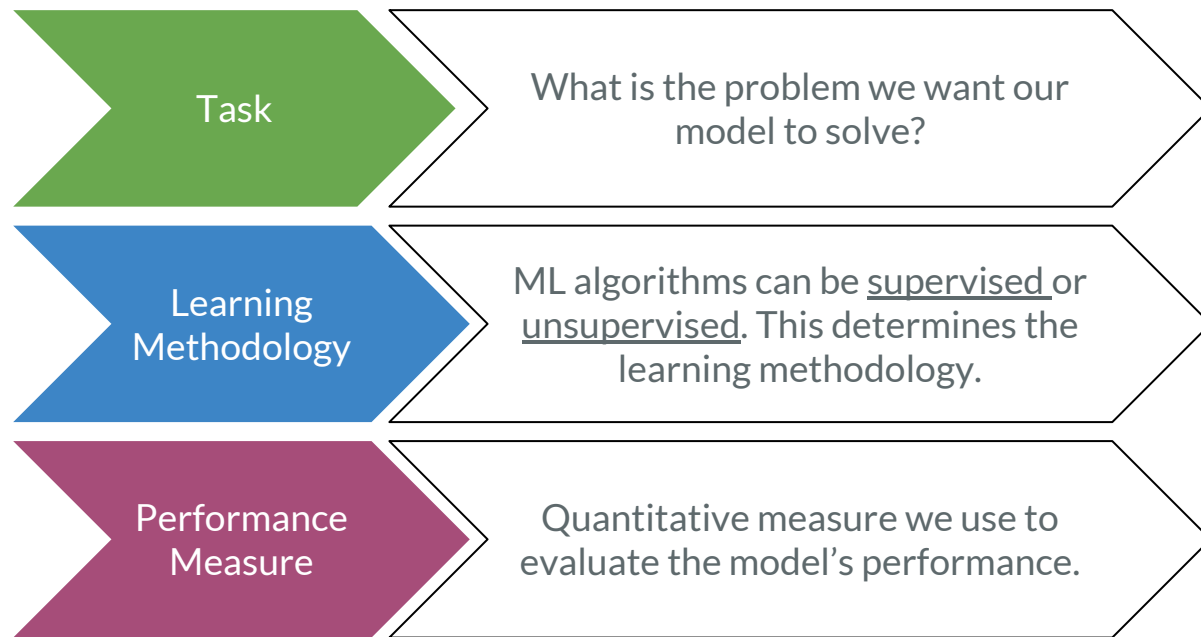
Computers acquire human intuition and quantify it, by extracting patterns from raw data

Machine learning models quantify and learn the patterns we observe in data.

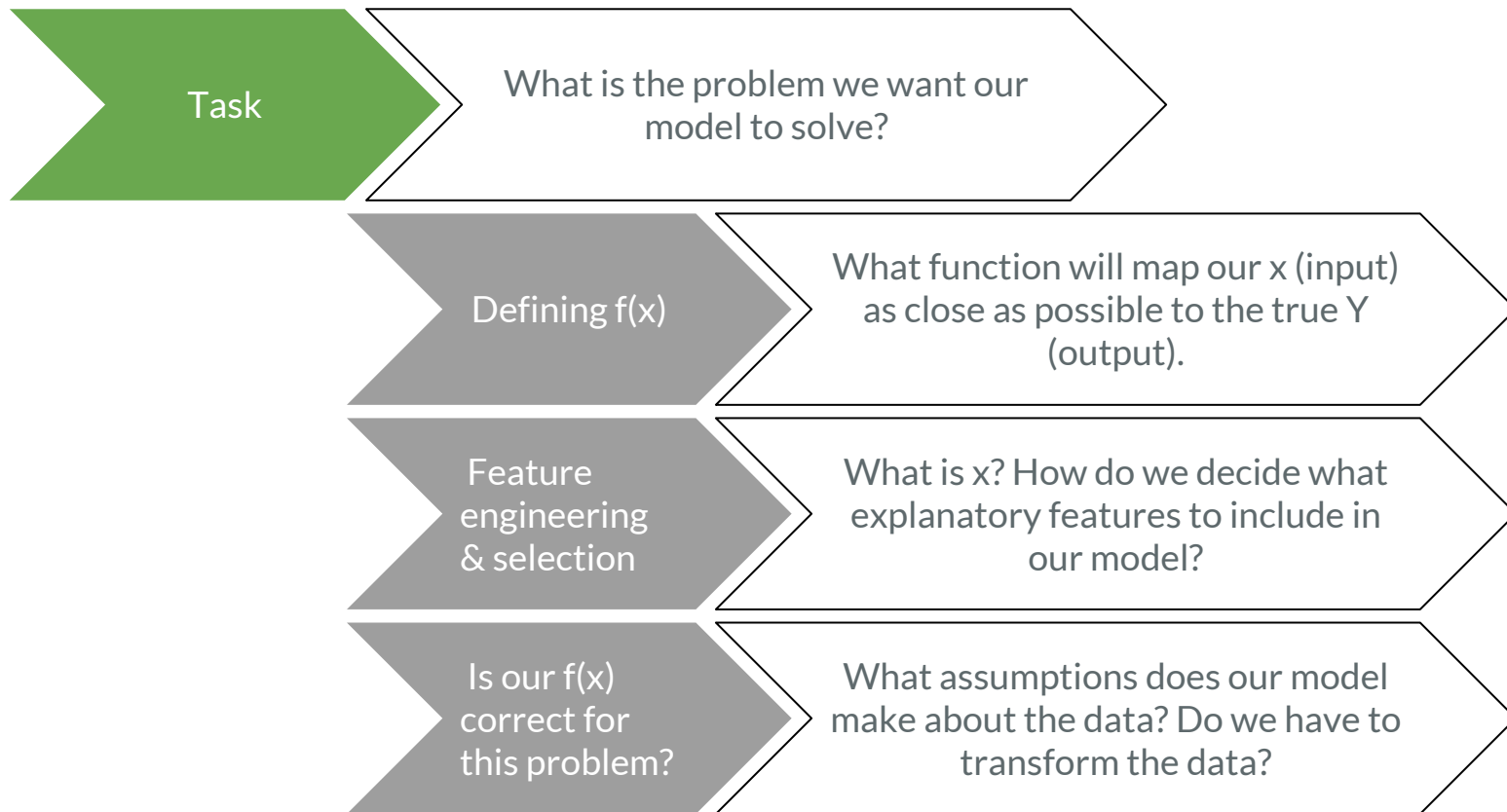


## Modeling Phase

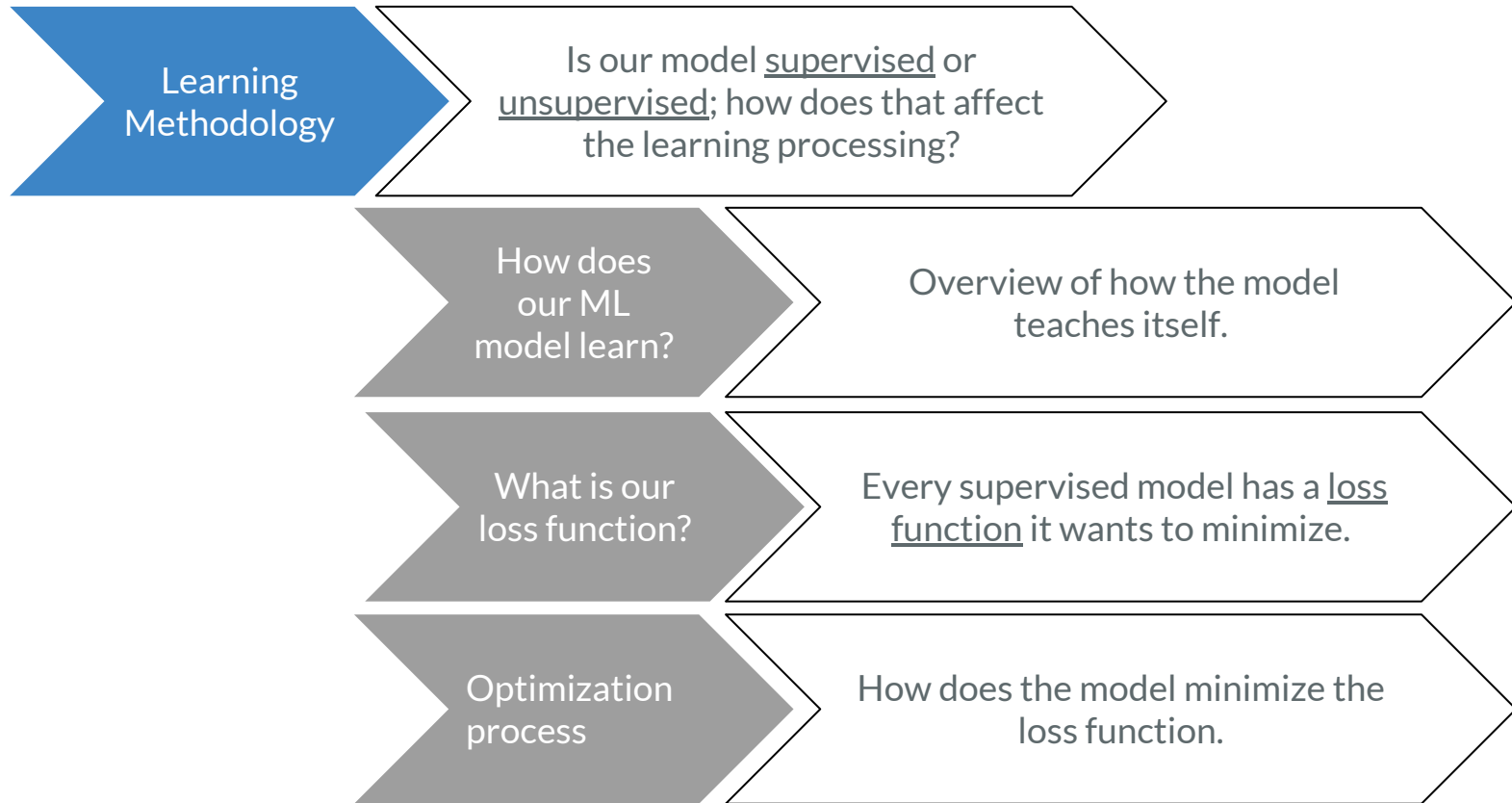
All models have 3 key components: a task, a learning methodology and a performance measure



