

Dataism - Week 3



Instagram (@thedazzleclub)

Data Science and Activism

How does an understanding of data science aid activism?

Our case studies fall into one of two categories:

- Criticize, subvert, draw attention to ways data is being (mis)used.
 - WMDs, CV Dazzle/Dazzle Club, limits of GDP
- Directly apply Data Science methods toward activist goals
 - Police Scorecard, VFRAME, Gringgo

Moreover, in both cases, a better understanding helps us figure out which questions to ask.

General Concepts/Tools from last week...

- What to consider when trying to design a useful metric/look critically at an existing one.
- Converting a metric to 0-100 percentile scale by comparing all entities to which the metric applies (ex. Police depts in CA)
- Measuring bias using normal distributions (Z-scores)
 - Concept of normal distributions in general is important!
 - <https://studiousguy.com/real-life-examples-normal-distribution/>
 - Ex. Gender bias in salary, identifying high polluters and low polluters
- Tools: Colab, numpy, pandas, matplotlib, computations in python (particularly Z-scores and percentiles), plotting histograms.

Goals for today...

Brief overview of case studies related to image recognition.

A concrete understanding of supervised learning + kind of problems it can address.

A detailed look at linear regression (example of supervised learning)

Discuss issues of bias, interpretability, transparency in deep learning models.

Share resources for using/learning ML (in activist projects).

Broad discussion in breakout groups.

Case Studies

Today's case studies are based on **computer vision** (specifically image recognition):

- Training computers to interpret and understand the visual world
- Field has been taken over by machine learning, specifically deep learning (i.e. models built with many-layered neural networks)

CV Dazzle/Dazzle Club

Problem:

- Companies like Clearview AI help law enforcement use face recog to identify people from photos or video
- Data often scraped from social media, with cooperation of social media companies
- Dubious claims about predicting emotion, likelihood that someone is a criminal...



Look N° 3

For DIS Magazine (2010)

Creative direction by Lauren Boyle and Marco Roso

Model: Jude

Hair: Pia Vivas

CV Dazzle/Dazzle Club

Problem:

- Racial bias built into tech: MIT study found facial analysis software was inaccurate up to 34% of the time in identifying dark-skinned women, compared to 0.8% for light-skinned men.
- (Note for protestors: wear a mask, don't photograph protestors faces, social media platform cooperate with police/immigration)



Look N° 3

For DIS Magazine (2010)

Creative direction by Lauren Boyle and Marco Roso

Model: Jude

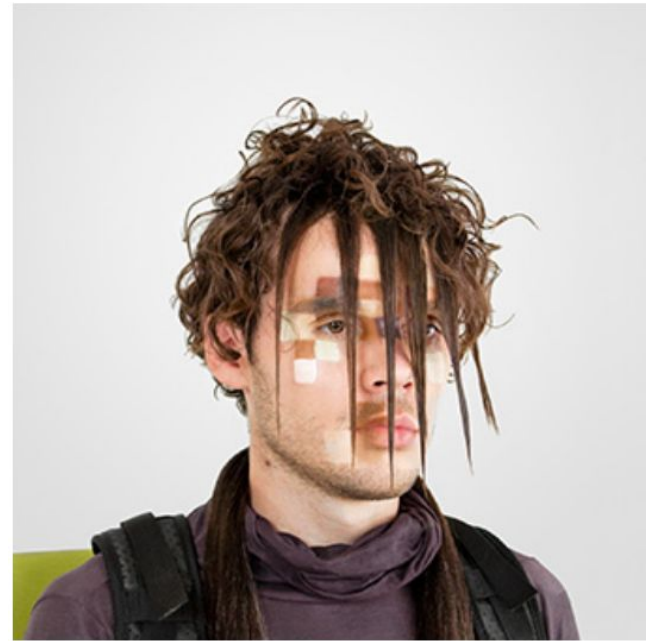
Hair: Pia Vivas

CV Dazzle

Response:

- Artist/researcher Adam Harvey created camouflage makeup to trick facial recog.
 - Many looks out of date (based on old algorithm)
 - Principle always holds
 - Find target algorithm
 - Reverse engineer it
 - Develop look based on vulnerabilities

But, Hanwang Technology in China claims to identify people wearing masks w/ 95% acc... wear sunglasses!



Look N° 3

For DIS Magazine (2010)

Creative direction by Lauren Boyle and Marco Roso

Model: Jude

Hair: Pia Vivas

Dazzle Club

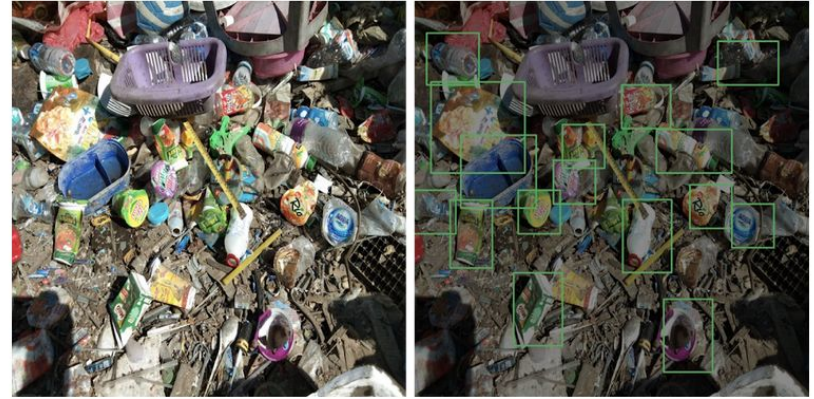
- Founded by: Emily Roderick, Georgina Rowlands, Anna Hart and Evie Price
- Public “walks” wearing CV Dazzle makeup
- Using art to question normalization of surveillance, changing understanding of what it means to exist and move in public spaces in the 21st century



