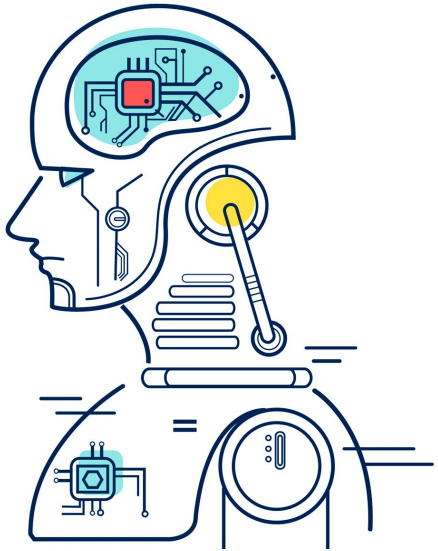


MACHINE LEARNING:

framing and building basics

JONATHAN MERLOS

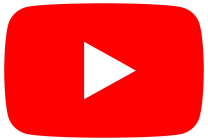
PROJECT MANAGER

Jonathan Merlos

Project Manager

jonathanmerlos

Why machine learning?



Machine Learning Series

ML: A light introduction
ML: Data Preparation
ML: Feature Engineering
ML: Classification
ML: Neural Networks
ML: Embedding
ML: Into Production & Fairness

based on Google **Machine Learning Crash Course**

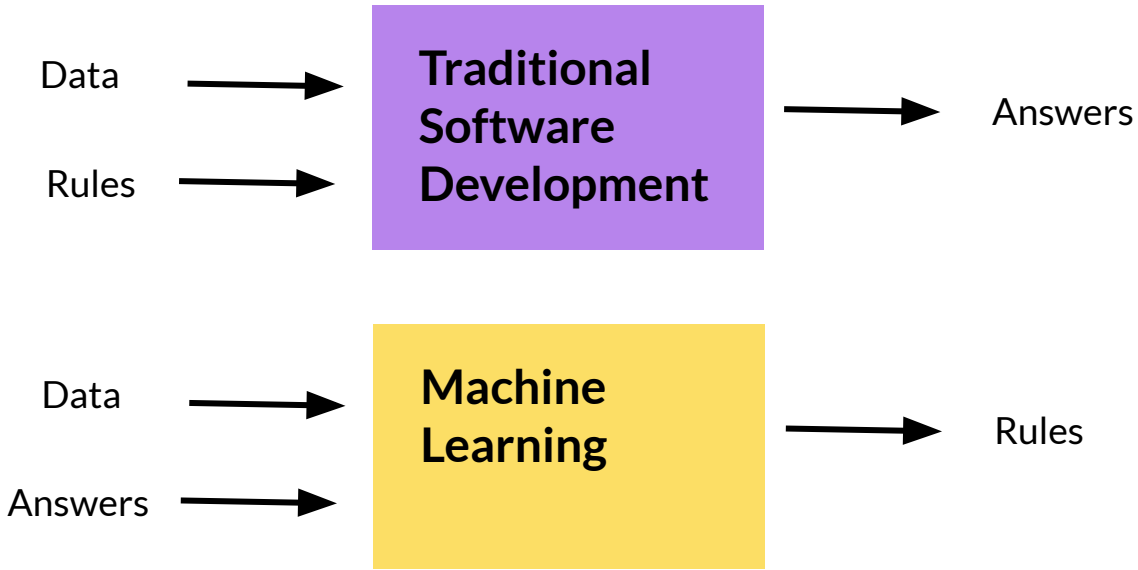
ML: A Light Introduction

Framing

Linear Regression Models

Reducing Loss

New Paradigm



Key Terminology

Labels

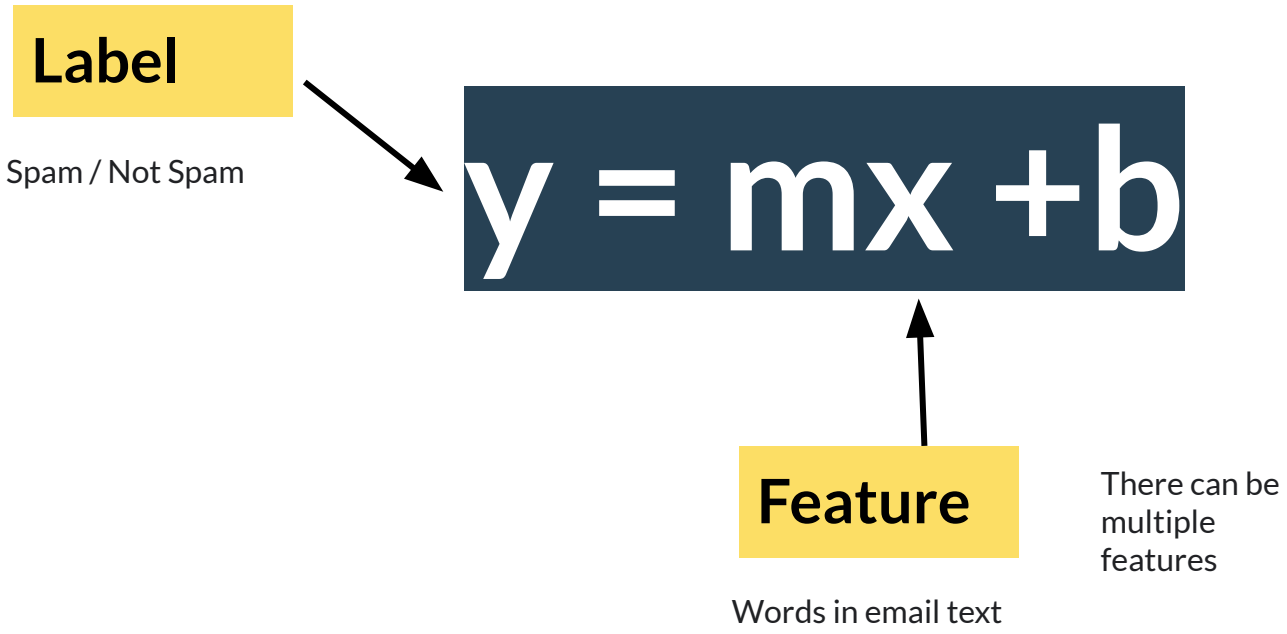
A **label** is the thing we're predicting—the **y** variable in simple linear regression.

