



Getting started with Machine Learning

Data Engineering on Google Cloud Platform

Agenda

What is Machine Learning?

Playing with ML

Effective ML

Creating ML datasets + Lab

Machine Learning: Way to derive insights from data



data

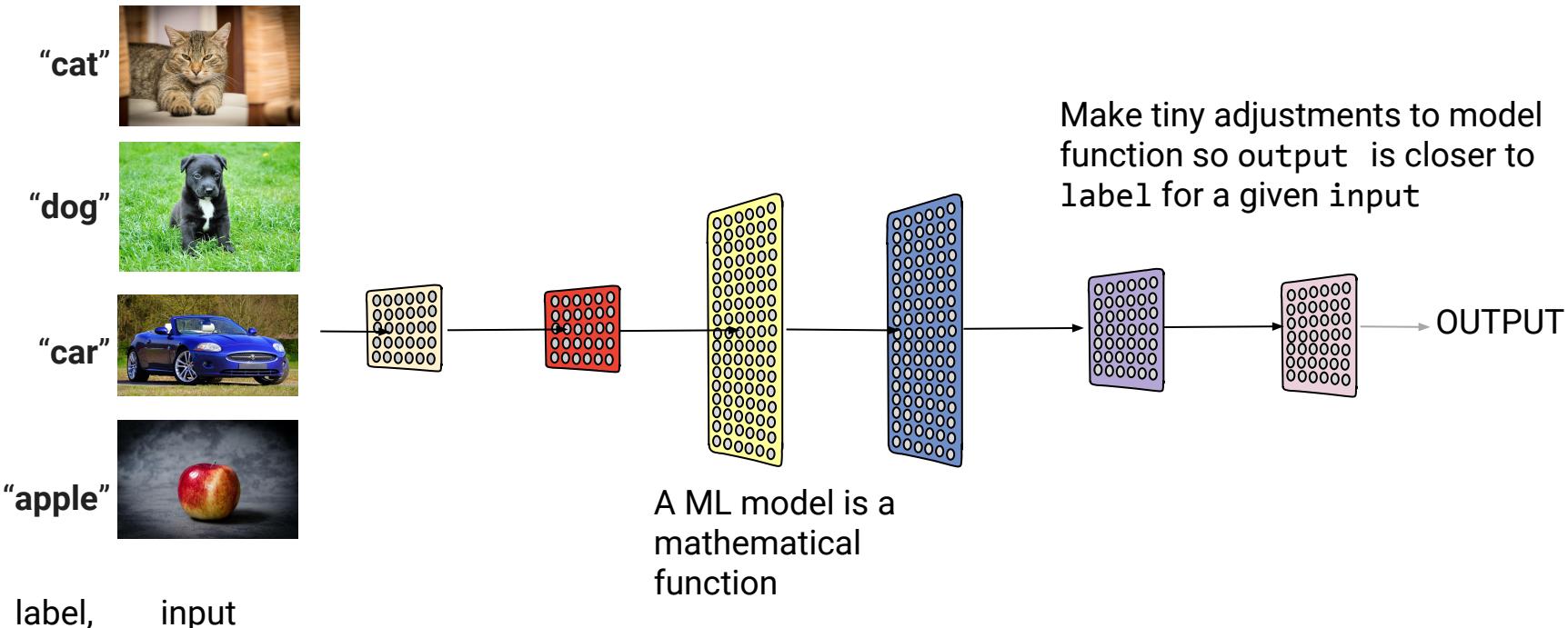


algorithm

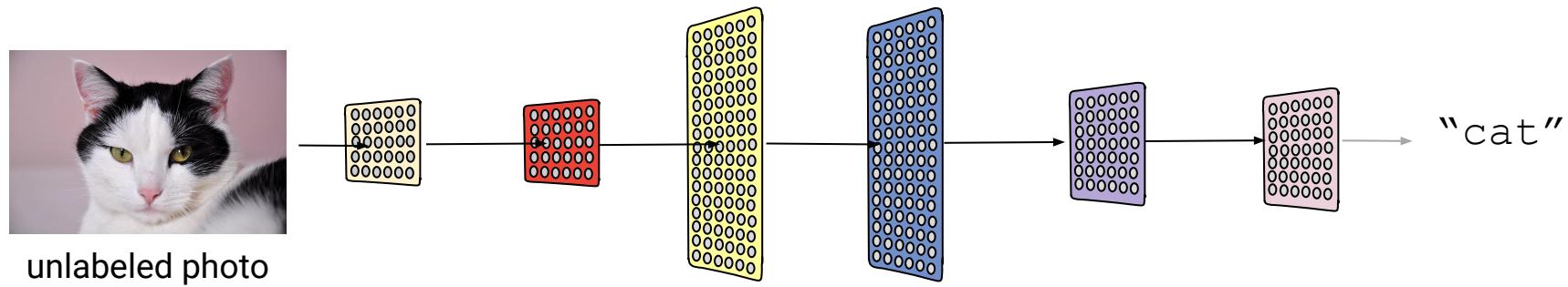


insight

Step 1: Train a ML Model with examples