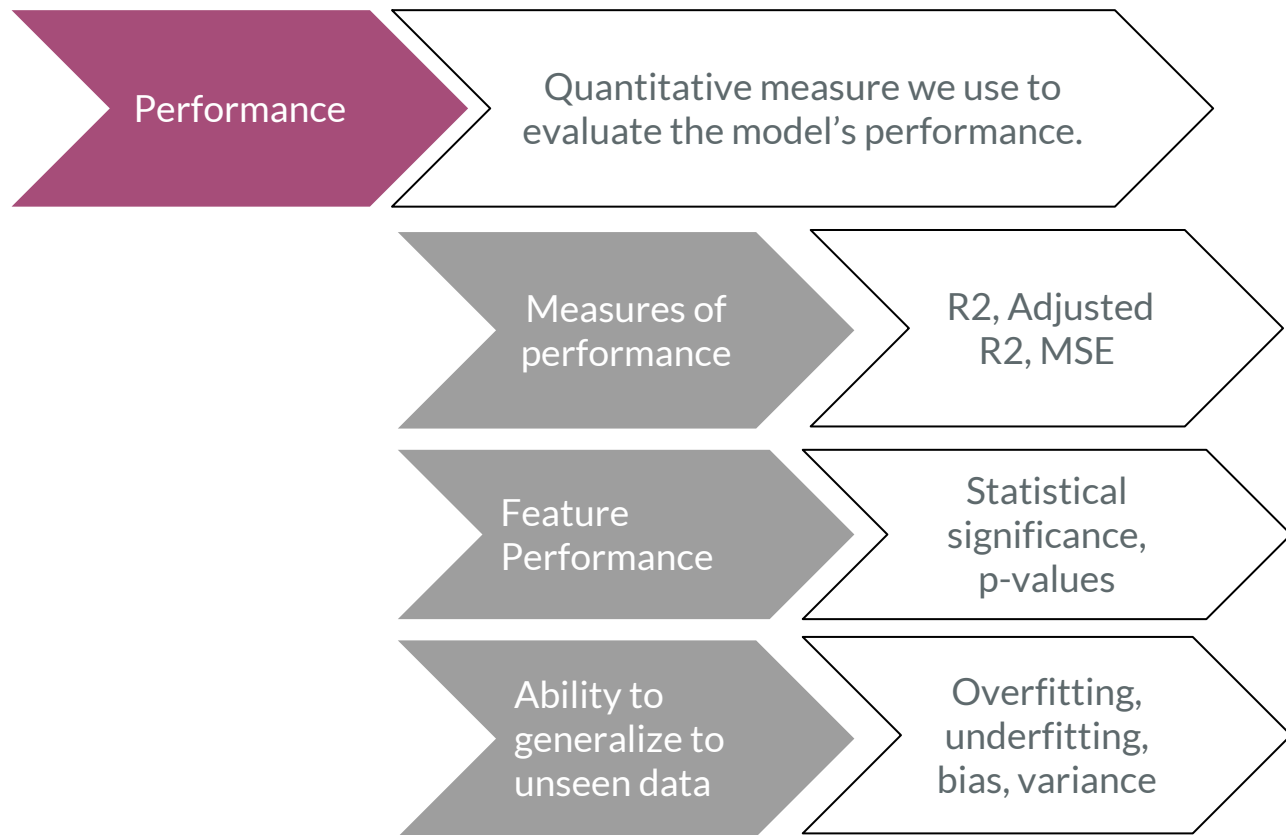
# Today we are looking closer at each component of the framework:



# Learning methodology: How does the model learn the function that best maps $x$ to the true $Y$ ?



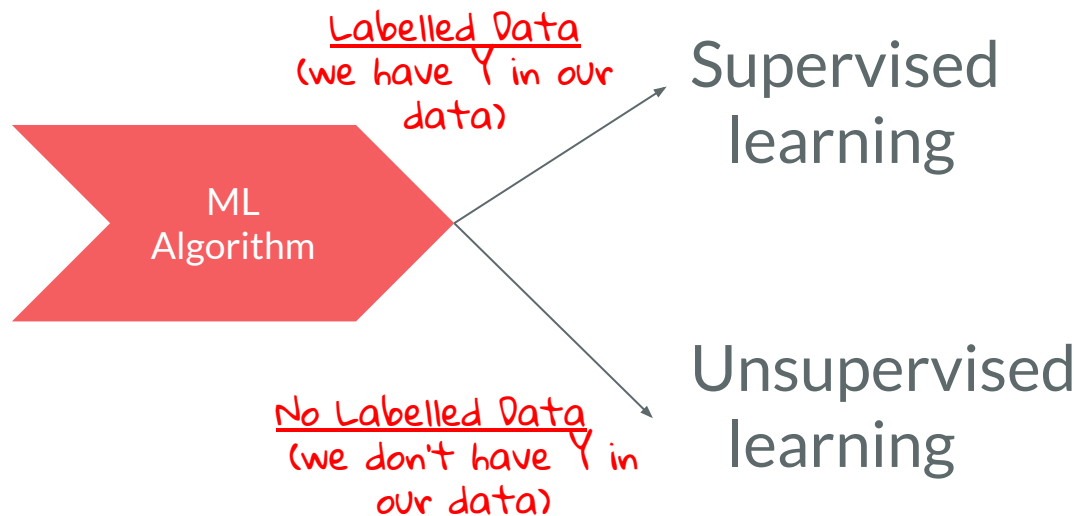
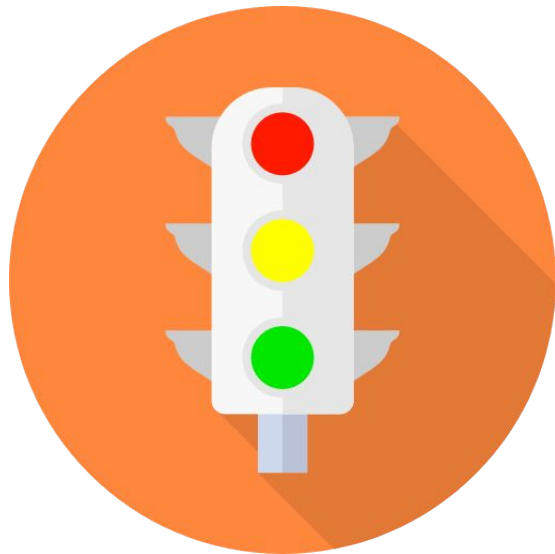
# Performance: How do we evaluate how useful the model is, and how we can improve it?



# Task and learning methodology

# Hold on! Important disclosure:

For the next few slides, we will be introducing the intuition behind machine learning models using **supervised learning** examples. Later in this module, we will explore how unsupervised learning is different.



# 1. Task



Task

What is the problem we want our  
model to solve?





Research  
Question

Recap: How does loan amount requested on Kiva vary by town in Kenya?

We have KIVA data about loan amount requested by borrowers all over Kenya.

We want to know how the loan amount requested varies by town.



Task

Building a model involves turning your research question into a machine learning question.

Research question

Machine learning task

How does loan amount requested on Kiva vary by town?

??



## Task

Firstly, let's establish a common vocabulary to talk about the data.

## Features

## Observations

	lender_count	loan_amount	location.country	location.country_code	location.geo.level	location.geo.pairs	location.geo.type	location.town
7	225	Kenya	KE	town	-1.166667 36.833333	point	Kiambu	
14	350	Kenya	KE	town	0.516667 35.283333	point	Eldoret	
33	1075	Kenya	KE	town	1 38	point	Kakamega North	

Location.town is an example of a feature. Every column in our data set is a feature!

Every row of our dataset is an observation. When we include the observation in our model it is part of our training set.

Task

A machine learning task has explanatory features and an outcome feature.

Explanatory features



Town borrower  
lives in

Outcome feature



Loan amount  
requested

An outcome feature is the **feature we expect to change** when the explanatory features are manipulated. In this example, we expect the loan amount to change when we change the location.

What would the outcome features and explanatory features be in the research questions below?

*Try identifying some:*

- What will the price of a stock be tomorrow?
- Does this patient have malaria?
- Would this person buy a car?

## Solutions:

The outcome feature might be regression (e.g. \$12) or a classification (e.g. Yes or No). We'll talk about this more later!



Explanatory features	Outcome feature
Price of a stock market index today	Company X's stock price tomorrow
Age, symptoms, travel history	Whether or not a patient has malaria
Income, location	Whether or not a person would buy a car

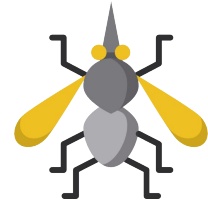
## Task

Let's define our explanatory and outcome features for this task

### ***Problem:***

I am the mayor of a 30,000 person town and need to justify spending budget on mosquito nets.

I want evidence on how the number of mosquito nets affects the number of cases of malaria. *Can you help?*

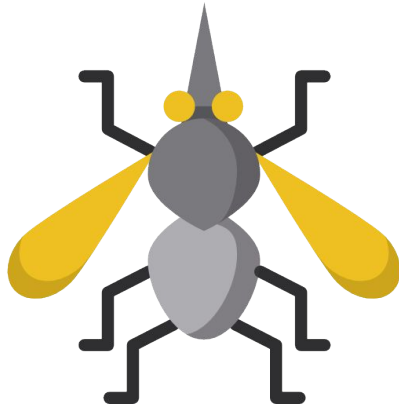


1. Research  
Question

2. Task

Let's start by identifying the research question!

The research question is what we want to find out from the data, formally stated.



How does the number of cases of malaria change when the number of mosquito nets changes?



Task

Next let's define our task!

1

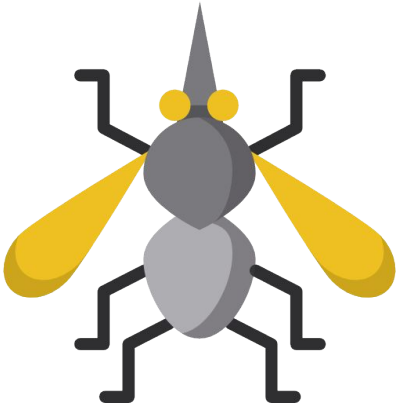
Define explanatory  
and outcome  
feature

2

Define  $f(x)$

3

Bring it all  
together



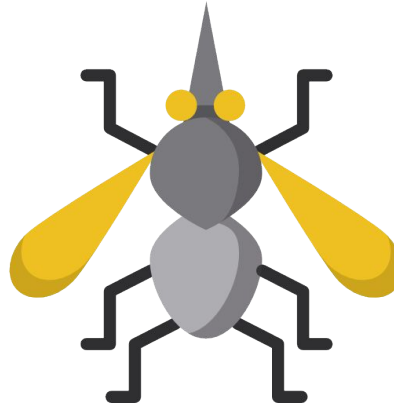
Define explanatory  
and outcome  
feature

How does the number of cases of malaria  
change when the number of mosquito nets  
changes?

### explanatory feature(s)

**X** Number of  
mosquito nets

2007: 1000  
2008: 2200  
2009: 6600  
2010: 12600



### outcome feature

**Y** Number of people  
with malaria

2007: 80  
2008: 40  
2009: 42  
2010: 35

We also call our explanatory features **X**, and our outcome feature **Y**.  
Looks like as mosquito nets increase, the number of malaria cases decreases.





## Task

What would you conclude from looking at this data? How many nets would you recommend?

X Number of  
mosquito nets

2007: 1000  
2008: 2200  
2009: 6600  
2010: 12600

Y Number of people  
with malaria

2007: 80  
2008: 40  
2009: 42  
2010: 35

You came to a conclusion by **recognizing a pattern in the data**. This is similar to how a machine learning algorithm would approach the same problem.