- Whether a mail is spam or not
- Future price of wheat
- The kind of animal shown in a picture
- The meaning of an audio clip
- Just about anything

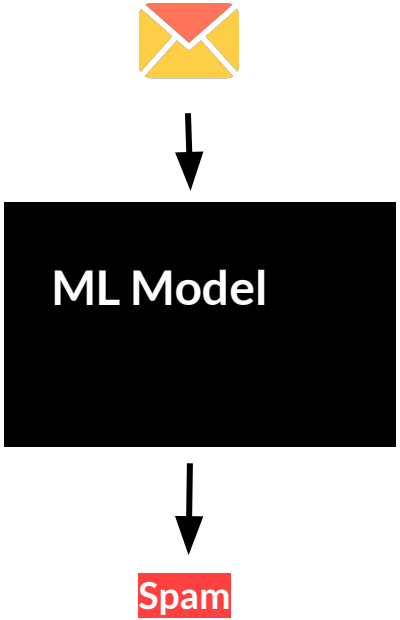
Feature

A **feature** is an input variable—the **x** variable in simple linear regression.

- Words in the email text
- Sender's address
- Time of day the email was sent
- Email contains the phrase "one weird trick."


Label
Spam / Not Spam
y = mx +b
Feature
Words in email text
There can be multiple features

Spam Detection Model



Types of Examples

Labeled

A **labeled example** includes both feature(s) and the label. Useful to **train** the model.

```
labeled examples: {features, label}: (x, y)
```

Sender	Sent Time	Words in body	Label
imspam@spam.com	00:24:15	200	Spam
jrobertson@gmail.co	16:15:24	50	Not Spam
shiwoo4@gmail.com	11:15:33	40	Not Spam

Unlabeled

An **unlabeled example** includes feature(s), but not the label. Useful to **test** the model.

```
unlabeled examples: {features, ?}: (x, ?)
```

Sender	Sent Time	Words in body	Label
dsmith5@gmail.com	09:15:16	65	?
notspam@spam.com	01:15:24	210	?
celesrios@gmail.com	13:11:04	108	?

Models

Training

Creating or **learning** the model.

Inference

Means applying the trained model to unlabeled examples.