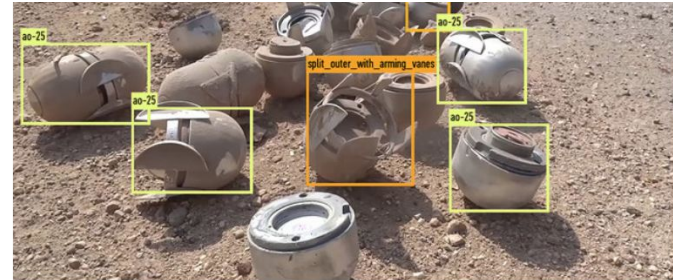
Emily Roderick, Evie Price and Anna Hart, founders of the Dazzle Club, wearing makeup designed to confuse facial recognition cameras. Photograph: Cocoa Laney/The Observer

Other projects

- Gringgo - Using image recognition to map location/type of garbage in Indonesia. Link to monetary reward for collection/disposal. Encourage citizen-based waste management infrastructure.
- VFRAME - Also Alex Harvey. Collection of open-source CV tools designed specifically for human rights investigations



A mock-up shows how Gringgo thinks the app will be able to identify waste through AI-powered image recognition



Munition Detector

Training object detection algorithms to locate illegal munitions

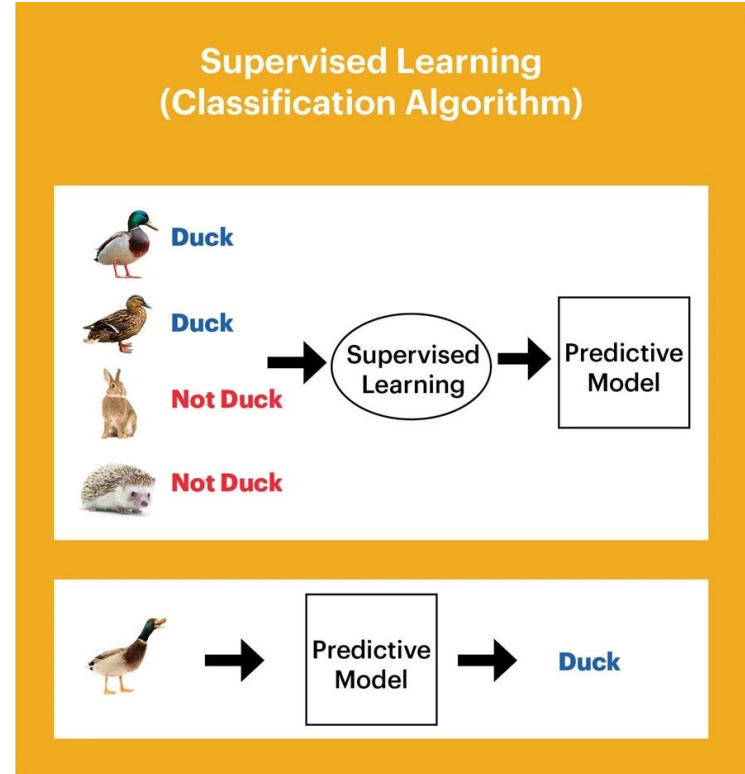
Underlying data science: All built on top of image recognition models constructed using machine learning, specifically *supervised* learning with deep neural nets.

Supervised Learning

Simplest starting point for machine learning, but ubiquitous.

What is machine learning?

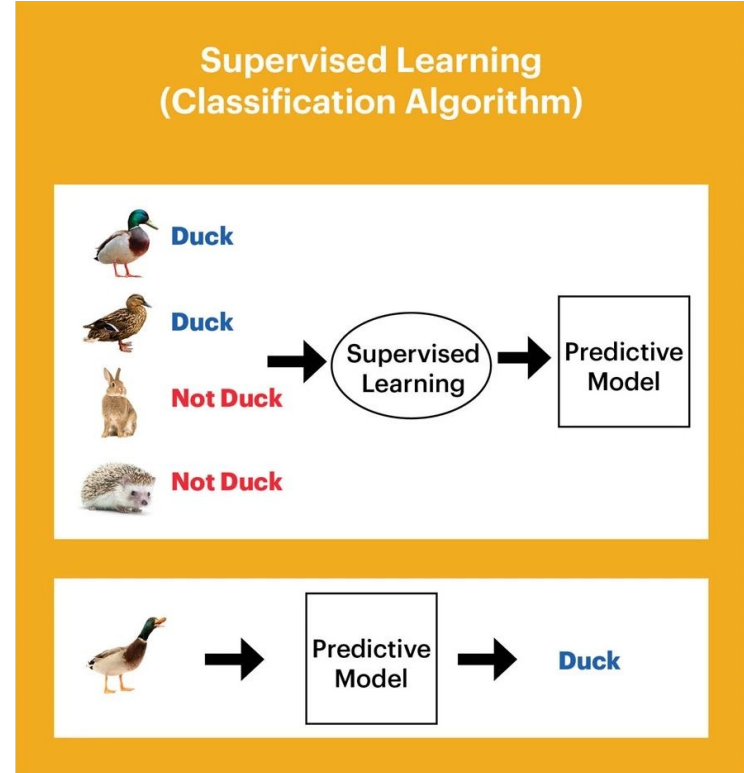
What is supervised learning?