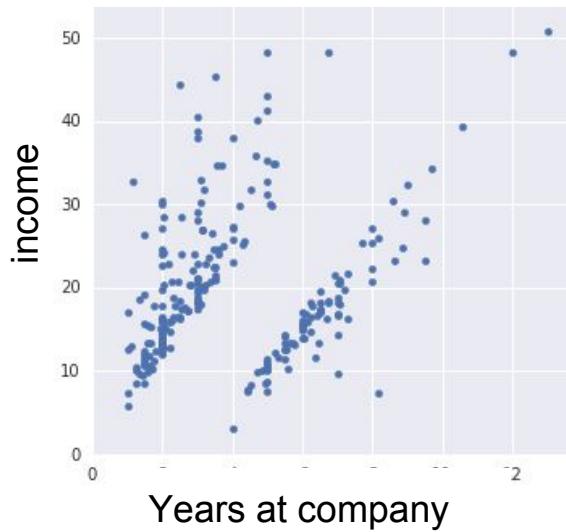
Step 2: Predict with a trained model



Do Now: In your own words, write down definitions for these ML terms

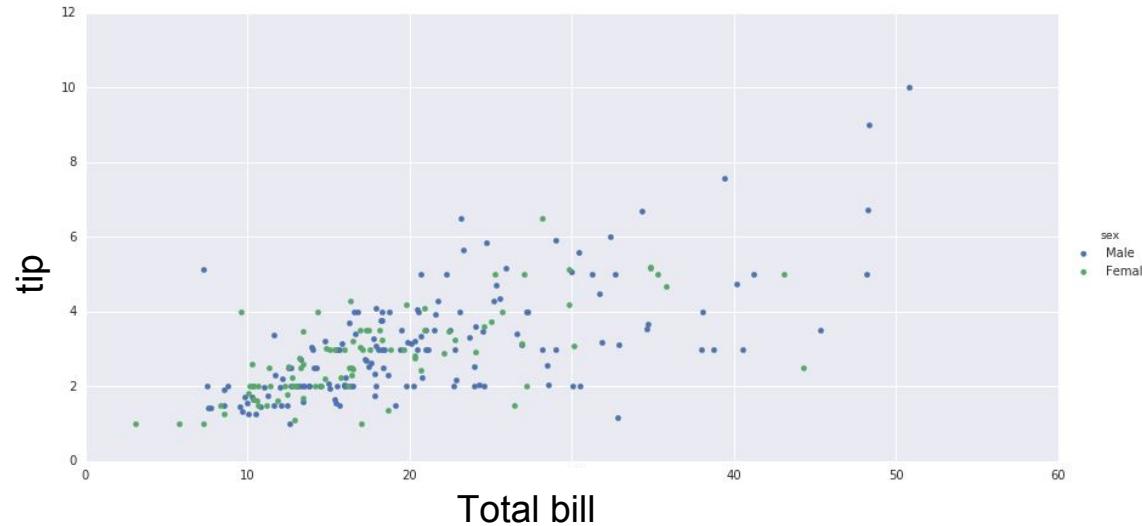
Term	Meaning
Label	
Input	
Example	
Model	
Training	
Prediction	

Machine Learning use cases



Clustering

Is this employee on the
“fast-track” or not?