Labeled examples

Unlabeled examples

ML Model

Predictions

ML: A Light Introduction

Framing

Linear Regression Models

Reducing Loss

Supervised learning

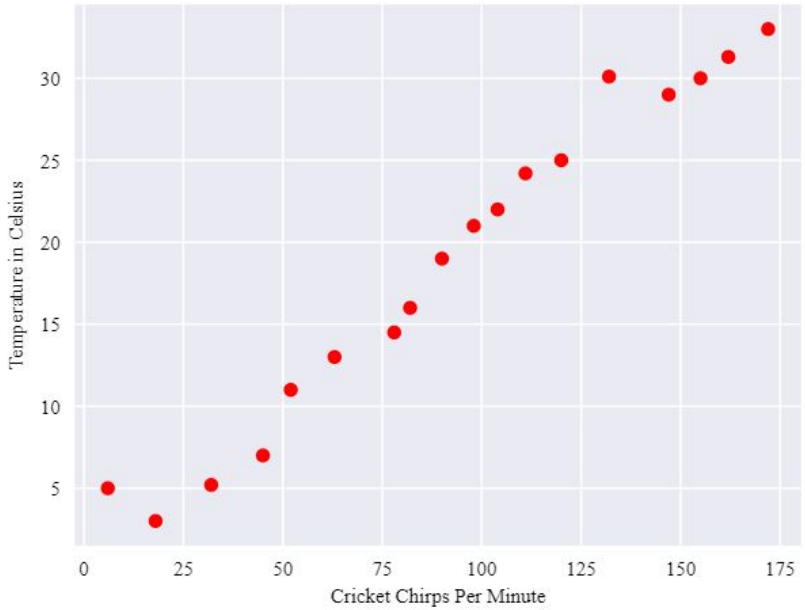
Unsupervised learning

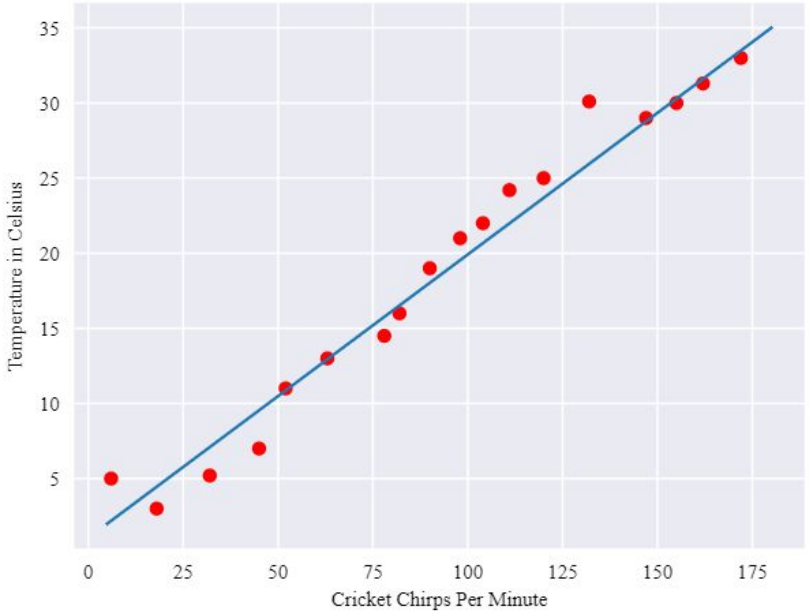
Supervised learning

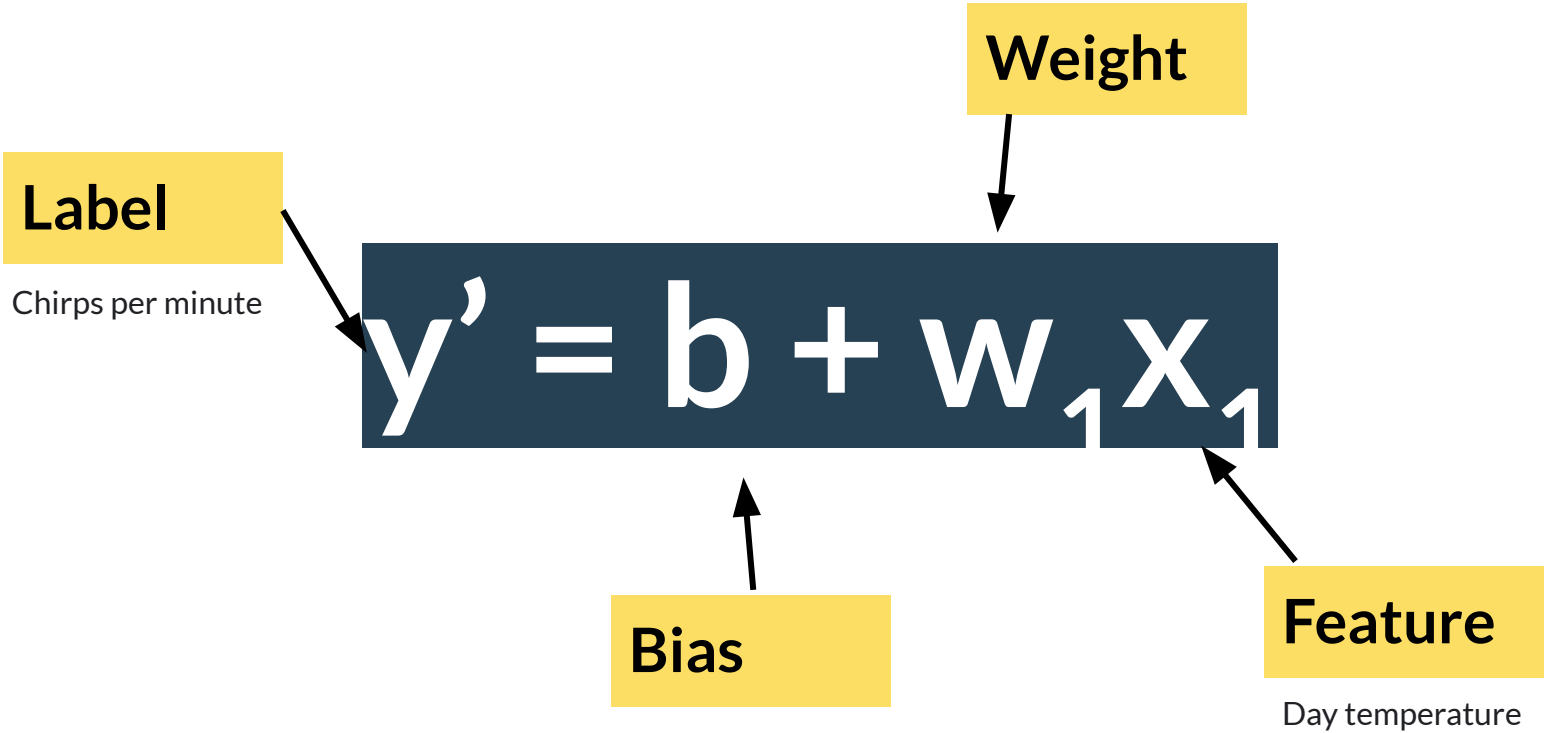
Regression

Classification

Chirps per Minute vs Celsius Temperature






Weight
Label
Chirps per minute
$y' = b + w_1 x_1$
Bias
Feature
Day temperature

Training and Loss

Training a model simply means learning (determining) good values for all the weights and the bias from labeled examples.

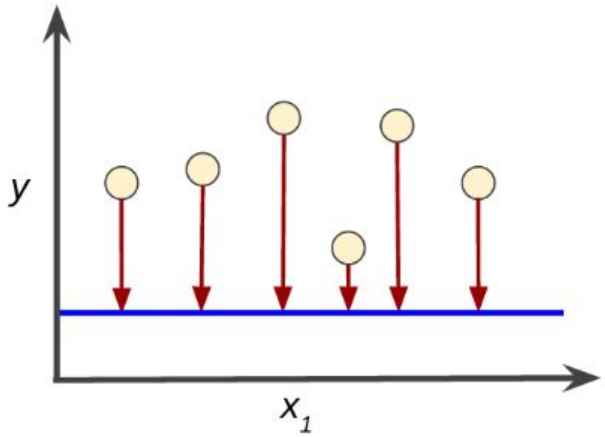
Empirical Risk Minimization

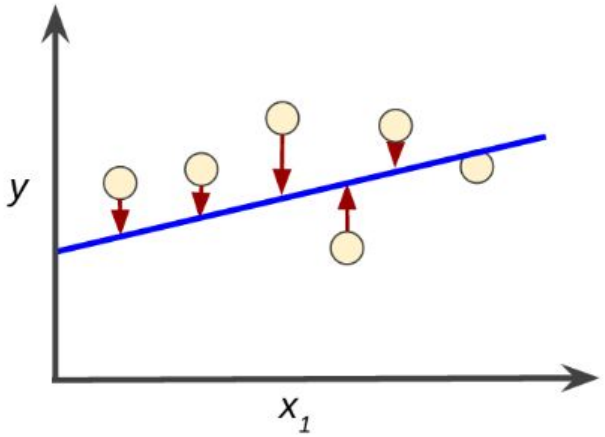
In supervised learning, a machine learning algorithm builds a model by examining many examples and attempting to find a model that minimizes loss.