Supervised Learning

Simplest starting point for machine learning, but ubiquitous.

- What is machine learning?
 - Vaguely: “Study of computer algorithms that improve automatically through experience.”
- What is supervised learning?
 - Learning a “function” that maps an input to an output based on a “training set” of example input-output pairs.



Supervised Learning

Opposite of classical programming

- Classically, you would start with an image as input and code a set of instructions (an algorithm) to determine who's in it (if president and chinstrap, then Abe Lincoln).
- Input is image data, a matrix of pixel values.
- Would be a nightmare to write such an algorithm directly!

An image, then, can be represented as a matrix of pixel values.



187	183	174	168	160	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	6	124	131	111	125	204	165	15	56	180
194	68	137	251	237	239	238	227	87	71	201	
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	80	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

187	183	174	168	160	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	6	124	131	111	125	204	165	15	56	180
194	68	137	251	237	239	238	227	87	71	201	
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	80	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

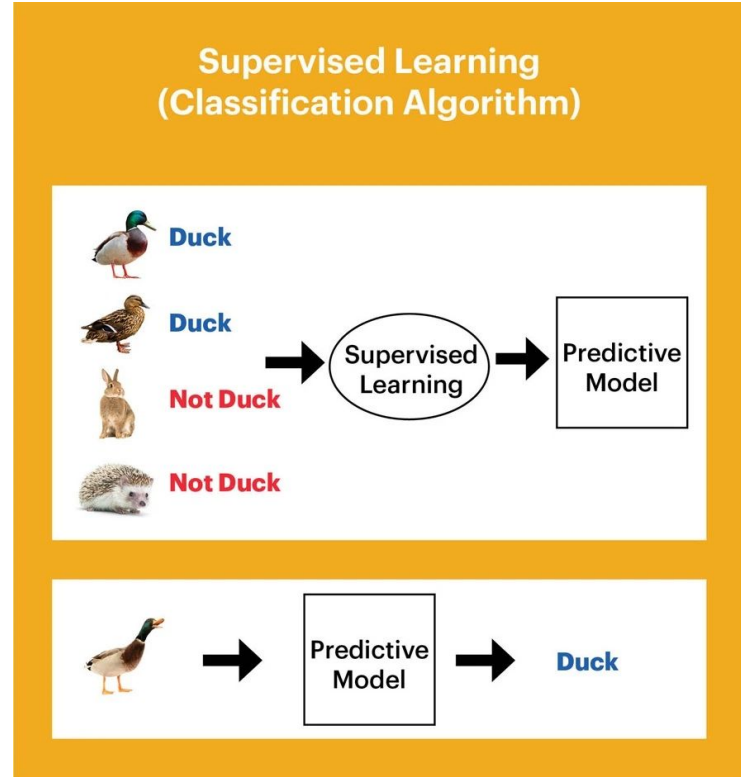
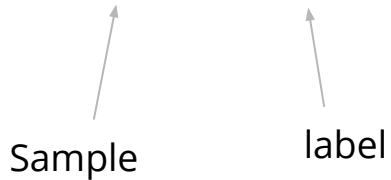
Supervised Learning

Opposite of classical programming

- Supervised learning:
 - Start with example input-output pairs (labeled data)

(image_1, duck), (image_2, duck),
(image_3, not duck)...