Task

Machine learning allows us to learn from history

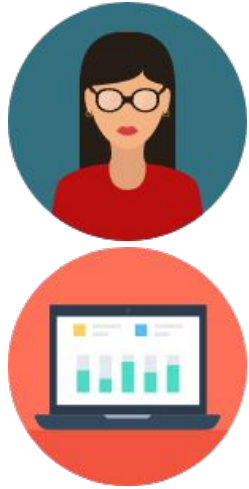
If Mr. Mayor had **no** machine learning methods to use, he could find an answer by trying a different # of nets year after year.

But this has an obvious **human cost**, and it would be very hard to update the model to account for, for example, new residents to his town.

Machine learning algorithms help answer questions without this human cost - we are **learning from data**, or in other words, **learning from history!**



## Human Intuition



“Over four years,  
increasing number of  
mosquito nets  
decrease the number of  
malaria cases.”



## Machine Learning Model

An increase in  $x$   
(mosquito nets) causes a  
decrease in  $Y$  (malaria  
cases).

- Humans form rules based upon observation and pattern recognition.
- ML model takes input  $x$  and maps it to the output  $Y$ .

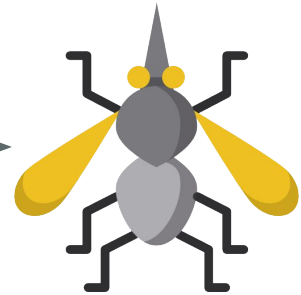
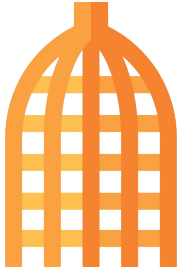
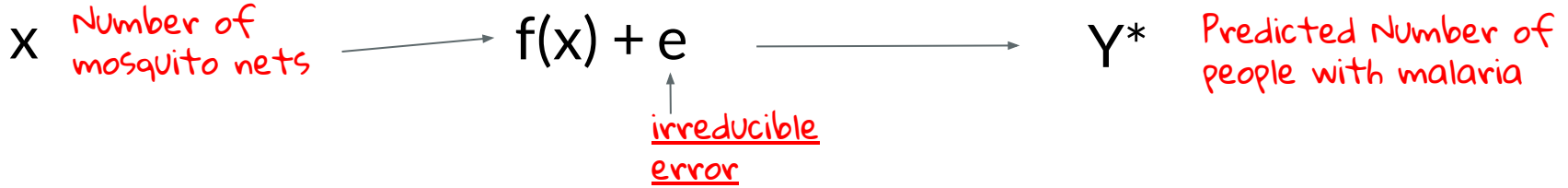
Define  $f(x)$

Our model  $f(x)$  is a function that maps our input  $x$  to a predicted  $Y^*$ .

explanatory feature(s)

model

predicted outcome



Define  $f(x)$

The goal of  $f(x)$  is to predict a  $Y^*$  as close to the true  $Y$  as possible.

My job is to make the predictions as **useful** as possible!



Our function  $f(x)$  maps an input  $x$  to a **predicted  $Y$** , which we refer to as  **$Y^*$** . We want to choose an  $f(x)$  that will map  $x$  as close to the **true  $Y$**  as possible.

$$f(x) + e = Y^*$$

**Predicted Number of people with malaria**

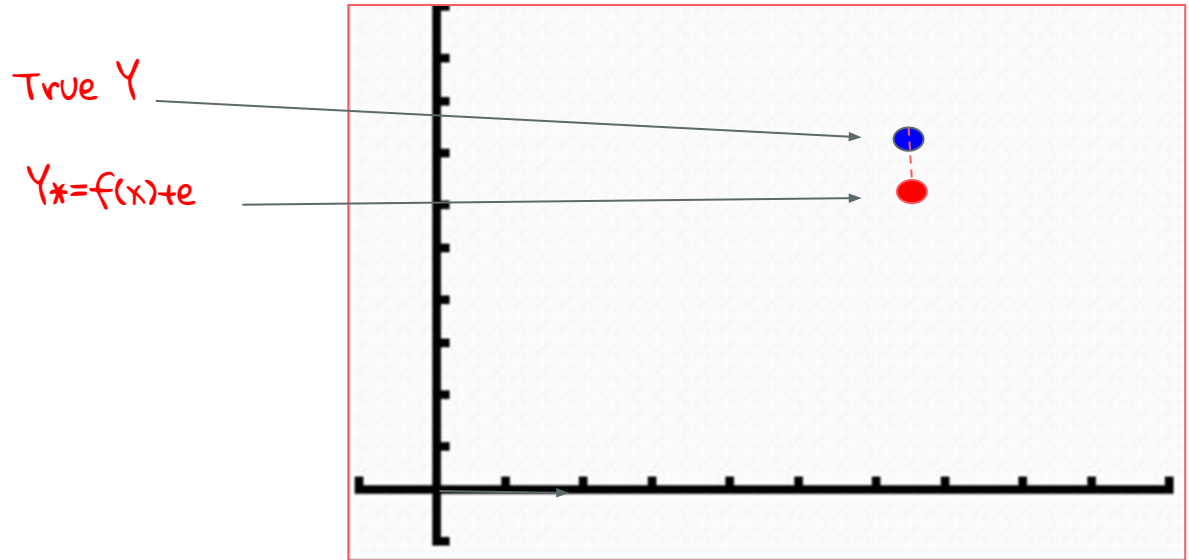
$e$  is **irreducible error**. This is the term that, no matter how good your model is, will never be reduced to 0.



Define  $f(x)$

We want  $Y_*$  to be close to true  $Y$  because we want the function to output useful predictions.

In this example, predicted  $Y$  appears close to the true  $Y$ . We will talk about how to quantify this in the next section.



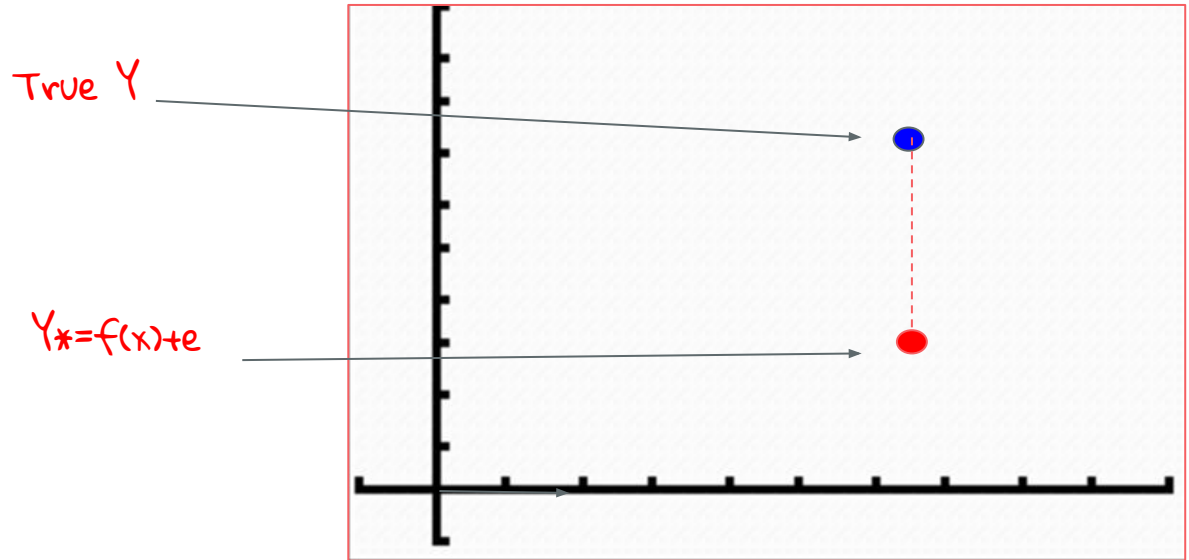


Define  $f(x)$

We want  $Y_*$  to be close to true  $Y$  because we want the function to output useful predictions.

In this example, predicted  $Y$  appears far from the true.

***This is probably not very useful.*** We will talk about how to quantify this in the next section.



Define  $f(x)$

What is  $f(x)$ ? It depends on the machine learning model or algorithm we choose.

$x$



$f(x)$



$Y^*$

explanatory  
feature(s), like  
number of  
mosquito nets

predicted outcome,  
e.g. Number of  
people with malaria

Examples of  $f(x)$ :

Supervised learning algorithms:

- Linear regression
- Decision tree
- Random forest
- ...

When you have labelled data



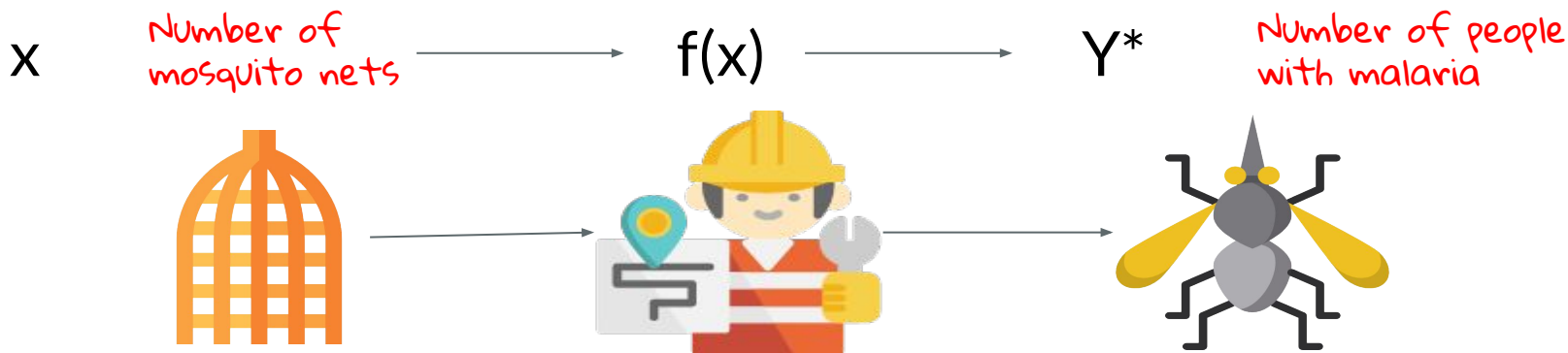
Bring it all  
together

Let's bring everything together.

## Research question

How does the number of cases of malaria change when the number of mosquito nets changes?

## Machine learning task



## Supervised learning task

The task function depends upon the data type you want to predict. Supervised learning problems fall into two main categories: regression & classification.