Loss

The penalty for a bad prediction. That is, loss is a number indicating how bad the model's prediction was on a single example.

Loss





What model is better?

Can we represent this numerically?

Squared Loss

A popular function to measure loss in linear regression.

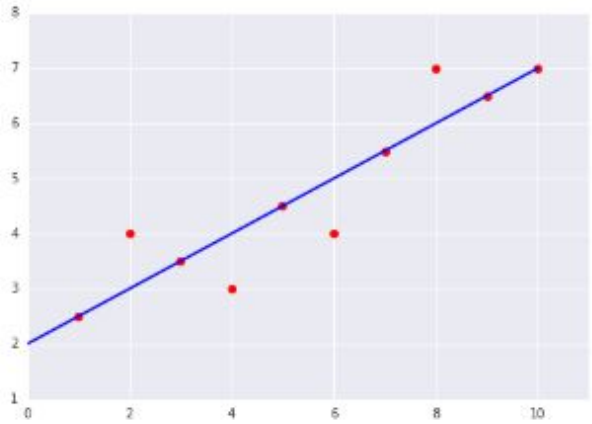
```
= the square of the difference between the label and the prediction
= (observation - prediction(x))2
= (y - y')2
```

Mean Square Error (MSE)

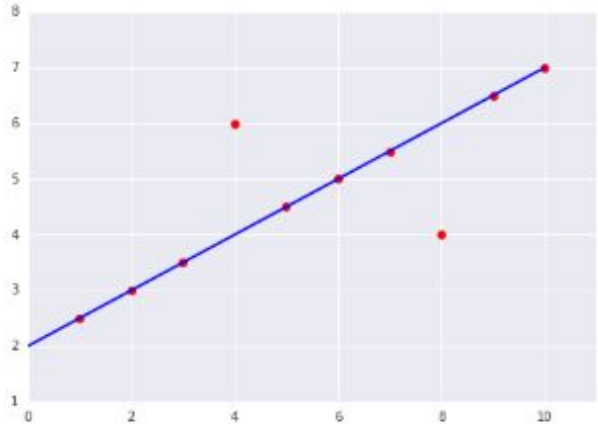
The average squared loss per example over the whole dataset

$$MSE = \frac{1}{N} \sum_{(x,y) \in D} (y - prediction(x))^2$$

Comparing Losses



MSE = 0.4



MSE = 0.8

What model is better? Why?

Lower Loss, always better?