Sample label



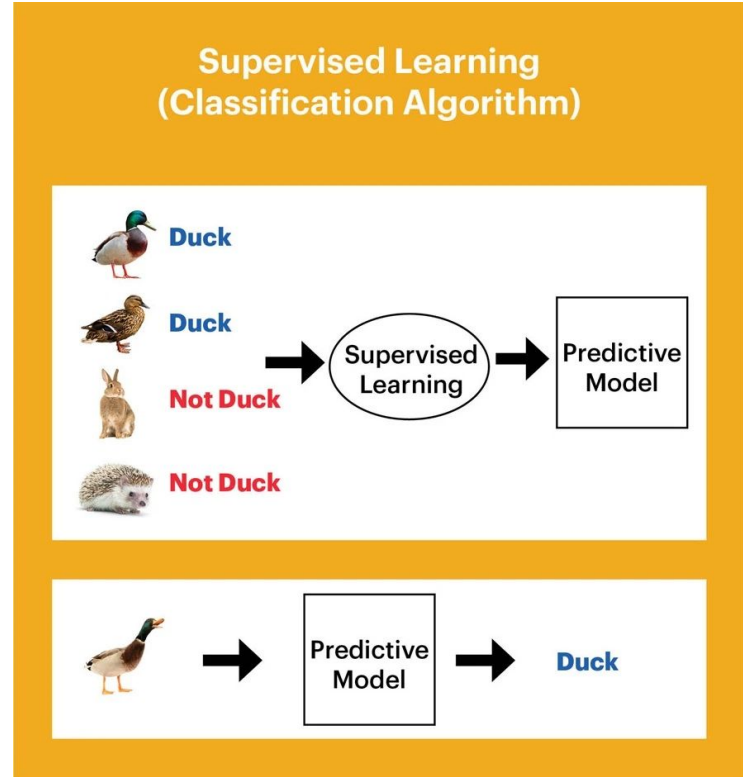
Supervised Learning

Opposite of classical programming

- Supervised learning:
 - Start with example input-output pairs (labeled data)
 - Use these to “learn” set of rules $f()$ that maps input to output

$f(\text{new_image}) = \text{duck}$

NOTE: To make more relevant, imagine pairs (face_image, identity)



Setting up a Supervised Learning Problem

1. Start with labeled data
 - a. Samples: x_1, x_2, x_3, \dots
 - b. Labels: y_1, y_2, y_3, \dots
 - c. Labeled samples: $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots$
2. Begin with an initial random model
 - a. Maps the x_1, x_2, \dots to **something random**: $f(x_1), f(x_2), \dots$
 - b. Ex. $f(\text{face}_1) = \text{Merkel}$ or $f(\text{face}_2) = \text{Merkel}$?
3. Define a “loss function”: number measuring how good/bad the predictions are.
 - a. Increase loss by a lot for each bad prediction, by a little (maybe zero) for a good prediction.



f
→ Merkel



f
→ Also Merkel?

f is the model. It's the map from input to prediction.

Setting up a Supervised Learning Problem

4. Gradually adjust the model \mathbf{f} to make the loss smaller and smaller (i.e. make better and better predictions)

Keyword: “Gradient descent”

5. Once the loss can't get any smaller, the best model \mathbf{f}_{best} has been found.

ALL SUPERVISED LEARNING MODELS ARE BUILT ON THIS TEMPLATE.



\mathbf{f}_{best}
→ Merkel



\mathbf{f}_{best}
→ Macron

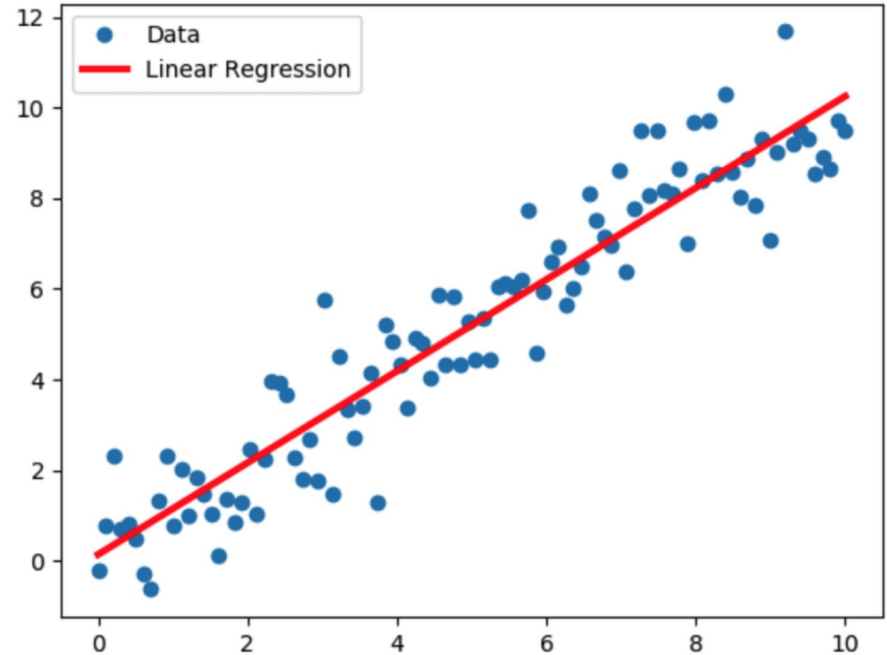
\mathbf{f} is the model. It's the map from input to prediction.

First Example: Linear Regression

- “Hello World” of machine learning.
- Linear Regression and slight generalizations are widespread tool.
 - See wiki page under ‘Applications’
- Contains all of the key ideas of supervised learning.

IDEA: Given a 2D-set of data points, find the “best-fit” line.

Uses: Make predictions (if your data is roughly linear) or explore relationship between two variables.



What do we need to understand this example?