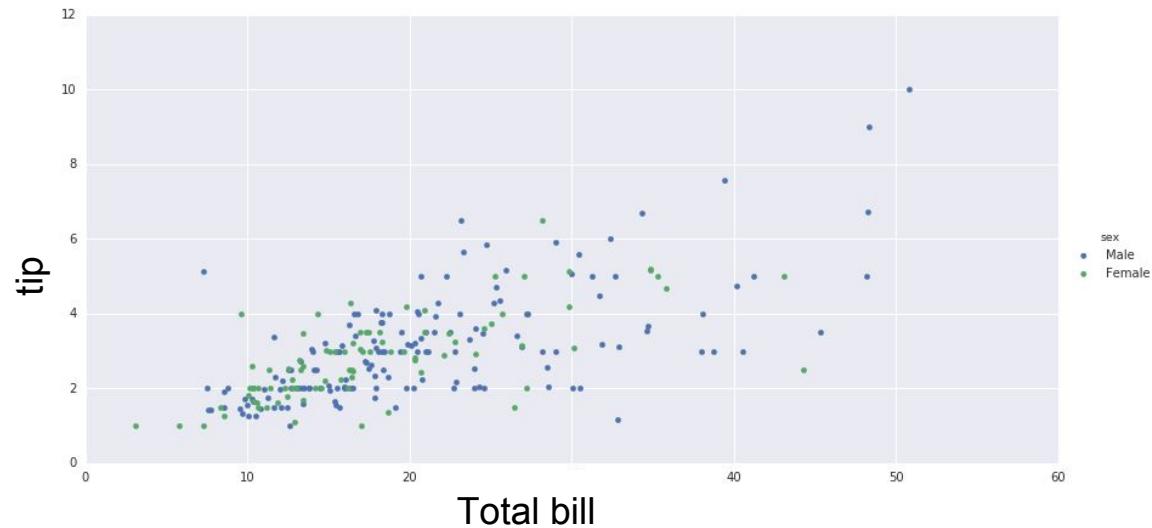
Regression

Predict the tip amount

Classification

Predict the gender of the
customer

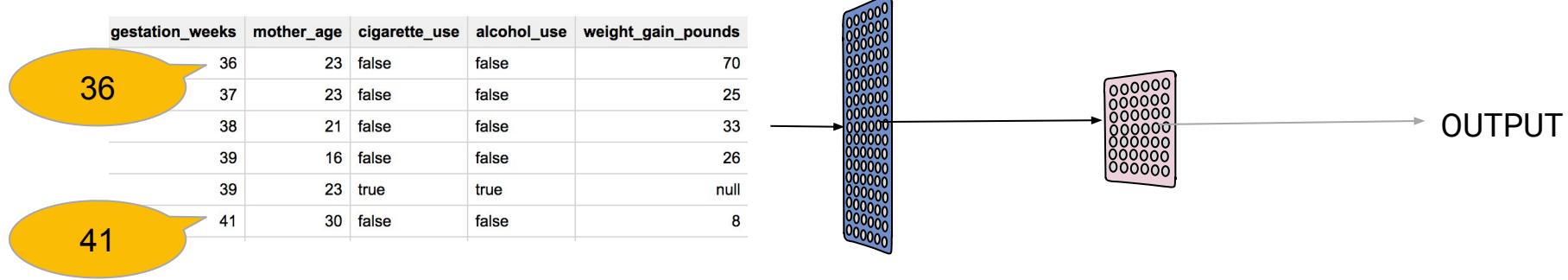
We'll focus on supervised learning



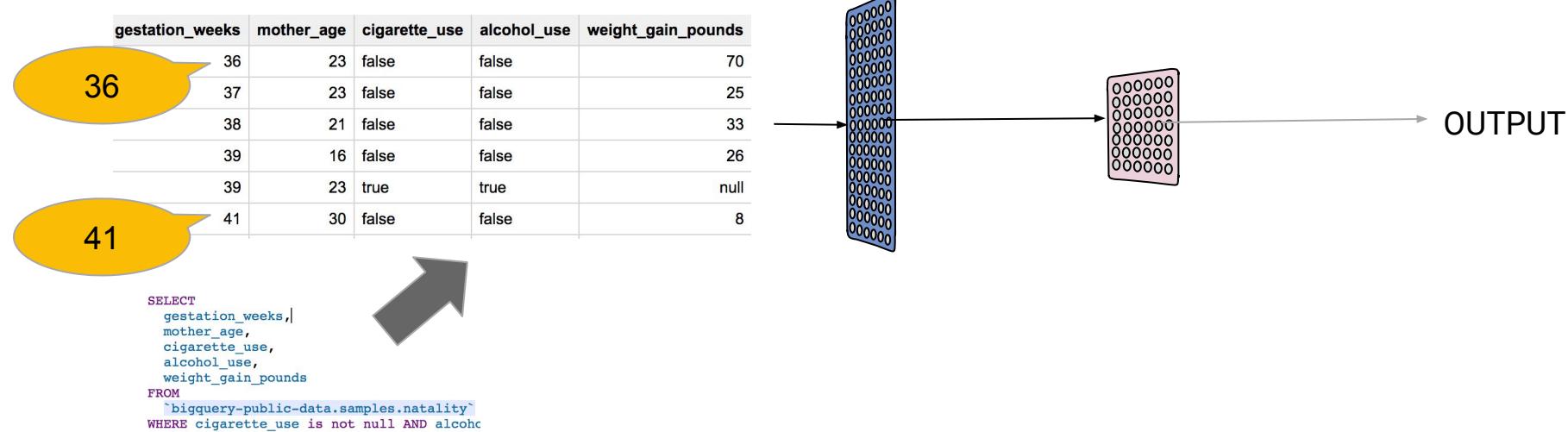
Regression
Predict the tip amount

Classification
Predict the gender of the customer

This regression model predicts a continuous number