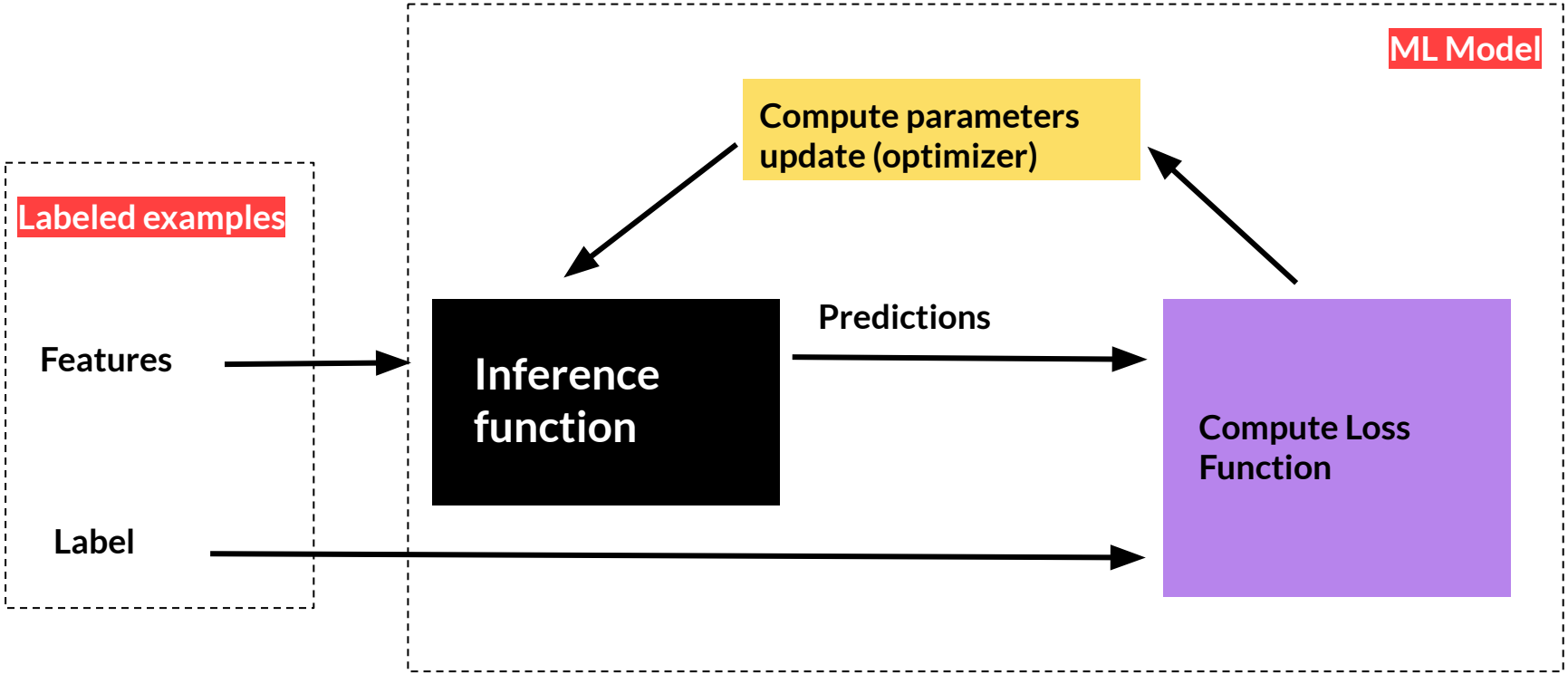
ML: A Light Introduction

Framing

Linear Regression Models

Reducing Loss

An iterative approach



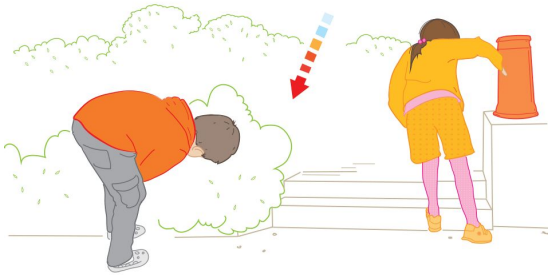
Compute parameters update (optimizer)

It's where the machine learning system examines the value of the loss function and generates new values for
b and w_1.

Assumptions:

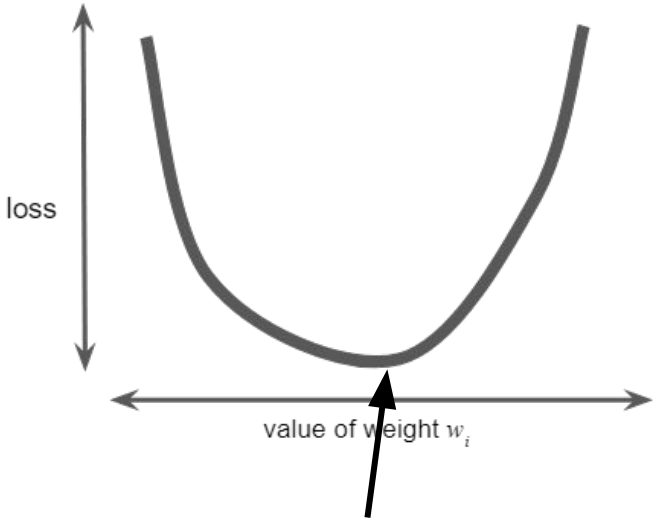
- This mysterious box devises **new values.**
- Then the machine learning system **re-evaluates** all those features against all those labels.
- Yields a new value for the loss function, which yields new parameter values.
- The learning continues **iterating** until the algorithm discovers the model parameters with the lowest possible loss. **(convergence).**

Hot and Cold game

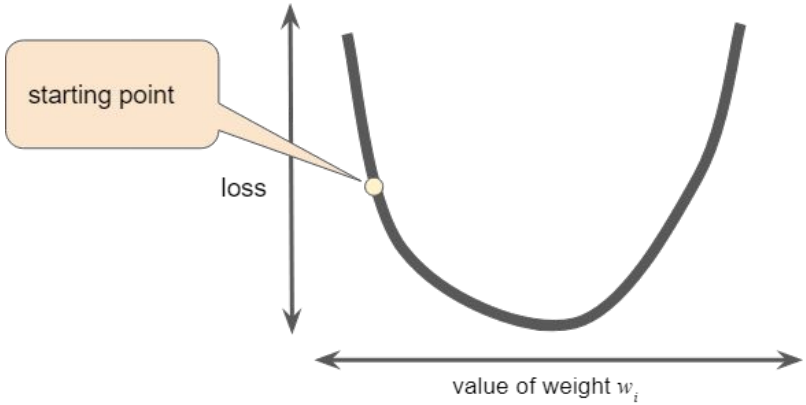


Gradient Descent

Convex problems



Singular Convergence Point



Gradient Descent Box

Old Parameters →

Gradient Descent Function

$$\nabla f(x,y) = \left(\frac{\partial f}{\partial x}(x,y), \frac{\partial f}{\partial y}(x,y)\right) = (e^{2y}\cos(x), 2e^{2y}\sin(x))$$