1. High school algebra
2. To code it: some practice with numpy and matplotlib (for visualization)
3. Calculus helps, but isn't necessary

Reward: Practice with important python libraries, understanding of how supervised learning *really* works, understanding of linear regression

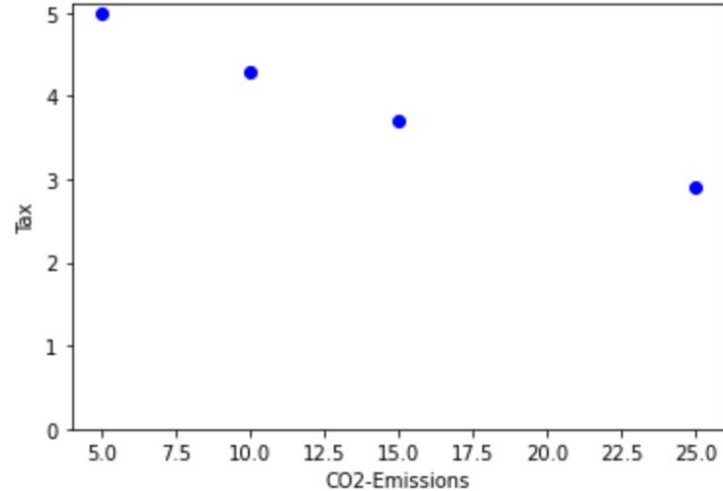
1. Start with labeled data

Example: Suppose we are interested in studying if CO2-emissions taxes are an effective way to decrease emissions.

Tax (\$/metric ton)	CO2-Emissions (metric tons)
5	5
10	4.3
15	3.7
25	2.9

samples

labels



Note: We might seem far away from face recognition now. I promise that understanding linear regression is a huge step toward understanding facial recognition.

2. Begin with initial random model.

x = emissions tax

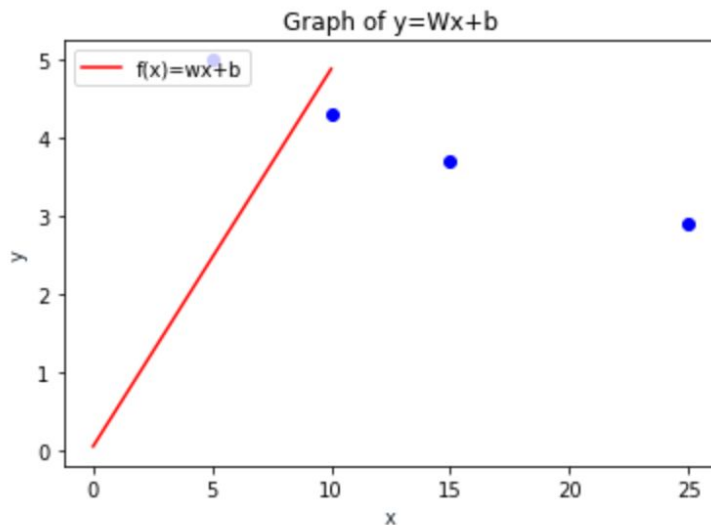
$$f_{ini}(x) = wx + b$$

w and b are called *weights*. In the beginning, they are chosen randomly.

$f_{ini}(x)$ is the initial model.

Essentially, start with a random line.

Ex. $w = 0.48, b = 0.05$



3. Define a Loss Function

Mean-squared error:

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

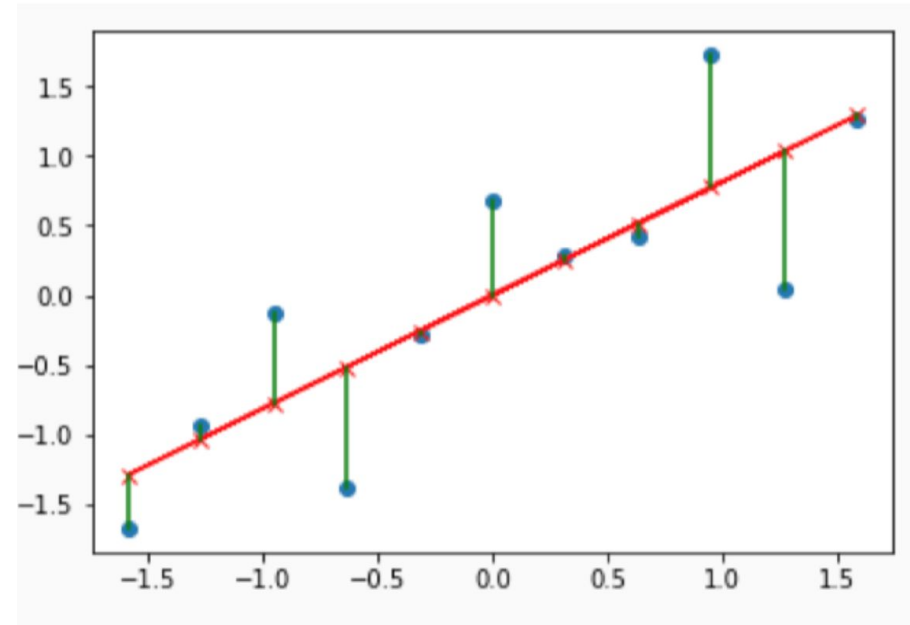
In this example $n = 4$ is the number of data points.

x_i are the different tax levels.

y_i are the corresponding emissions levels.

$f(x_i) = wx_i + b$ is the model's prediction.

The loss measures how far the data points are on average from the current line. The best line makes this as small as possible.



The loss is the average of the squared lengths of the vertical green lines (note: the above picture is from a different example).

In our example, with $w = 0.48$ and $b = 0.05$, the initial loss = 26.24.