### The Task

#### Regression

Continuous variable

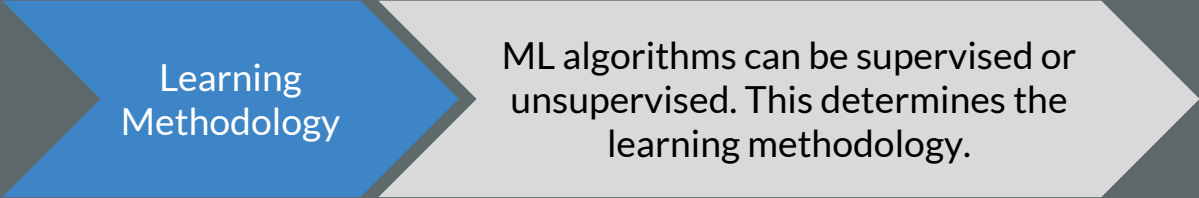
#### Classification

Categorical variable

A regression problem is when we are trying to **predict a numerical value**, such as “dollars” or “weight”.

A classification problem is when we are trying to **predict whether something belongs to a category**, such as “red” or “blue” or “disease” and “no disease”.

## 2. Learning Methodology



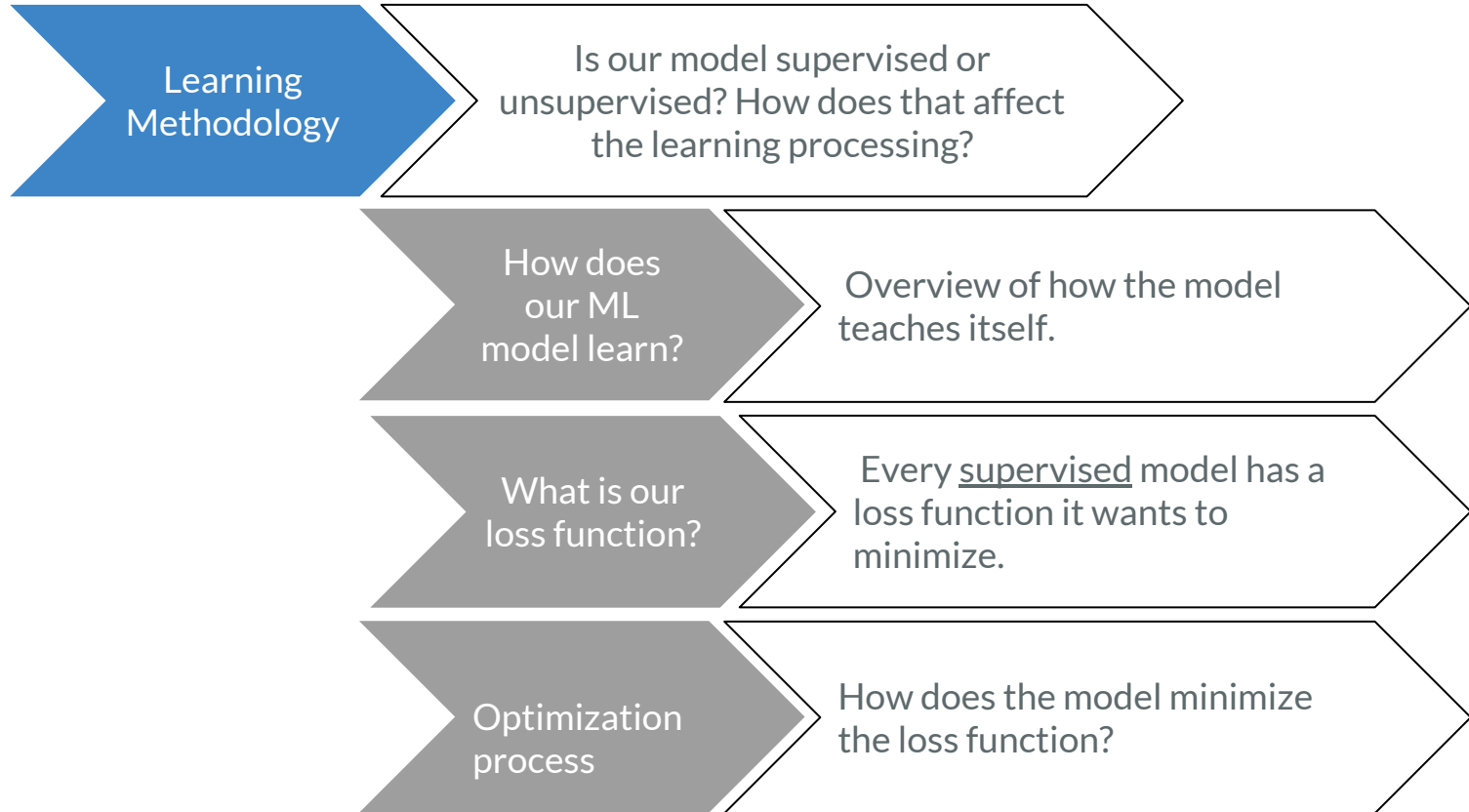
Learning  
Methodology

The diagram consists of two chevron-shaped boxes pointing to the right. The first box is blue and contains the text 'Learning Methodology'. The second box is light gray and contains the text 'ML algorithms can be supervised or unsupervised. This determines the learning methodology.'.

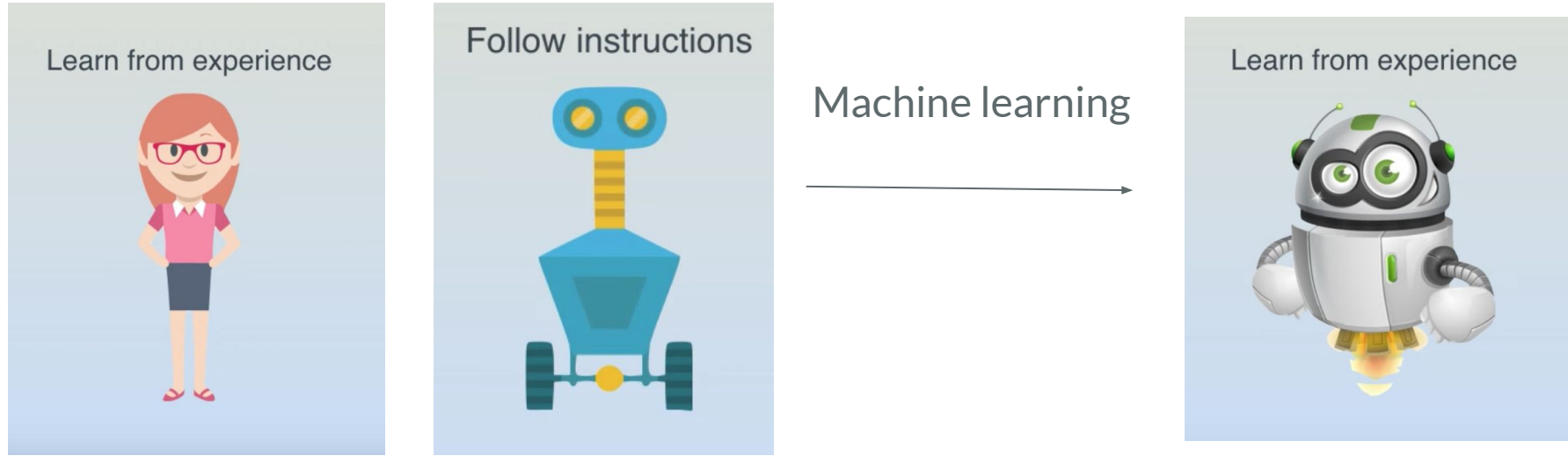
ML algorithms can be supervised or  
unsupervised. This determines the  
learning methodology.



Learning methodology: how does the model learn the function that best maps  $x$  to the true  $Y$ ?



Recall that machine learning is a subset of data science that allows machines to learn from raw data.



Traditional software programming involves giving machines instructions which they perform. Machine learning involves allowing machines to learn from raw data so that the computer program can change when exposed to new data (learning from experience).

Source: <https://www.youtube.com/watch?v=IpGxLWOIZy4>

What do we mean when we say a machine  
"learns from experience"?

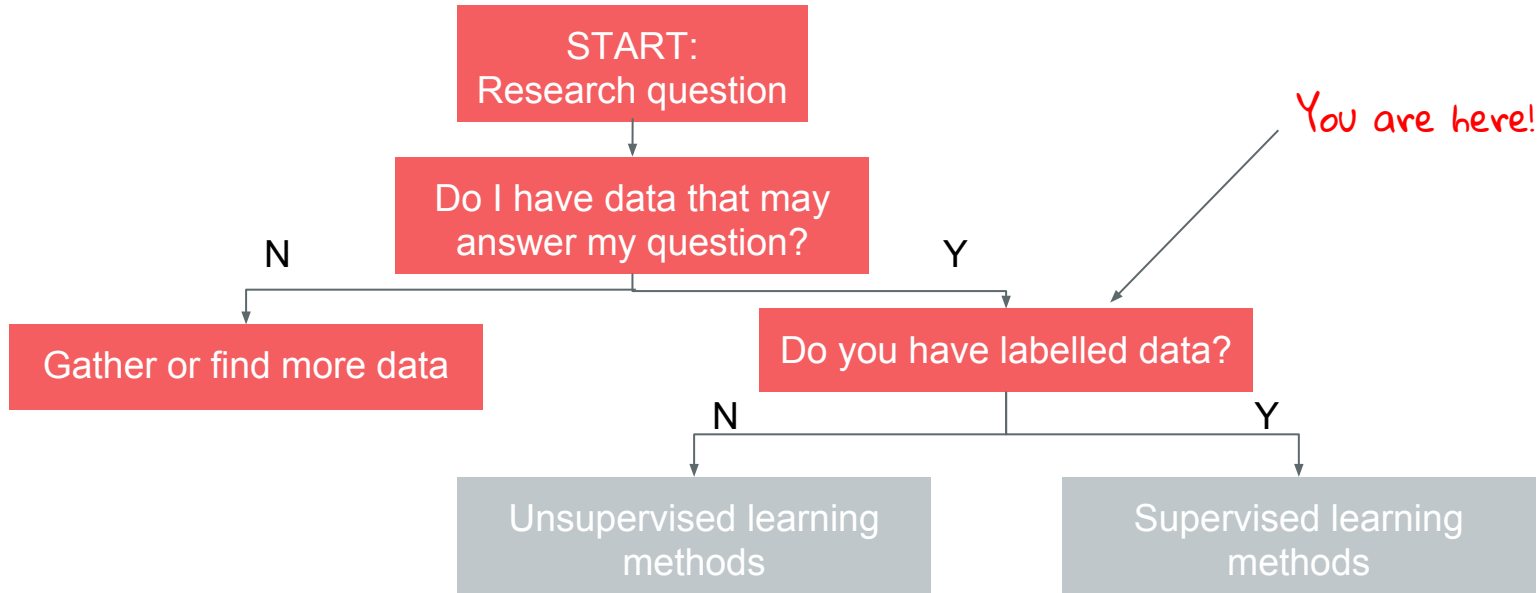
Machine learning is a subset of  
data science that allows machines  
to learn from raw data.



How does Mr.  
Model learn from  
raw data?



How the algorithm learns depends upon type of data you have.



# What does labelled data mean?

Do you have labelled data?

Yes

The outcome feature ( $Y$ )  
you are interested in  
predicting is recorded in  
the data. If you have a  
labelled  $Y$ , you can use  
supervised learning  
methods.