→

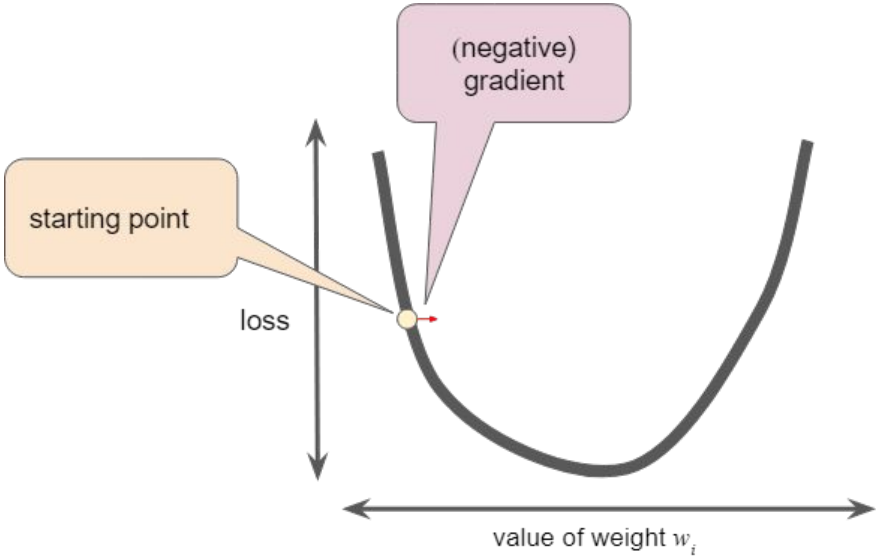
Vector

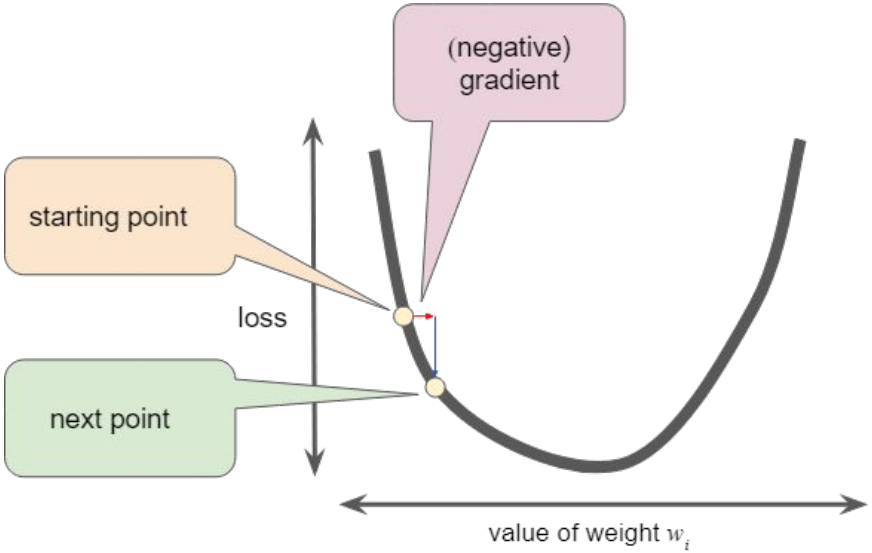
Direction

Magnitude

∇f	Points in the direction of greatest increase of the function.
$-\nabla f$	Points in the direction of greatest decrease of the function.

Negative Gradient Descent





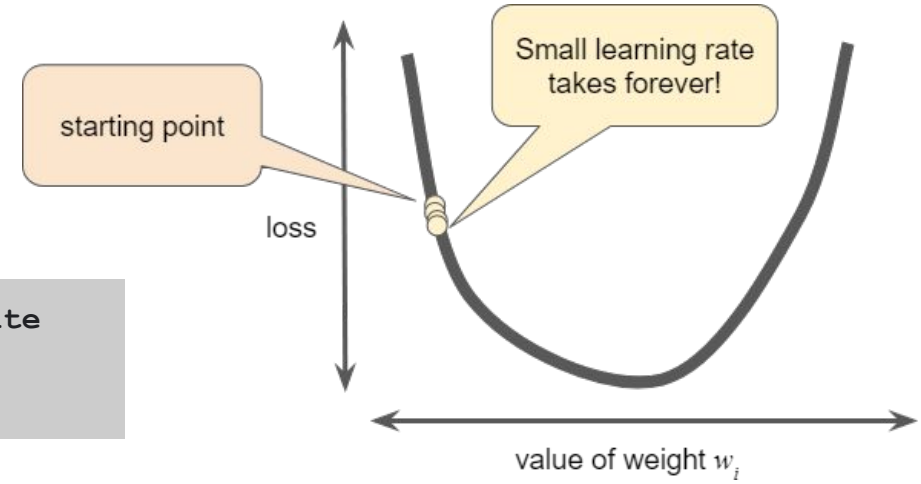
Hyperparameters

Hyperparameters are the knobs that programmers tweak in machine learning algorithms.