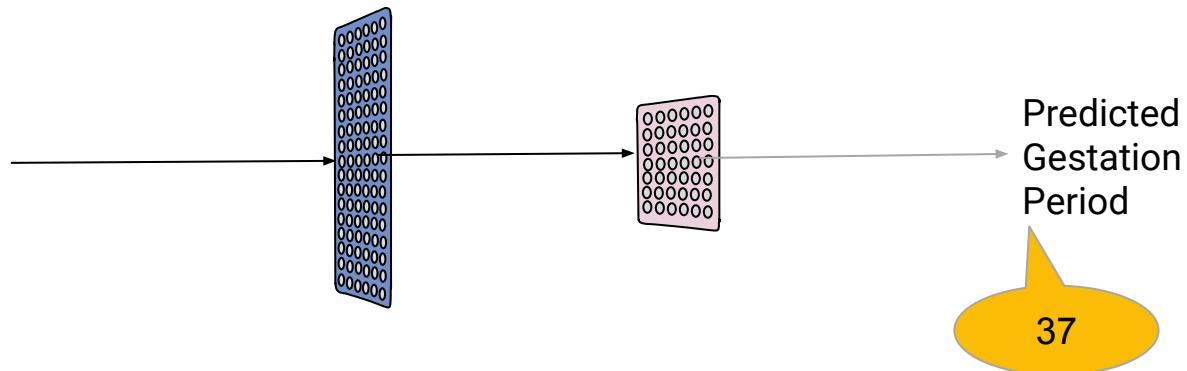


A common source of structured data for ML training is your data warehouse

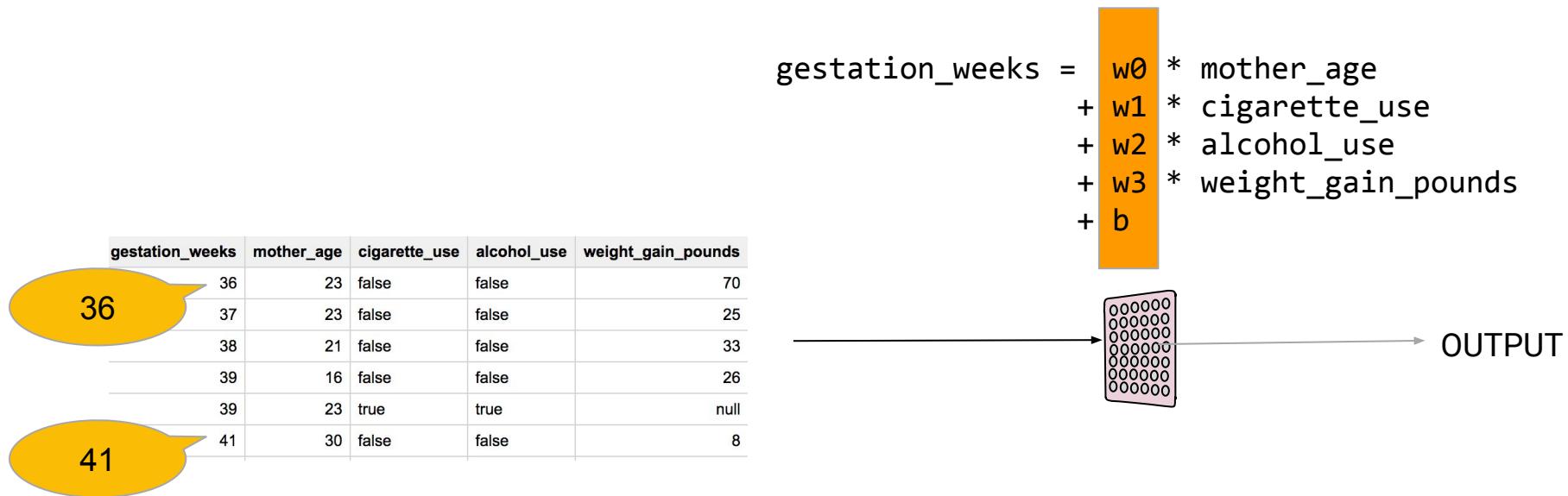


The model is fed information collected in real-time, and used for prediction

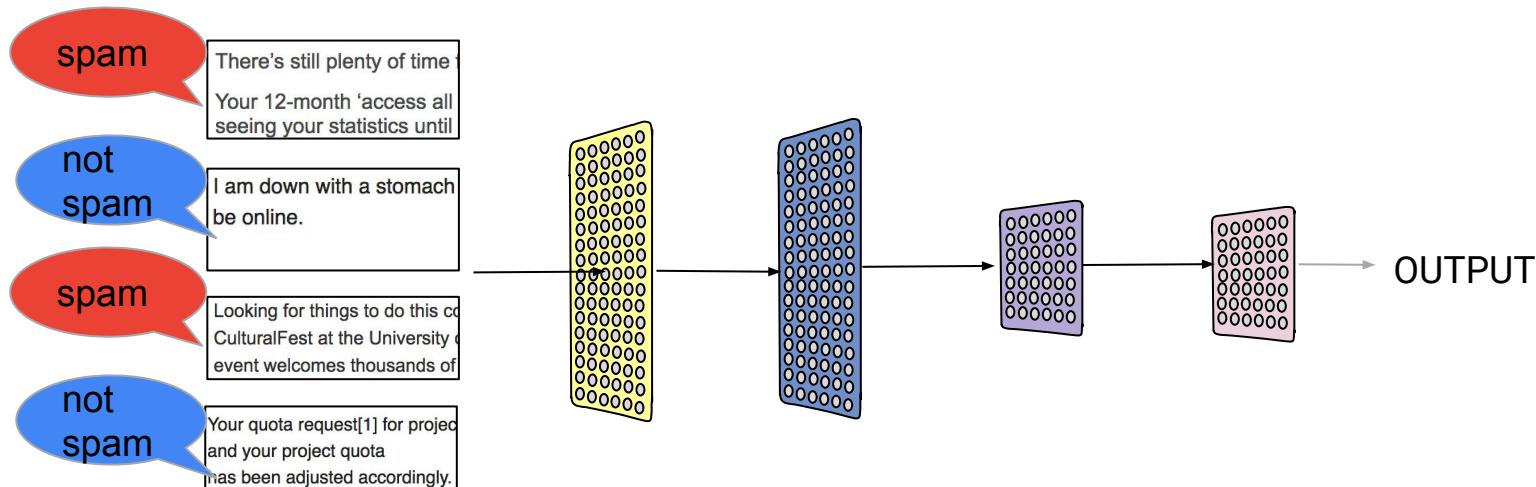
```
{  
  "mother_age": 23,  
  "cigarette_use": false,  
  "alcohol_use": true,  
  "weight_gain_pounds": 35  