$Y$ =Number of people  
with malaria

2007: 80

2008: 40

2009: 42

2010: 35

No

The outcome feature ( $Y$ )  
is not recorded in the  
data. You do not have a  
labelled  $Y$ .

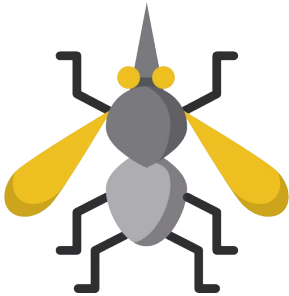
$Y$ =Number of people  
with malaria

2007:

2008:

2009:

2010:



Whether or not you have labelled data determines whether it is a Supervised or Unsupervised Learning problem

$$f(x) + e = Y^*$$

## Supervised Learning

- For every  $x$ , there is a  $Y$
- Goal is to **predict**  $Y$  using  $x$

## Unsupervised Learning

- For every  $x$ , there is no  $Y$
- Goal is not to predict, but to **investigate**  $x$



Do you have labelled data?

Yes

No

$Y$  is in your data

$Y$  is not in your data

Most problems you will initially encounter are supervised algorithms.

*How do supervised algorithms learn?*

## supervised learning

# Intuitive explanation for how supervised algorithms learn:

Y Number of people  
with malaria

2007: 80

2008: 40

2009: 42

2010: 35



Imagine you are a teacher and you ask your students a question.

The labels Y provide the correct answer for the problem the students are trying to solve. **Since you know the correct answer, you can reward good student performance and punish poor performance.** This encourages ongoing learning!



## supervised learning

Extending this example, you (the researcher) are the teacher and Mr. Model is the student.



Mr. Model

I want to get the correct answer for predicting  $Y$  and be the best student in the class.



Great Mr. Model!  
Once you give me your answer I will let you know the correct answer.

Every time Mr. Model predicts  $Y^*$ , you compare  $Y^*$  to the true  $Y$  to see how well he did.



## supervised learning

Mr. Model starts trying to provide an estimated  $Y^*$  by guessing.

I have never seen this problem before! I'll just start by randomly guessing an answer and see what happens.



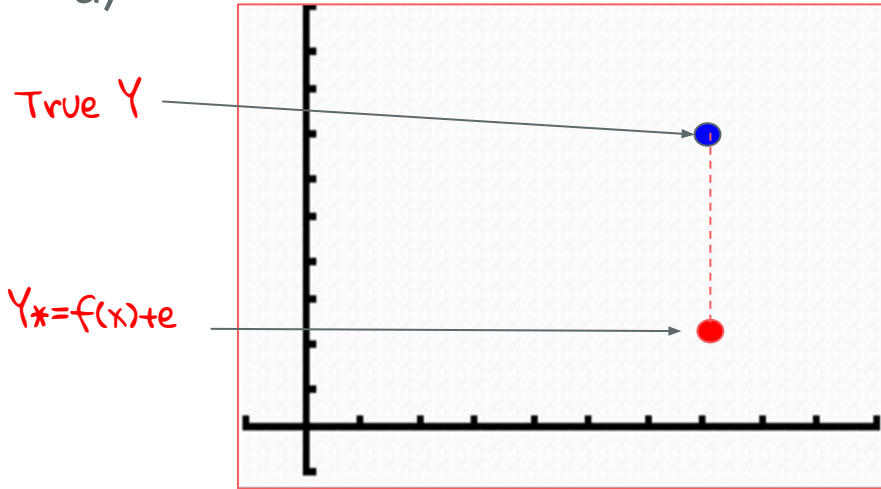
Mr. Model

$Y^*$	$Y$
Predicted Number of people with malaria	Actual Number of people with malaria
2007: 1	2007: 80
2008: 2000	2008: 40
2009: 300	2009: 42
2010: 40	2010: 35

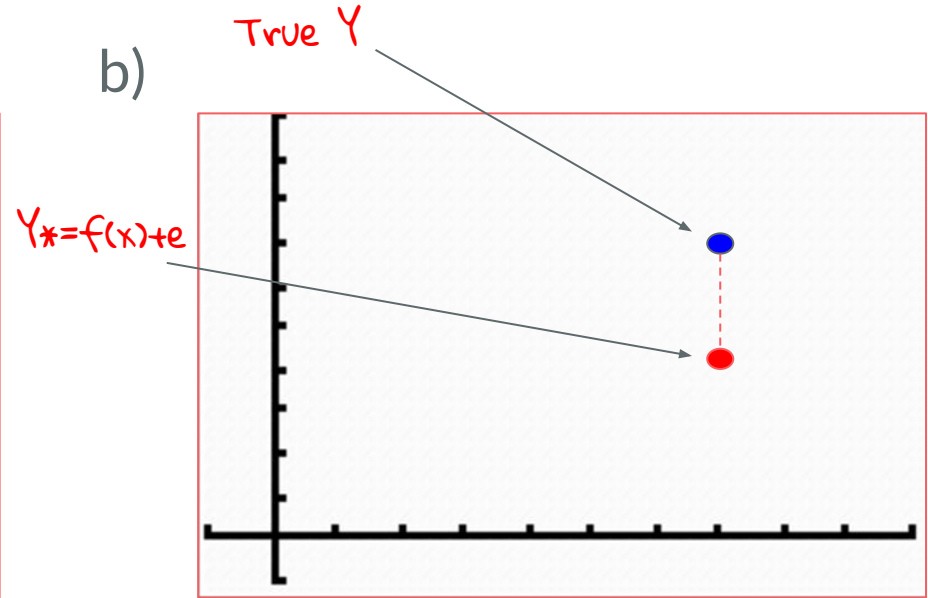
Unsurprisingly, the results appear terrible, judging from the fact that actual numbers are very different from the predicted numbers. *To quantify how bad or good results are, we use  $Y - Y^*$ .*

Which model is more useful at  
mapping  $x$  close to true  $Y$ ?

a)



b)



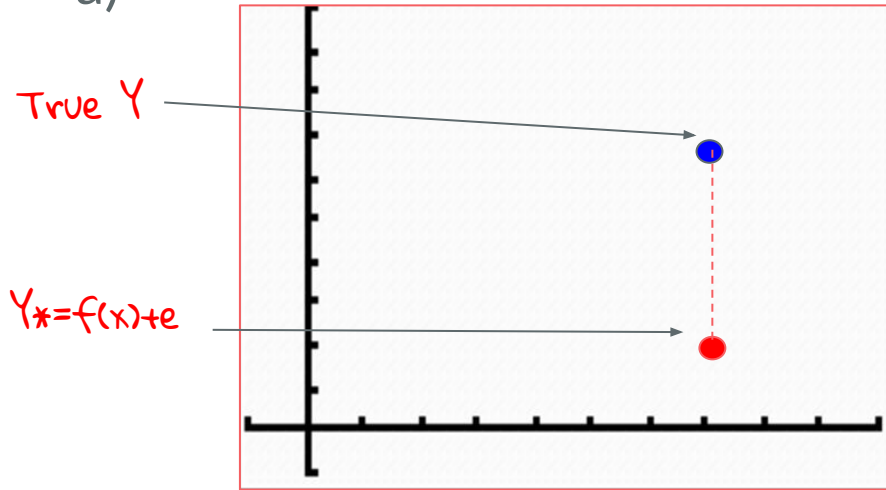
Which prediction was worse, a) or b)?



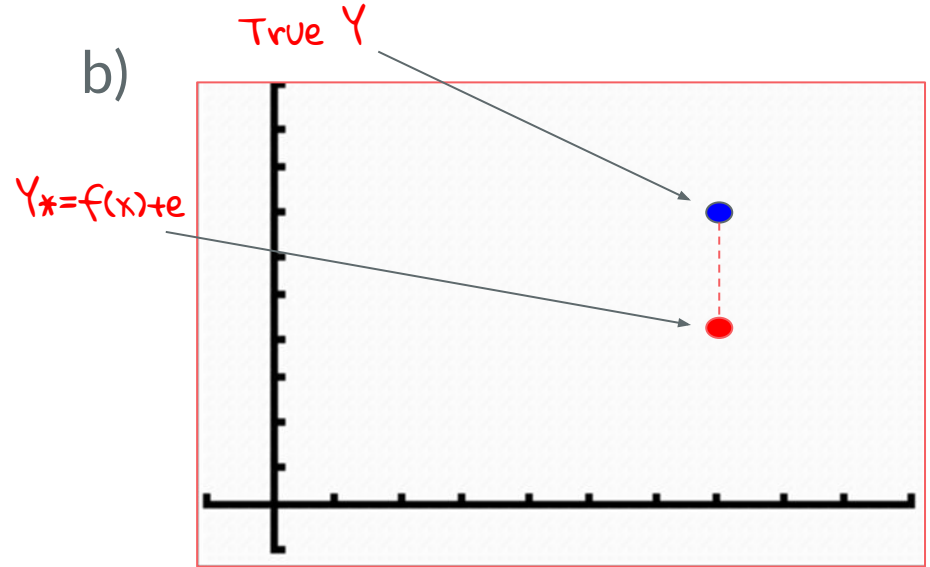
We can immediately tell that  
 $b$  is better!

We can see that the  $f(x)$  in  $b$  maps  $x$  to a  $Y^*$  much closer to the true  $Y$ . A loss function allows us to quantify this difference.

a)



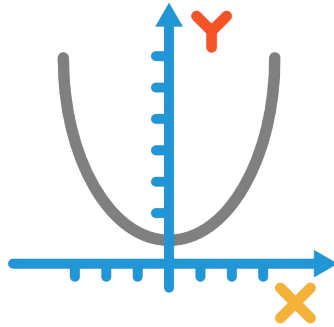
b)



A model's goal is to minimize the loss function.



**The Task**



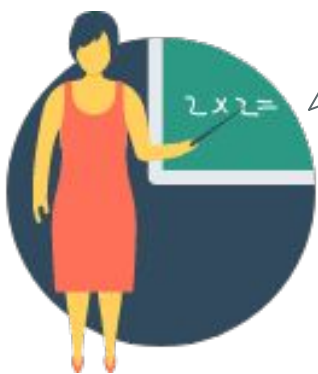
**The Loss Function**

A loss function quantifies how unhappy you would be if you used  $f(x)$  to predict  $Y^*$  when the correct output is  $y$ . It is what we want to minimize.

Another way to think about it is that a loss function quantifies how well our  $f(x)$  fits our data.

We have already seen one simple example:  $Y - Y^*$ , or the difference between the predicted  $Y$  and the actual  $Y$ . Later, we will see more sophisticated loss functions.

# supervised learning



Since I know the right answer, I can compare predicted  $Y^*$  to the true  $Y$  to help guide Mr. Model.

$Y^*$

Predicted Number of people with malaria

2007: 1  
2008: 2000  
2009: 300  
2010: 40

$Y$

Actual Number of people with malaria

2007: 80  
2008: 40  
2009: 42  
2010: 35

We first decide on how to measure how unhappy we are with these results. We call this our **loss function**. On the next slide, we show a few different possible loss functions we can use to assess Mr. Model.