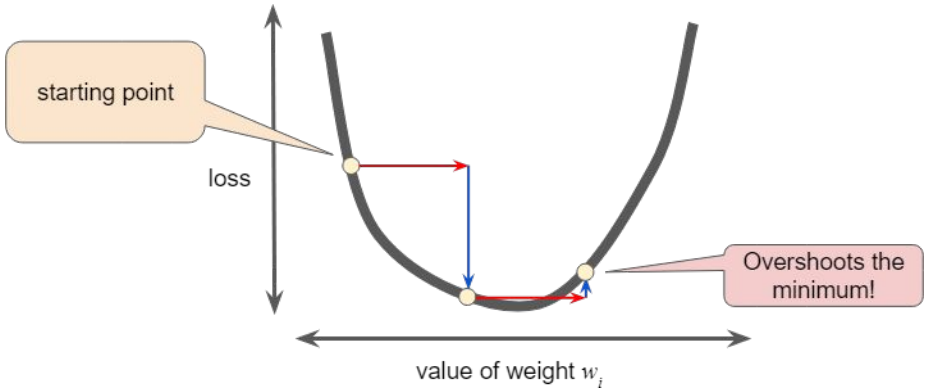
Learning Rate

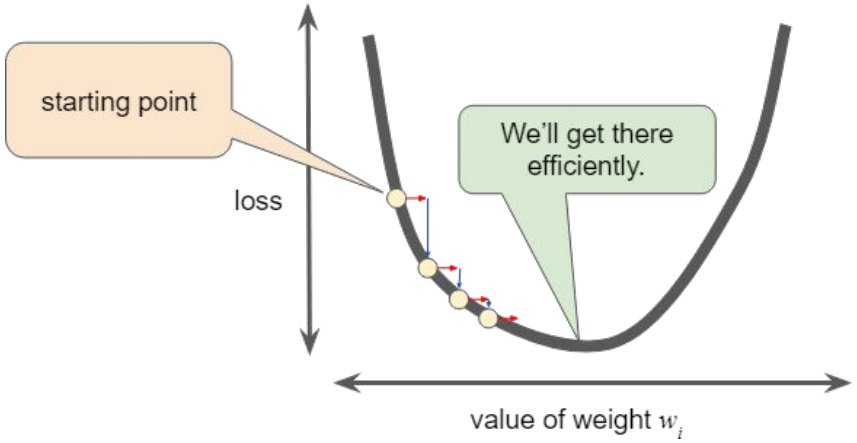
Gradient descent algorithms multiply the gradient by a scalar known as the **learning rate** (also sometimes called **step size**)

```
= gradient descent magnitude * learning rate
= 2.5 * 0.01
= 0.025
```



Learning Rate





Goldilocks Learning Rate

Related to how flat our loss function is.

More Hyperparameters

Batch Size

A **batch** is the total number of examples you use to calculate the gradient in a single iteration.

A very large batch may cause even a single iteration to take a very long time to compute.

Enormous batches tend not to carry much more predictive value than large batches.

Stochastic Gradient Descent (SDG)

Batch size of 1 per iteration.

Mini-batch stochastic gradient descent (mini-batch SGD)

Batch size of 10-1000 per iteration.

More Hyperparameters

Epoch

A **full training pass** over the entire dataset such that each example has been seen once.

Thus, an epoch represents **N/batch size** training iterations, where N is the total number of examples.

Supervised learning

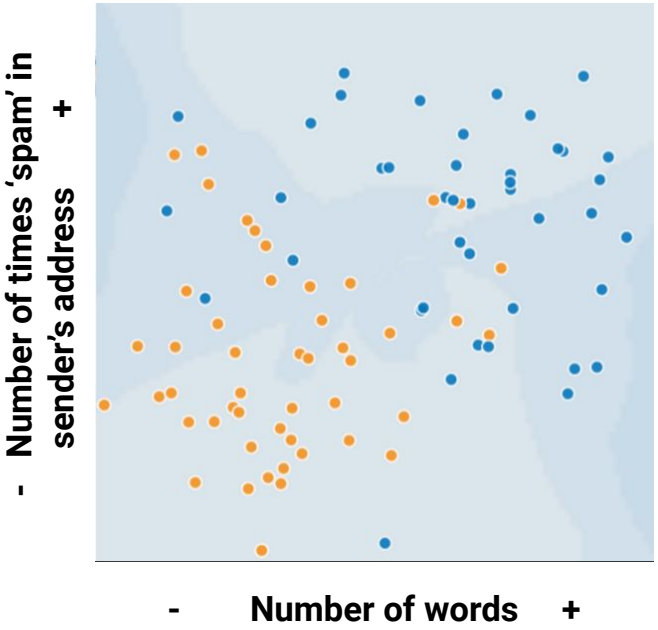
- Regression
- Classification

What can we use?

Sender	Sent Time	Words in body	Label
imspam@spam.com	00:24:15	200	Spam
jrobertson@gmail.co	16:15:24	50	Not Spam
superspam@spami.com	16:14:13	191	Spam
shiwoo4@gmail.com	11:15:33	40	Not Spam
jerickson@hotmail.com	03:24:16	71	Not Spam
notspam@imspam.com	22:04:15	220	Spam
isspam@notspam	16:09:07	114	Spam

...