Loss Function

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

This symbol \sum means “take the sum.” In our example:

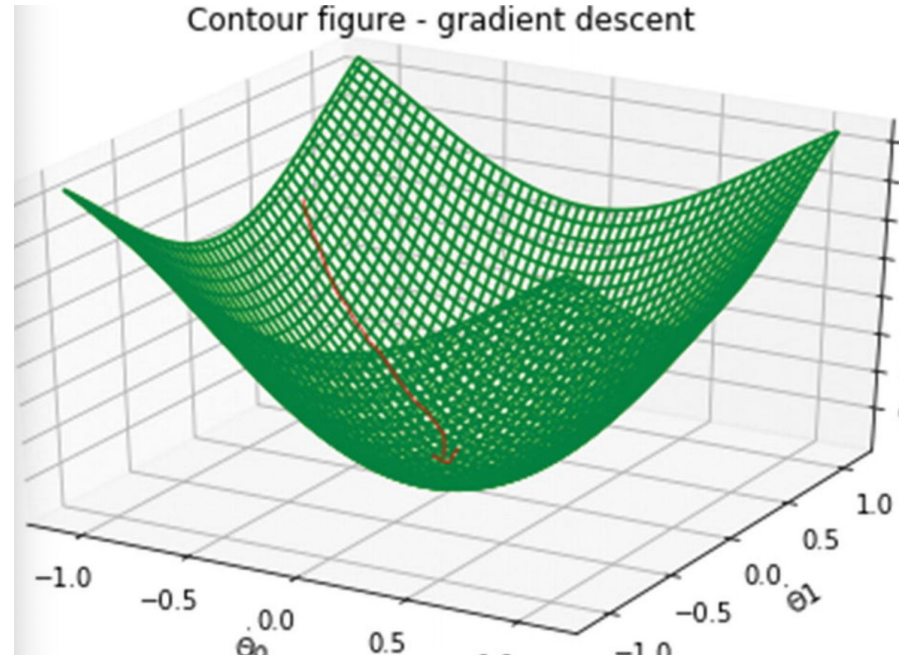
$$Loss = \frac{1}{4} \left((f(x_1) - y_1)^2 + (f(x_2) - y_2)^2 \right. \\ \left. + (f(x_3) - y_3)^2 + (f(x_4) - y_4)^2 \right)$$

4. Gradually Adjust Model to Decrease Loss

The graph of the loss function looks like this. A point on the horizontal plane corresponds to a fixed w and b , and the height of the graph above that point is the corresponding loss.

We want to reach the valley, where the loss is as small as possible. The arrow starts at our initial random model.

The mathematical method for walking downhill into the valley is called *gradient descent*.

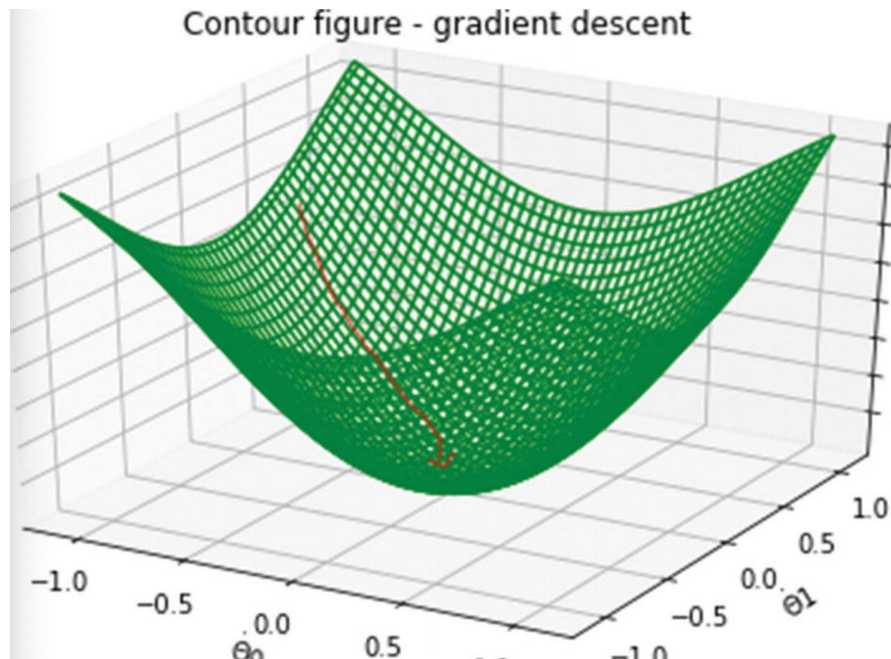


Gradient Descent (Hardest Part)

To reach the valley, iterate the following process:

Take a small step in the direction of steepest descent.

This should get us into the valley quickly.