Supervised  
learning task

Recall that there are two different types of tasks:



The Task

Regression

Continuous variable

Classification

Categorical variable

A regression problem is when we are trying to predict a numerical value given some input, such as “dollars” or “weight”.

A classification problem is when are trying to predict whether something belongs to a category, such as “red” or “blue” or “disease” and “no disease”.

The choice of loss function depends upon the type of task. We will discuss loss functions for both types of task.



## Regression

Continuous variable

absolute error  
(L1)

least squares error  
(L2)

Including mean squared error (MSE), root mean squared error (RMSE)

## Classification

Categorical variable

log loss

hinge loss

There are a few different loss functions we could choose from, depending on the problem we are trying to solve.

## Regression

Continuous variable

absolute error  
(L1)

root mean squared  
error (RMSE)

least squares error  
(L2)

mean squared error  
(MSE)

## Classification

Categorical variable

log loss

hinge loss

# Regression loss functions

1. L1 norm (mean absolute error)
2. L2 norm (least squares error)
  - Mean squared error



Our outcome feature is continuous:  
the number of people who have  
malaria.

Outcome feature in  
data is continuous      →      Regression  
task      →      L1 or L2 loss  
function

Y

Number of people  
with malaria

2007: 80

2008: 40

2009: 42

2010: 35



L1 and L2 loss are two possible options for assessing how unhappy we are with Mr. Model's choice of  $f(x)$ .



absolute error  
(L1)

Also called L1 loss, this minimizes the **sum** of absolute errors between True Y and predicted  $Y^*$ .

$$S = \sum_{i=1}^n |y_i - f(x_i)|.$$

least squares error  
(L2)

Also called L2 loss, this minimizes the **square** of the error between True Y and predicted  $Y^*$ .

$$S = \sum_{i=1}^n (y_i - f(x_i))^2$$

mean squared error

Takes the mean of the L2 loss over all observations.

How bad were Mr. Model's initial results?  
Let's compute the L1 norm.



How well did Mr.  
Model's random  
guess perform?

$$S = \sum_{i=1}^n |y_i - f(x_i)|.$$

absolute error (L1)

$Y^*$	$Y$
Predicted Number of people with malaria	Actual Number of people with malaria
2007: 1	2007: 80
2008: 2000	2008: 40
2009: 300	2009: 42
2010: 40	2010: 35

$$(|1-80|+|2000-40|+|300-42|+|40-35|) \\ = 2,302$$

How bad were Mr. Model's initial results? Let's compute the L2 norm.



How did Mr. Model's initial random guess do?

$$S = \sum_{i=1}^n (y_i - f(x_i))^2$$

Least squares error  
(L2)

$Y^*$

Predicted Number of  
people with malaria

2007: 1  
2008: 2000  
2009: 300  
2010: 40

$Y$

Actual Number of  
people with malaria

2007: 80  
2008: 40  
2009: 42  
2010: 35

$$(80-1)^2 + (40-2000)^2 + (42-300)^2 + (35-40)^2 = 3,914,430$$

We can normalize our L2 loss by computing mean squared error or root mean squared error.

least squares error  
(L2)

Also called L2 loss, minimizes the square of the error between True Y and predicted Y\*.

$$S = \sum_{i=1}^n (y_i - f(x_i))^2$$

mean squared error

Takes the mean of the L2 loss over all observations.

$$\text{MSE} = \text{mean}(S)$$

root mean squared  
error

Takes the square root of the mean of the L2 loss.

$$\text{RMSE} = \text{sqrt}(\text{mean}(S))$$



Mean squared error takes the  
average L2 error per  
observation.



How did Mr. Model's  
initial random guess  
do?

MSE = mean(S)

Mean squared error

$Y^*$   
Predicted Number of  
people with malaria

2007: 1  
2008: 2000  
2009: 300  
2010: 40

$Y$   
Actual Number of  
people with malaria

2007: 80  
2008: 40  
2009: 42  
2010: 35

$$\begin{aligned} & ((80-1)^2 + (40-2000)^2 + (42-300)^2 \\ & + (35-40)^2) / 4 \\ & = 978,607.5 \end{aligned}$$

Root mean squared error takes the square root of the average L2 error per observation.



How did Mr. Model's initial random guess do?

$$\text{RMSE} = \sqrt{\text{mean}(S)}$$

Root mean squared  
error

$Y^*$ Predicted Number of people with malaria	$Y$ Actual Number of people with malaria
2007: 1	2007: 80
2008: 2000	2008: 40
2009: 300	2009: 42
2010: 40	2010: 35

$$\begin{aligned} & (((80-1)^2 + (40-2000)^2 + (42-300)^2 + (35-40)^2) / 4)^{(1/2)} \\ &= 989.25 \end{aligned}$$

We can compute for each loss functions how  
unhappy we are with Mr.Model's initial random guess.

Don't worry about these numbers. What's important is you understand how we are transforming them step by step.

2,302

absolute error (L1)

Also called L1 loss,  
minimizes the sum of  
absolute errors between  
True Y and predicted Y\*.

$$S = \sum_{i=1}^n |y_i - f(x_i)|.$$

3,914,430

least squares error  
(L2)

Also called L2 loss,  
minimizes the square of  
the error between True Y  
and predicted Y\*.

$$S = \sum_{i=1}^n (y_i - f(x_i))^2$$

978,608

mean squared error

Takes the average L2 loss  
per observation in the  
data.

$$\text{MSE} = \text{mean}(S)$$

989

root mean squared  
error

Takes the square root of  
the average L2 loss per  
observation in the data.

$$\text{RMSE} = \text{sqrt}(\text{mean}(S))$$

RMSE is the square root of the average L2 loss per observation.



Mr. Model

This job isn't done  
until I reduce  
RMSE.

There are five steps to RMSE:

$Y - Y^*$	For every observation in our dataset, measure the difference between true $Y$ and predicted $Y$ .
$^2$	Square each $Y - Y^*$ to get the absolute distance, so positive values don't cancel out negative ones when we sum.
Sum	Sum across all observations so we get the total error.
mean	Divide the sum by the number of observations we have.
root	Take the square root of the mean calculated above.

MSE



# Which loss function should we use?

1. L1 norm (mean absolute error)
2. L2 norm (least squares error)



Each loss function has important pros  
and cons.

absolute error (L1)

vs.

least squares error  
(L2)



	Robust?	Stable Solution?	How many solutions?
L1	Robust	Not stable	Multiple possible solutions
L2	Not very robust	Stable	One possible solution

MSE and RMSE are both normalized versions of L2 error. If we decide to use L2, we will choose MSE or RMSE.

If we decide to use least squares error (L2), we may decide to report RMSE OR MSE

MSE

VS.

RMSE



The key difference between RMSE and MSE is that taking the root in RMSE normalizes the error to the same units of measurement.

This makes the error term more interpretable.

Both MSE and RMSE amplify and severely penalize large errors more than small ones by squaring the error.

# Classification loss functions

1. Log loss
2. Hinge loss



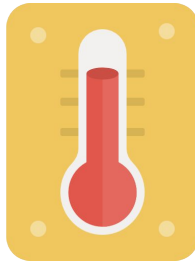
Define explanatory  
and outcome  
feature

Let's define a slightly different task so  
we can discuss hinge and log loss.

**Task:** We want to predict whether or not a patient has malaria  
using their temperature.

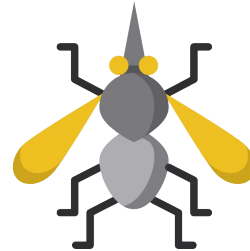
X      Temperature  
         of patient

39.5°C  
37.8°C  
37.2°C  
37.2°C



Y      Does the  
         patient have  
         malaria?

No  
Yes  
Yes  
No



Our outcome feature is categorical: we want to predict whether or not someone has malaria. This is a binary classification problem.

This is a classification task, so we can focus on using either log loss or hinge loss.

But first, what is a classification task?

Classify based  
upon probability  
threshold

Classification tasks output the probability of belonging to a class. Normally, based upon a threshold of 50% we then assign the predicted class.

<u>outcome feature</u>	<u>predicted probability</u>	<u>predicted outcome</u>
Y Does the patient have malaria?	What is the probability that the patient has malaria?	Y* Does the model predict that the patient has malaria?
No	0.55	Yes
Yes	0.80	Yes
Yes	0.85	Yes
No	0.2	No



Classify based  
upon probability  
threshold

We can evaluate accuracy by looking just at the predicted outcome vs. the actual outcome. Here, accuracy is 75%!

outcome feature

Y

Does the  
patient have  
malaria?

No  
Yes  
Yes  
No

predicted probability

What is the probability  
that the patient has  
malaria?

0.55  
0.80  
0.85  
0.2

predicted outcome

Y\*

Does the model  
predict that the  
patient has  
malaria?

Yes  
Yes  
Yes  
No

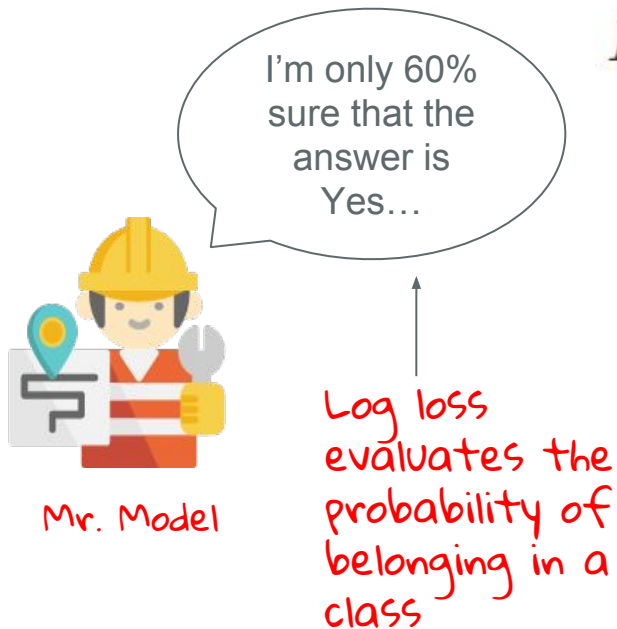
However, we are missing out on using probability, which is important information about how certain the model is about its prediction. Let's look at some loss functions that utilize this metric.





## Log loss

For every prediction Mr. Model Makes, we can measure the logarithmic loss. What is logarithmic loss?



$$L = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

- The smaller the log loss, the smaller the uncertainty, the better the model
  - A perfect classifier would have log loss = 0
- Log loss heavily penalizes classifiers that are confident about an incorrect classification
- Ways to improve log loss:
  - Are there problematic errors in dataset?
  - Do we want to smooth the probabilities?

## Hinge loss

For every prediction Mr. Model Makes, we can also measure the hinge loss. What is hinge loss?

Hinge loss is the logical extension of the regression loss function, **absolute loss**.

**Absolute loss:**  $Y - Y^*$ , where  $Y$  and  $Y^*$  are integers.

**Hinge loss:**  $\max(0, 1 - (Y^*)(Y))$

Where  $Y$  can equal -1 (no) or 1 (yes) for each class.