Let’s build the model



Helpers

Loss Function
MSE

Optimizer
SGD-MbSGD

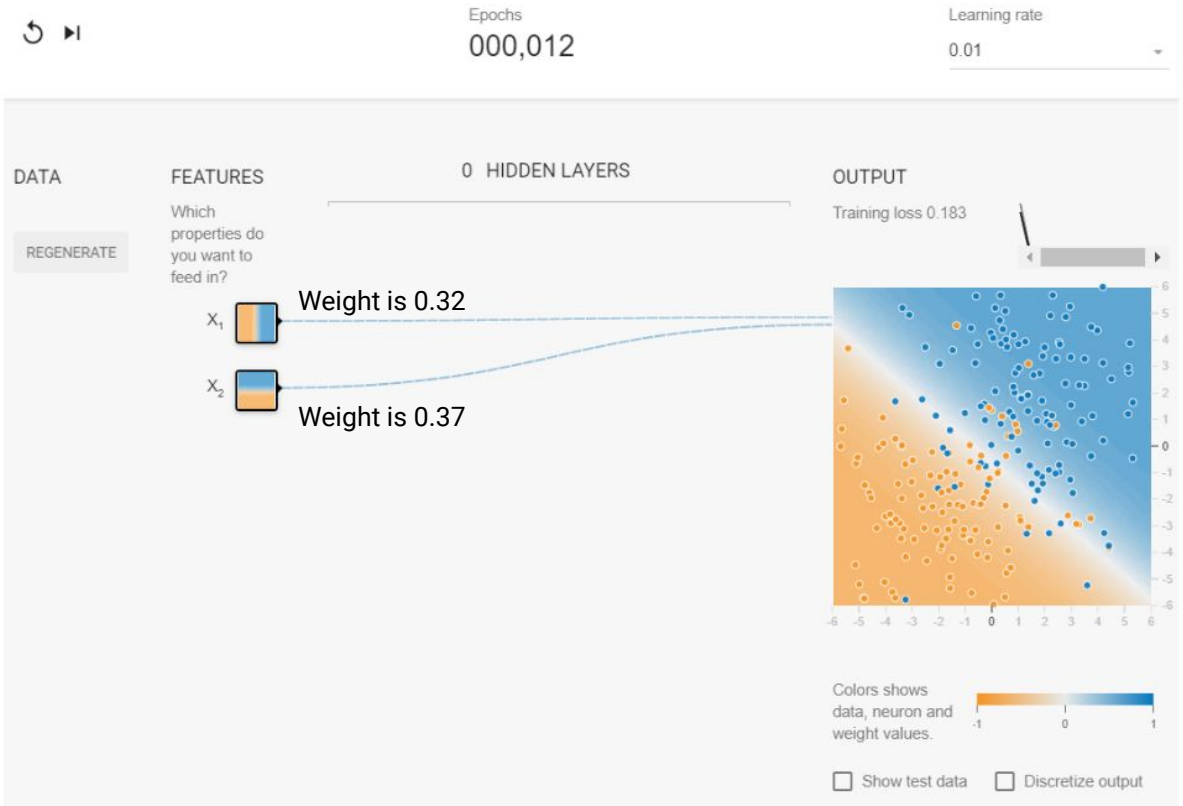
Hyperparameters

Learning Rate

Batch Size

Epochs

The Model



Parameters

Learning Rate
0.01

Batch Size
Full Dataset

Epochs
12

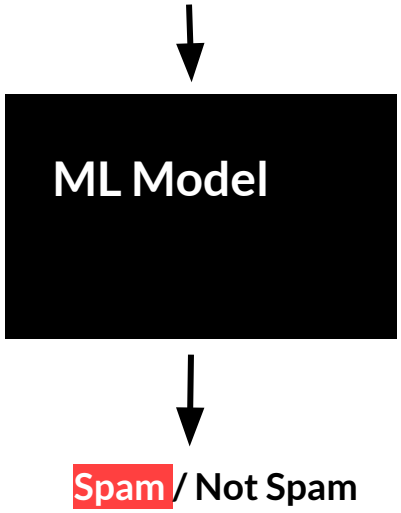
w_1 0.32

w_2 0.37

MSE 0.183

Our results

- Number of words
- Frequency of word ‘spam’ in sender

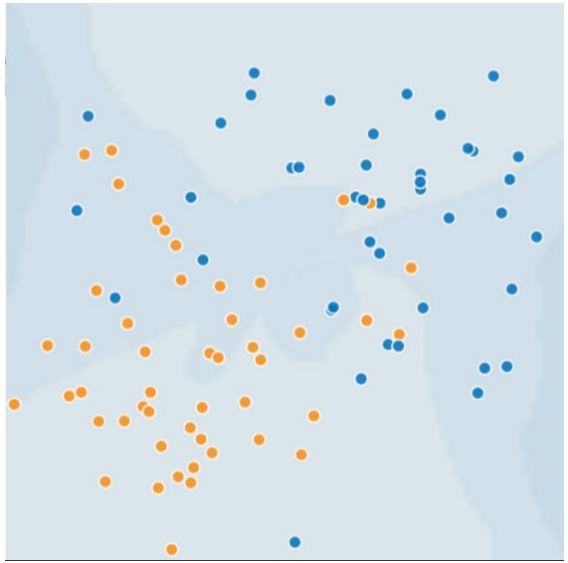


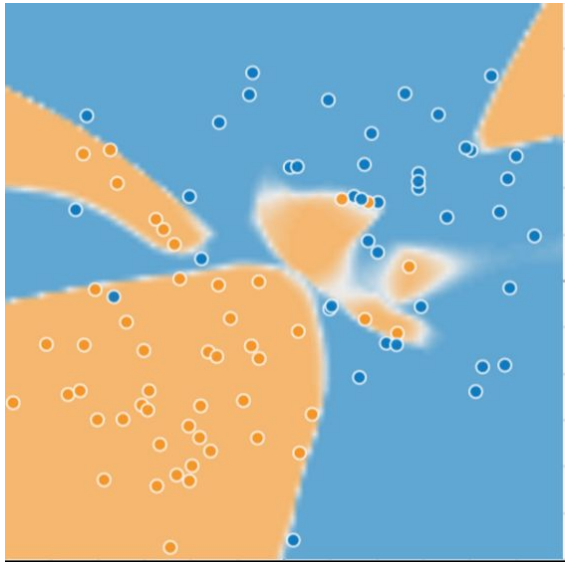
Spam / Not Spam

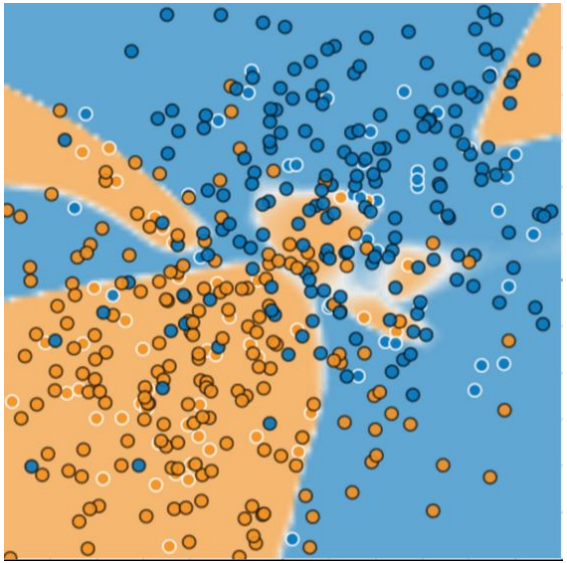
95%

Accuracy

What's coming?







Low Loss.
Still a bad model.

Join the community

/chiseiai

/chiseiai

/chiseiai

Jonathan Merlos

Project Manager

jonathanmerlos

Thank you!