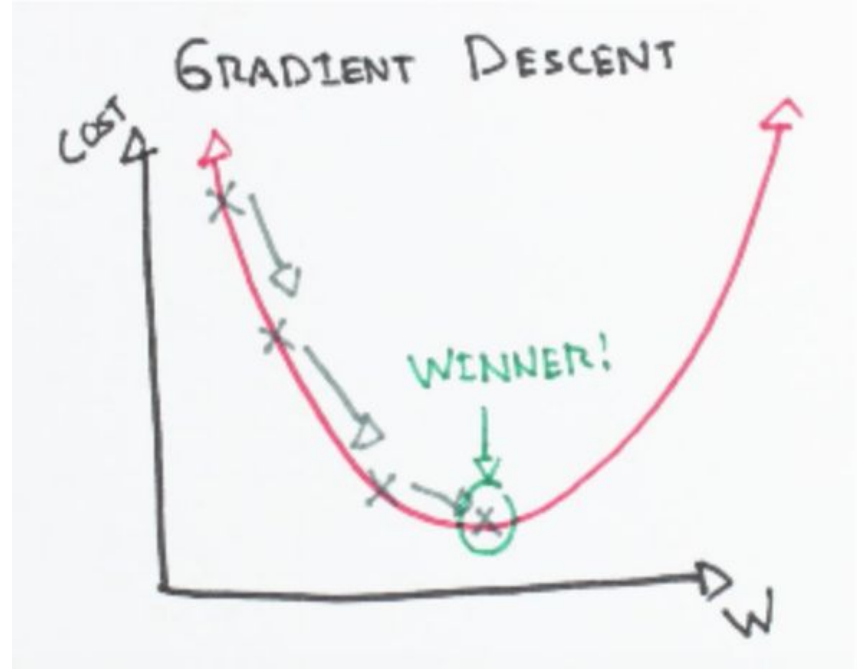
Gradient Descent

The mathematical version of “taking a small step in the direction of steepest descent” is to update the weights as follows:

$$w_{new} = w_{old} - \alpha \cdot \frac{2}{n} \sum_{i=1}^n (f(x_i) - y_i) \cdot x_i$$

$$b_{new} = b_{old} - \alpha \cdot \frac{2}{n} \sum_{i=1}^n (f(x_i) - y_i)$$

α is the length of our step. Usually it's a small number like $\alpha = 0.1$. Called the *learning rate*.



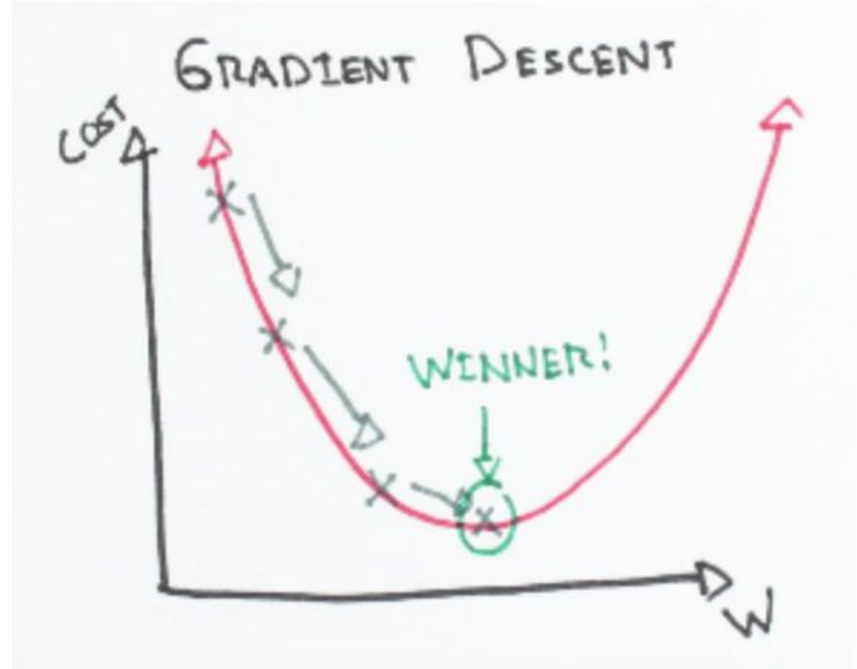
Gradient Descent

The mathematical version of “taking a small step in the direction of steepest descent” is to update the weights as follows:

$$w_{new} = w_{old} - \alpha \cdot \frac{2}{n} \sum_{i=1}^n (f(x_i) - y_i) \cdot x_i$$

$$b_{new} = b_{old} - \alpha \cdot \frac{2}{n} \sum_{i=1}^n (f(x_i) - y_i)$$

α is the length of our step. Usually it's a small number like $\alpha = 0.1$. Called the *learning rate*.