}
```



The model may even have only one layer



A classification model can be used to detect whether email is spam or not



The input here is text

The inputs for unstructured data are still ultimately just numbers

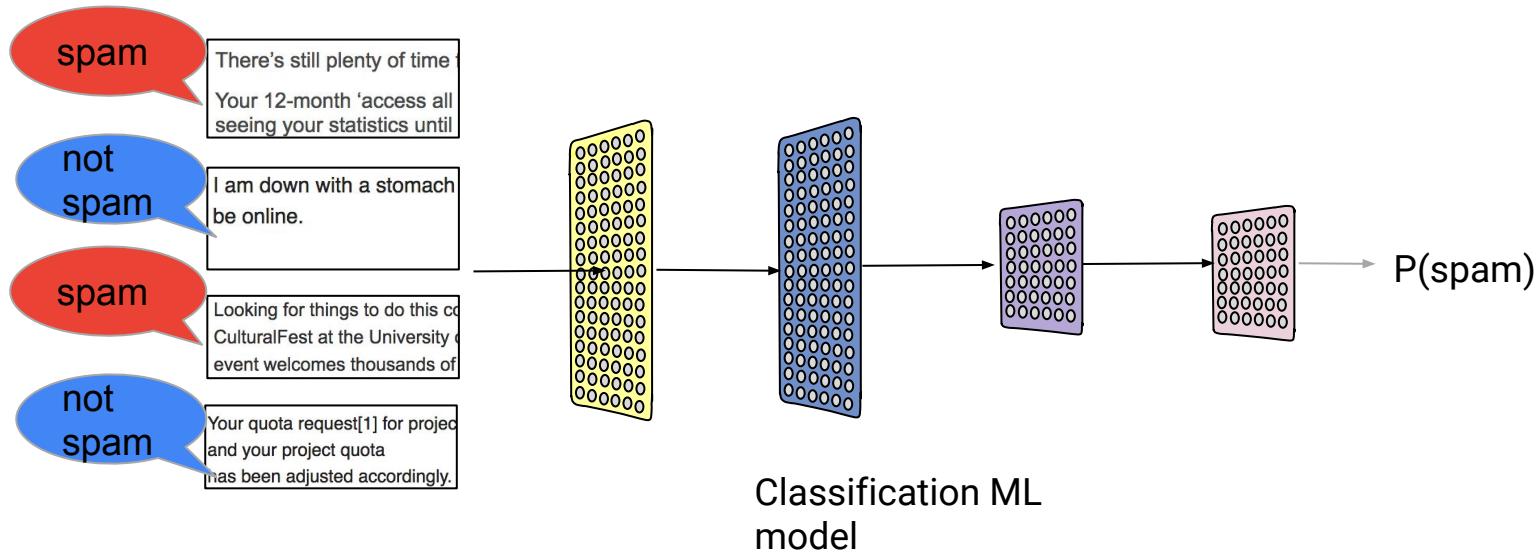


N-dimensional array of pixel values

There's still plenty of time
Your 12-month 'access all
seeing your statistics until