For each observation, if  $Y^* == Y$  (both are 1 or both are -1), hinge loss = 0. If  $Y \neq Y^*$ , hinge loss **increases**.

*The cumulated hinge loss is therefore the upper bound of the number of mistakes made by the classifier.*

Sources: [https://en.wikipedia.org/wiki/Hinge\\_loss](https://en.wikipedia.org/wiki/Hinge_loss);

[http://scikit-learn.org/stable/modules/generated/sklearn.metrics.hinge\\_loss.html](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.hinge_loss.html)



How do we choose a loss function for a classification problem?

***Depends on the question you want to answer!***

E.g. For a problem where we are trying to assess patient health, we know that *false positives* (the model predicts you do have malaria, but you actually don't) are safer and generally more preferable than *false negatives* (the model predicts you don't have malaria, but you actually do.)

Therefore it is probably safer to evaluate our output as a **probability** of whether or not you have malaria. We will use **log loss**.

## A note on categorical loss functions ...

*We provide only a broad conceptual overview of log loss and hinge loss as we will not be using them in our coding lab. However, we encourage you to explore them further.*

*More resources can be found at the end of this module.*

Mr. Model computed an initial guess  
using RMSE.

**How does he improve on his initial  
guess?**

Our initial RMSE is very high. Mr. Model tries a different  $f(x)$  and compares RMSE.



Oh no! That wasn't very good, let's try something else!

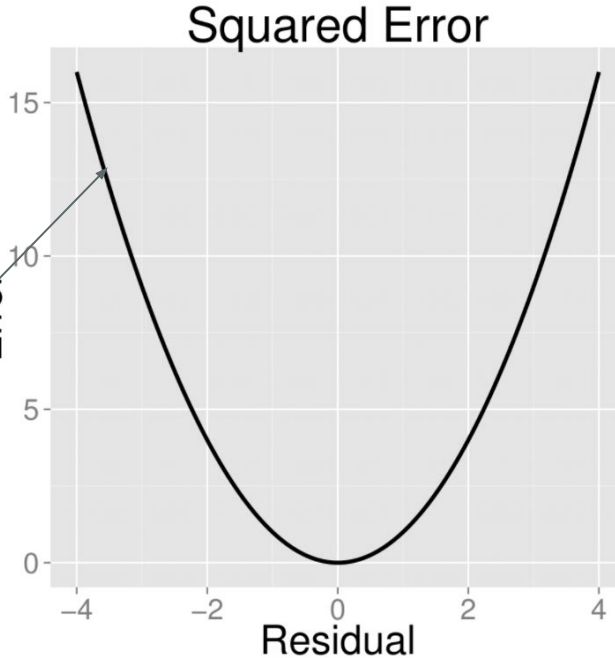
If the new  $f(x)$  reduces the loss, **Mr. Model keeps changing the  $f(x)$  in that direction.** After every change, he measures whether the loss has increased, decreased or stayed the same.

	1st initial guess	2nd update	3rd update
RMSE	1,000	1,300	800
# nets	300	100	400

As he updates his guesses, we see Mr. Model is **learning** - he changes in response to a higher MSE.

The process of changing  $f(x)$  to reduce the loss function is called **learning**. It is what makes ordinary least squares (OLS) regression a machine learning algorithm.

Mr. Model's initial random  $f(x)$  gives us a high initial error



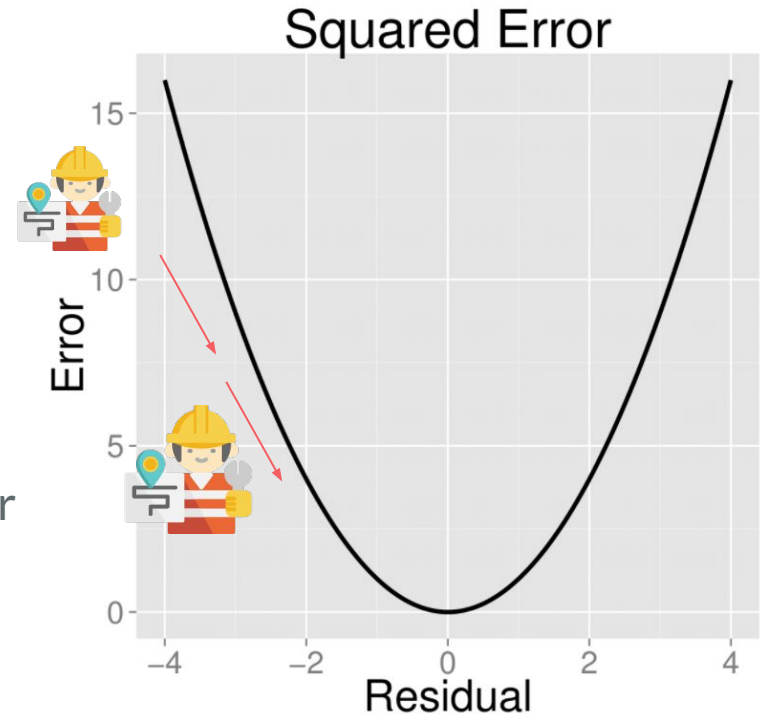
For every  $f(x)$  we choose there is an associated loss.

The learning process involves updating  $f(x)$  in order to reach the global minimum loss.

Mr. Model starts with a random  $f(x)$  and update  $f(x)$  to make our loss as small as possible.

Mr. Model's job is to change the parameters so that every time he changes  $f(x)$ , the loss goes down.

Mr. Model succeeds when he reduces the error to its **minimum**.

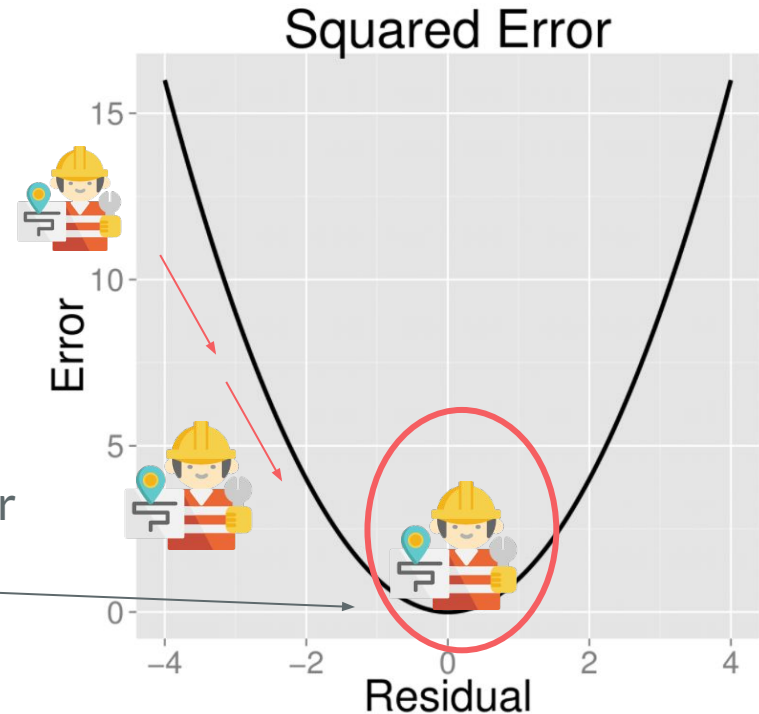




## When does Mr. Model stop?

Mr. Model's job is to change the parameters so that every time he changes  $f(x)$ , the loss goes down.

Mr. Model succeeds when he reduces the error to its **minimum**.



Learning  
Methodology

How does  
our ML  
model learn?

What if we don't have labelled  
data?

Do you have labelled  
data?



Yes

The outcome feature ( $Y$ )  
you are interested in  
predicting is recorded in  
the data. If you have a  
labelled  $Y$ , you can use  
supervised learning  
methods.

$Y$ =Number of  
people with  
malaria

2007: 80  
2008: 40  
2009: 42  
2010: 35

No

The outcome feature ( $Y$ )  
is not recorded in the  
data. You do not have a  
labelled  $Y$ .

$Y$ =Number of people  
with malaria

2007:  
2008:  
2009:  
2010:



Learning  
Methodology

How does  
our ML  
model learn?

When does Mr. Model stop if  
we don't have labelled data?

What happens if we don't actually  
have  $Y$  in our data?

We turn to unsupervised learning  
techniques



Unsupervised learning does not have labelled data. However, it is the most promising current area of research in machine learning. Unlocking unsupervised learning will fundamentally change our world.



## Unsupervised Algorithms

- For every  $x$ , there is no  $Y$ .
- We do not know the correct answers so we cannot act as a teacher.
- Instead, we try to get an understanding of the distribution of  $x$  to draw inference about  $Y$ .

# Why is unsupervised learning important?



Supervised  
learning is the  
icing on the  
cake

Unsupervised  
learning is the  
cake itself

Yan Lecun, a deep learning researcher, made the analogy that if intelligence was a cake, unsupervised learning would be the cake and supervised learning would be the icing on the cake.

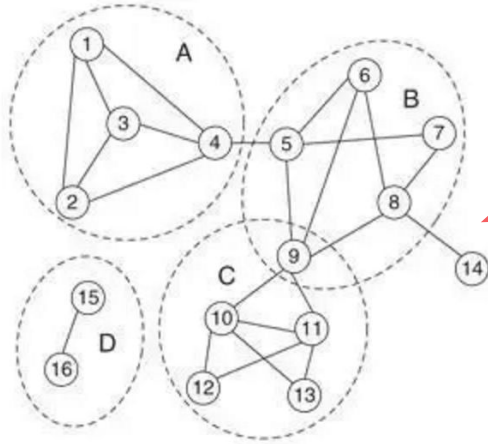
We know how to make the icing, but we don't know how to make the cake. **Unsupervised learning is the holy grail of machine learning.**

To reach true machine intelligence, ML needs to get better at unsupervised learning.

Humans learn mostly through unsupervised learning: we absorb vast amounts of data from our surroundings without needing a label.

Examples of unsupervised learning: This clustering algorithm predicts a user's friends based upon their activity on social networks.