5. Best Model when loss is minimal

$f_{\text{final}}(x) = -0.1x + 5.4$, i.e.,

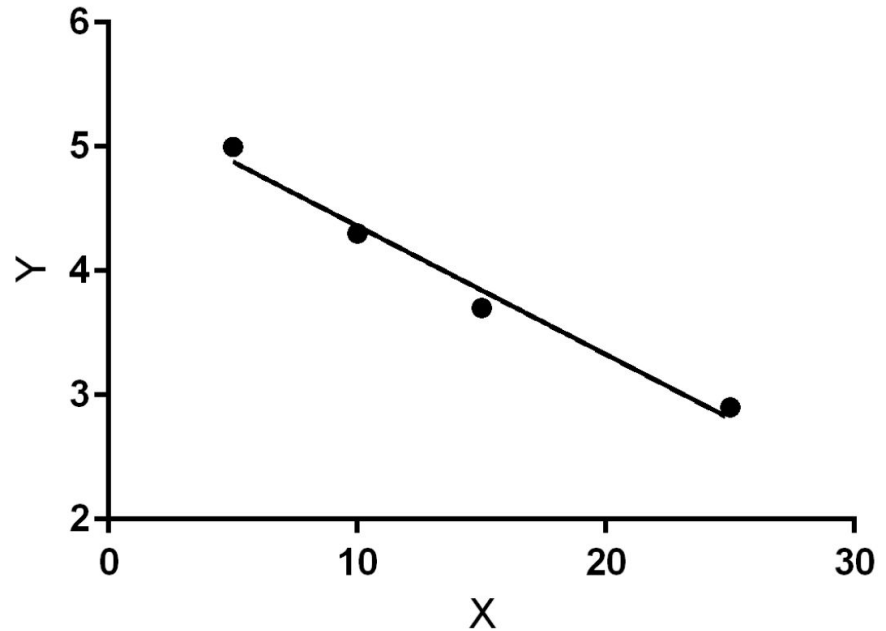
$w = -0.1$, $b = 5.4$

See

<https://towardsdatascience.com/linear-regression-using-gradient-descent-97a6c8700931>

for an explanation on how to code this.

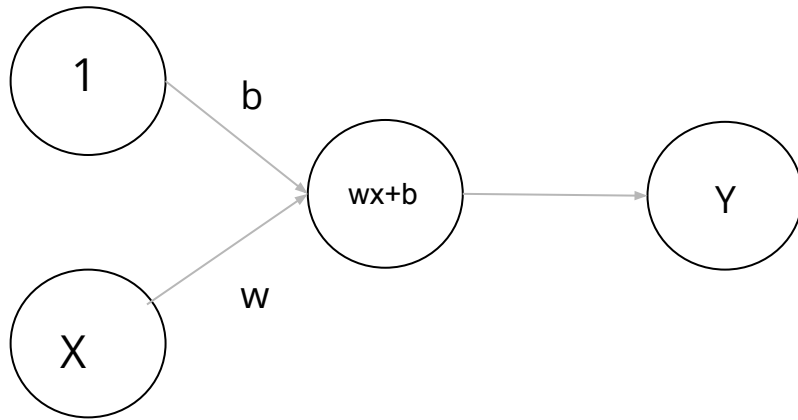
Will also include as an assignment :)



Notes:

- The reason for doing linear regression in this way was to develop an intuition for how supervised learning works in general.
- In practice you can use excel, R or other tools to do linear regression.
- There's a lot more to know about regression.
- Once you have your model, you can make predictions. In our example:
 $f(20\$) = -0.1(20) + 5.4 = 3.4$ metric tons of CO₂.
- Roughly speaking, we say the emissions tax “influences” emissions if $w \neq 0$.

P.S. That was a neural network :)



6. Problems

Labelled data

- Labelling data (ex. Tagging untagged photos) requires many hours of human labor
- Often outsourced
- Sometimes labelling requires specialist expertise (ex. Identifying tumor types from medical images)

6. Problems

Interpretability/Transparency

- In linear regression, clear what information from the data the model is using to make predictions.
- With deep NNs used in face recog., not clear at all, including to human who coded it
- Imagine company claims to be able to identify criminals, say using labeled dataset of faces marked “criminal” or “not criminal”
 - If most of the faces in criminal dataset were black, model uses this correlation. Assumes “black” => “criminal”
 - Models don’t “predict” in the sense that they find the causes for outcomes, just find correlations in training data
 - Such a model could reinforce itself (by using it to catch criminals)
- There is research toward adding a penalty for “unfairness” while training. Requires mathematical definition of “fairness.”
- Exacerbated by fact that training data is kept secret

7. Resources

Resources for learning/using ML (in order of difficulty):

- Google AutoML: Train vision and language models without any coding. System tries to create best deep learning model for the job
- Fast.ai: bottom up MOOC focusing on building your own projects as fast as possible, slowly add theory
- Deeplearning.ai: opposite approach
 - Tensorflow in practice specialization
- CS231: the best course I know for *really* understanding what's going on
 - Slowly build up to coding CNNs for image recog. In python from scratch
 - Deep investigation of regularization, hyperparam optimization etc.
- Deep Learning Book (Goodfellow et. al.) - for when you wanna be an expert/a researcher

The barrier to entry is not as crazy as you might think! With couple months of coding/python and high school math, you can already get started.

7. Resources

- Mozilla Common Voice
 - <https://voice.mozilla.org/en>
 - Donate your voice/validate donated recordings
 - Provide open-source voice database
 - Allow development of voice systems by more than big tech companies (also open-source those)
- Help collect/annotate data (esp. in your specialization)
- Design platforms (ex. apps) to aid data collection/annotation
 - Some groups use apps to encourage data collection

Discussion Questions

What are other potential uses of image recognition (or supervised learning in general) toward activist goals?

What do we gain or risk in exploiting the weaknesses in surveillance algorithms? Does it matter when they can be improved so quickly?

Is some amount of public surveillance OK? Is it inevitable?

How can we start/contribute to machine learning based projects without knowing machine learning?