Each word is mapped to a vector
e.g., "the" could be [0 4 5 0 3 4]
Coming up with an appropriate vector for a word
is itself a machine learning problem

The output of the model might be the probability that the email is spam



Machine Learning used in lots of industries



Manufacturing

- Predictive maintenance or condition monitoring
- Warranty reserve estimation
- Propensity to buy
- Demand forecasting
- Process optimization
- Telematics



Retail

- Predictive inventory planning
- Recommendation engines
- Upsell and cross-channel marketing
- Market segmentation and targeting
- Customer ROI and lifetime value



Healthcare and Life Sciences

- Alerts and diagnostics from real-time patient data
- Disease identification and risk satisfaction
- Patient triage optimization
- Proactive health management
- Healthcare provider sentiment analysis



Travel and Hospitality

- Aircraft scheduling
- Dynamic pricing
- Social media—consumer feedback and interaction analysis
- Customer complaint resolution
- Traffic patterns and congestion management



Financial Services

- Risk analytics and regulation
- Customer Segmentation
- Cross-selling and upselling
- Sales and marketing campaign management
- Credit worthiness evaluation



Energy, Feedstock and Utilities

- Power usage analytics
- Seismic data processing
- Carbon emissions and trading
- Customer-specific pricing
- Smart grid management
- Energy demand and supply optimization

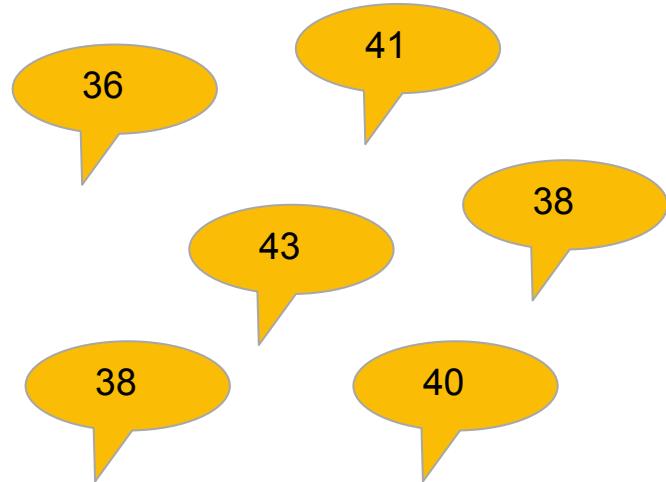
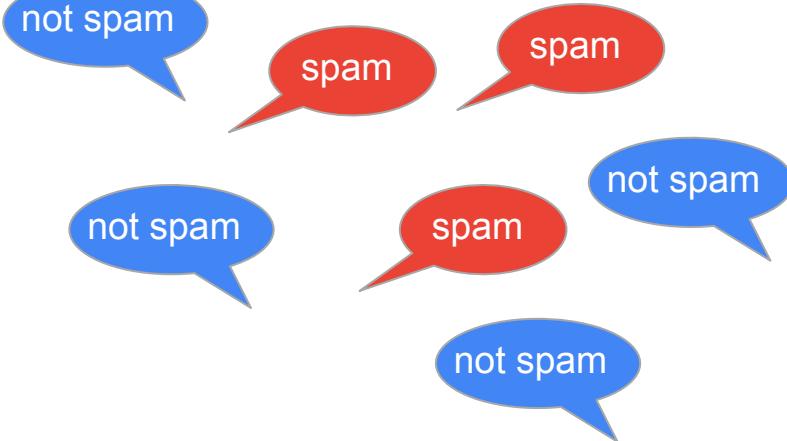
Do Now: Pick 5 use cases from previous slide and fill out this table

Use case	Label	Input(s)	Classification or regression?

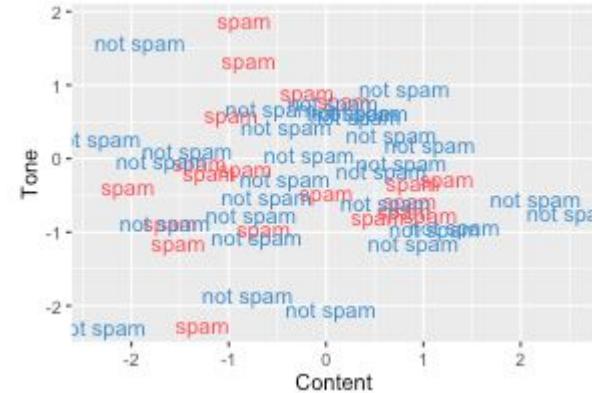
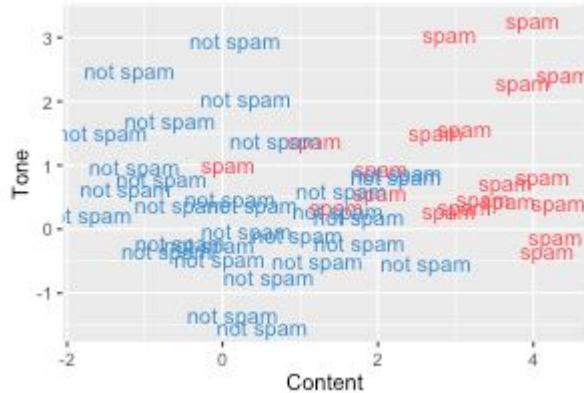
Agenda