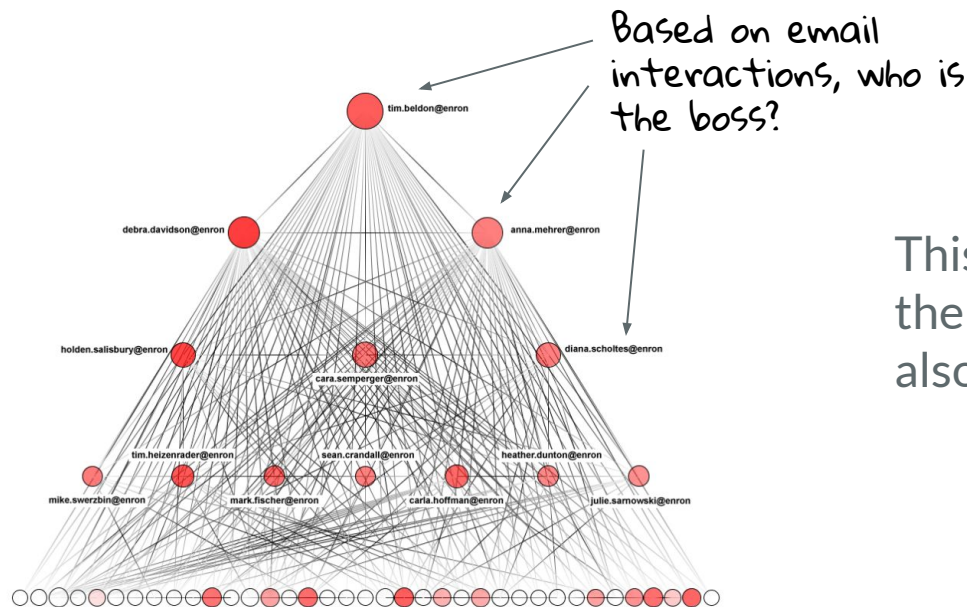
**Social Network Analysis:** In a social network, clustering can be used to find users that interact a lot with each other (say, via e-mails). This is shown in the figure below where the users have been clustered into four clusters - A,B,C and D.



We don't have any labelled data that tells us any node is friends with another node (i.e., x is friends with Y).

Instead, we can use user interactions to provide the labels. The assumption is that if you are interacting heavily with someone they are more likely to be your friend.

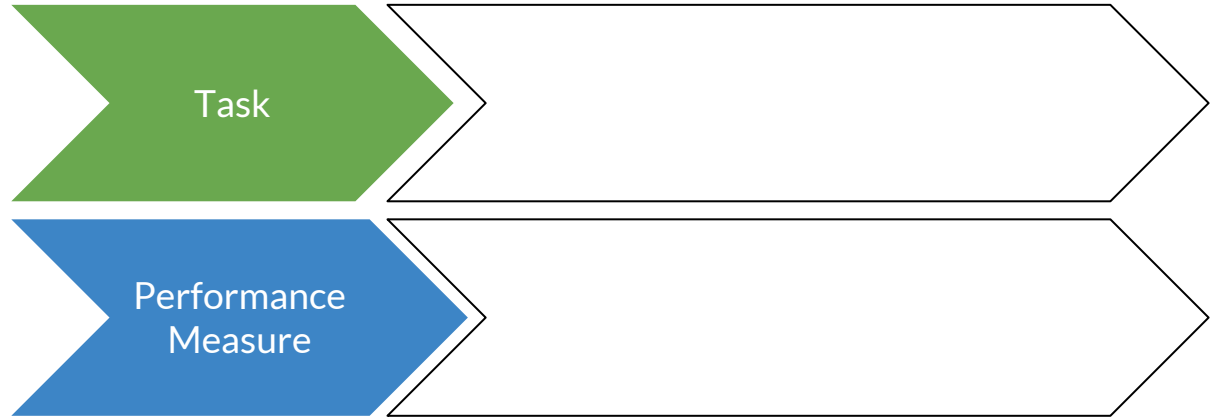
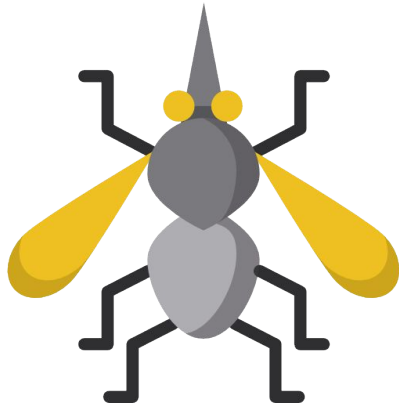
The clustering algorithm uses email patterns to predict the hierarchy of a business organization.



This algorithm not only takes in account the people involved in an interaction but also the directionality of the interaction.

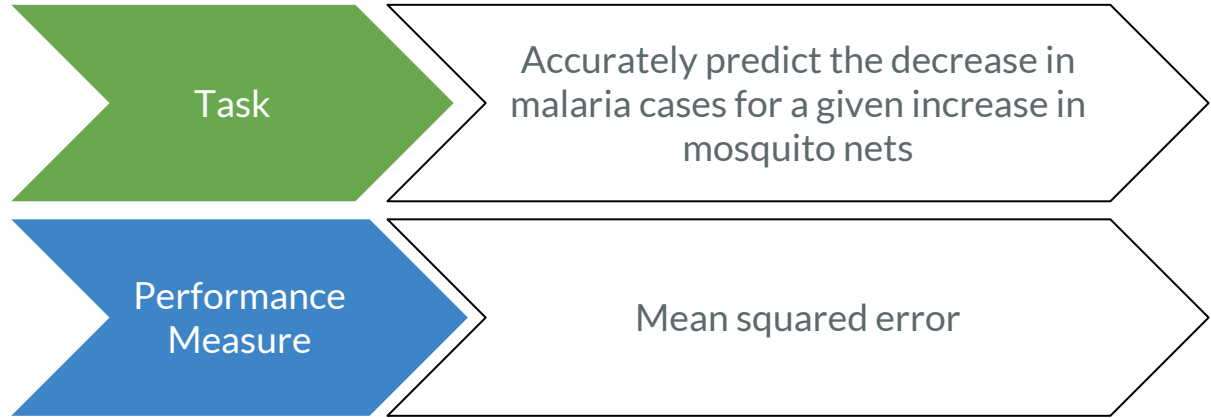
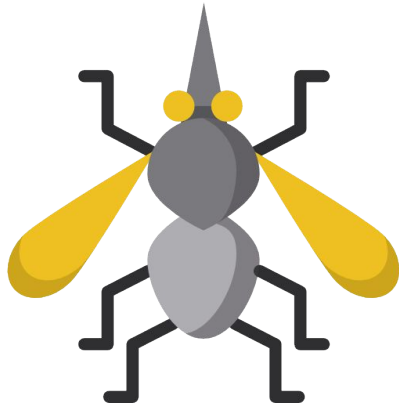
Figure 1: Enron North American West Power Traders Extracted Social Network

What is the performance measure for our malaria net example?

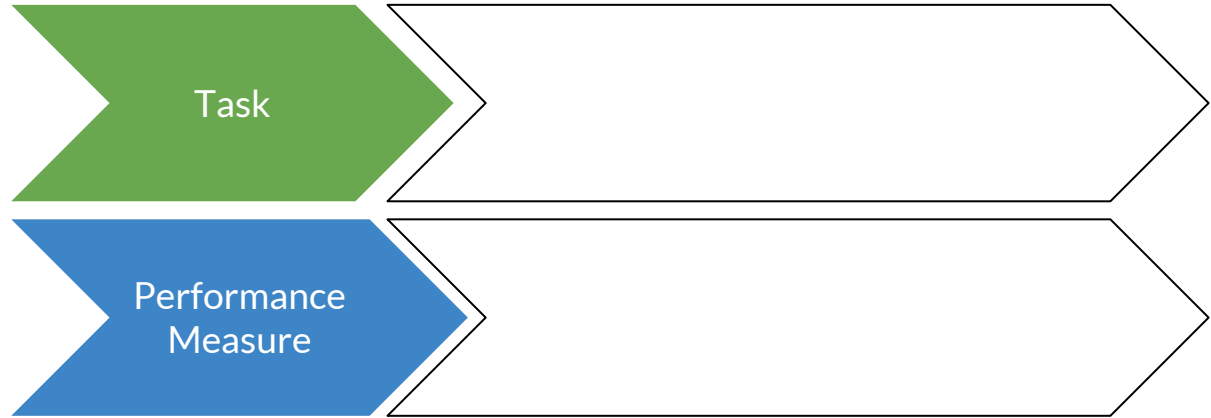
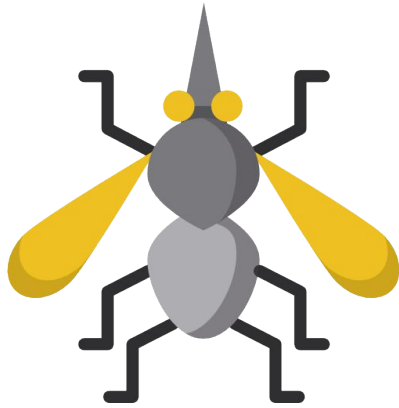




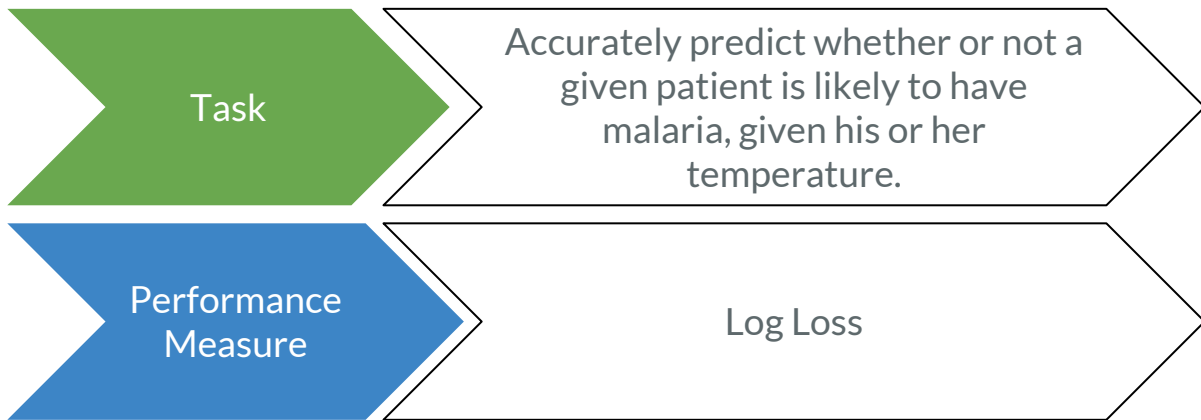
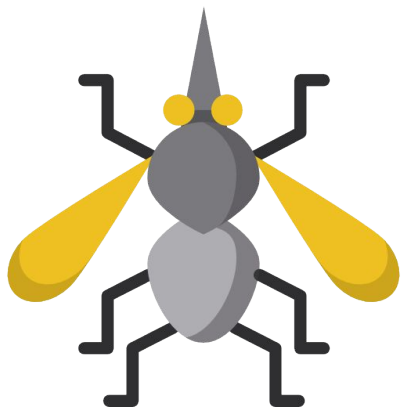
What is the task and the performance measure for our malaria net example?



What is the task and the performance measure for our malaria patient example?



What is the task and the performance measure for our malaria patient example?



In the next class, we will  
introduce model  
selection and evaluation.



# Additional resources



# Additional resources

Rosasco, Lorenzo, et al. "Are loss functions all the same?." Neural Computation 16.5 (2004): 1063-1076. <http://web.mit.edu/lrosasco/www/publications/loss.pdf>

"Loss Functions for Regression and Classification", David Rosenberg, NYU: <https://davidrosenberg.github.io/ml2015/docs/3a.loss-functions.pdf>