Playing with ML

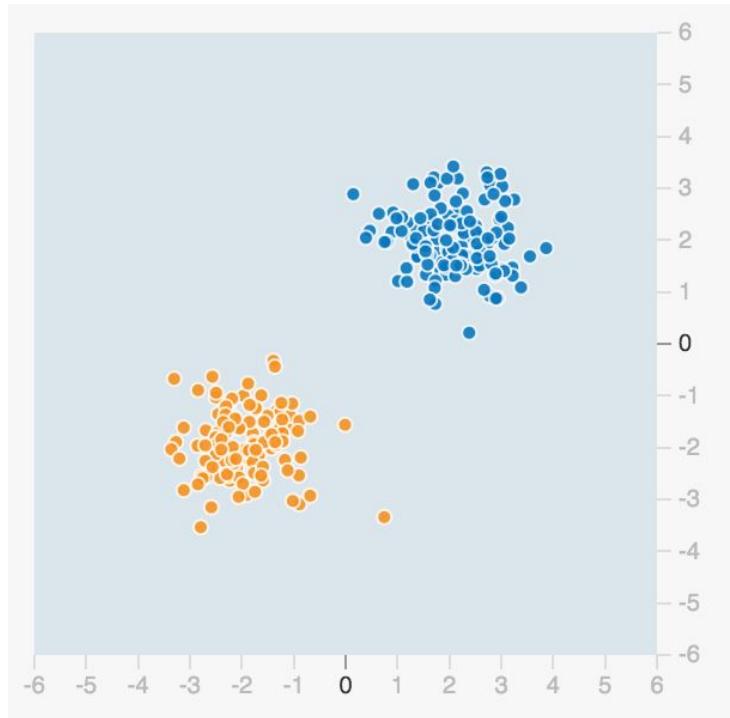
Machine Learning is an approach to making many similar decisions based on data



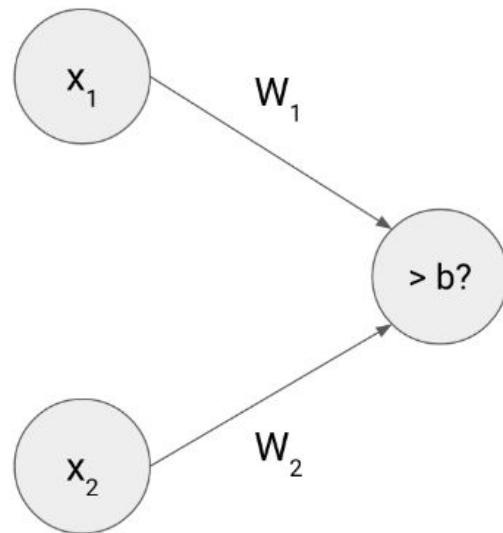
ML = Pattern recognition from examples



How do you classify these points?



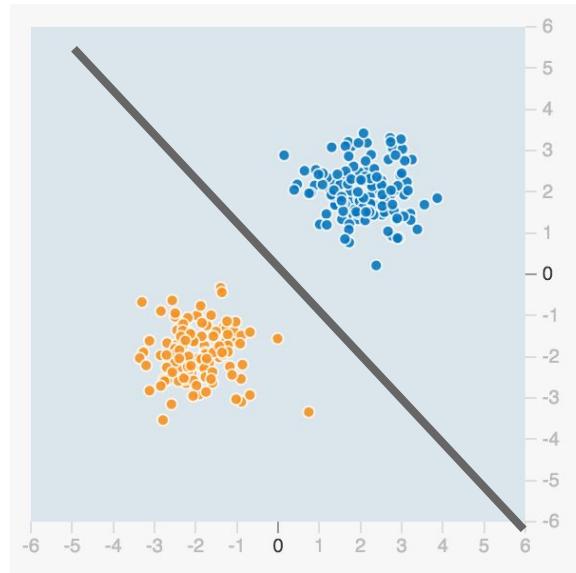
We could create a ML model consisting of a single neuron



$$w_1x_1 + w_2x_2 > b$$

Graphically, that translates to: find a line
that separates the two sets of points

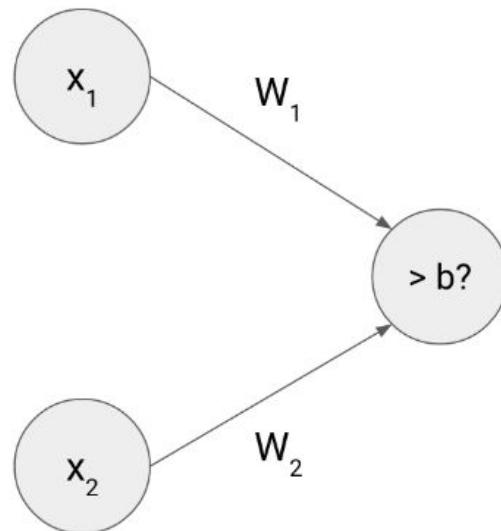
bias
(threshold)



$$w_1x_1 + w_2x_2 > b$$

weights

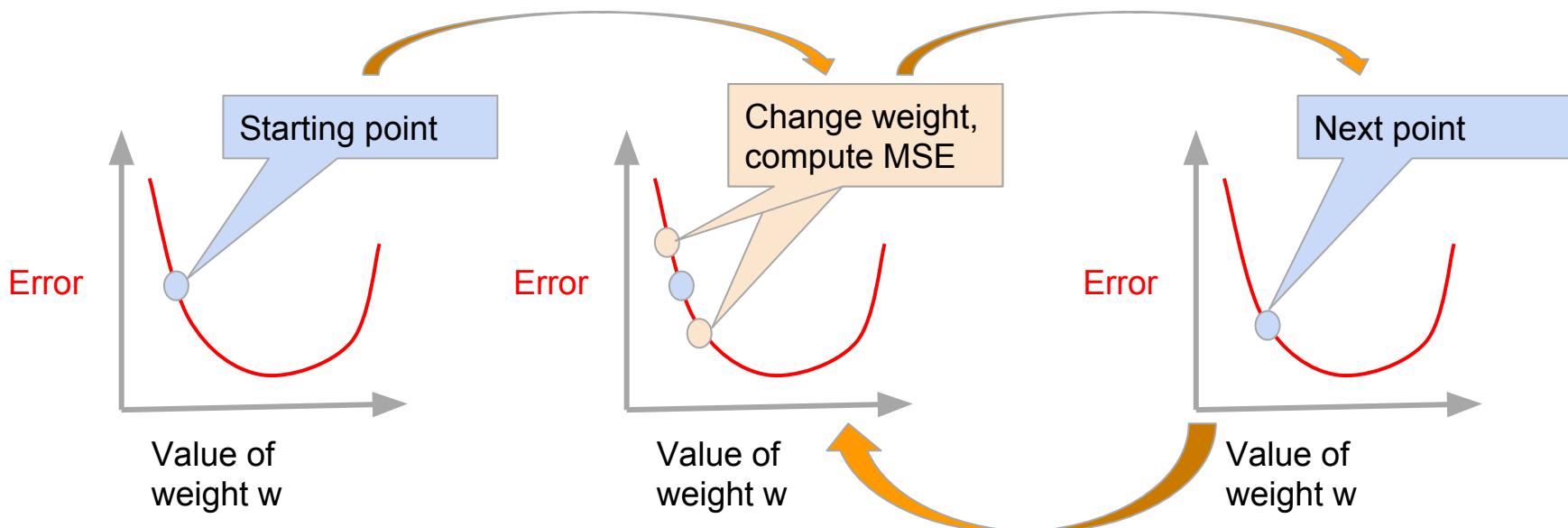
We can use gradient descent to find the best weights and bias



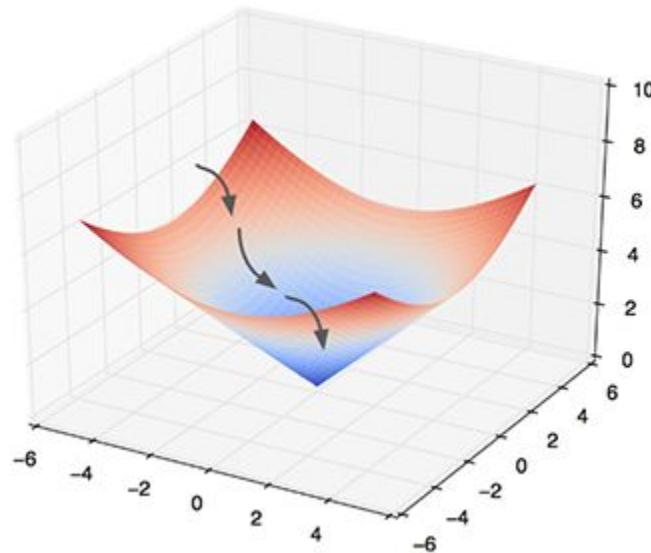
$$w_1x_1 + w_2x_2 > b$$

The computer tries to find the best **parameters**

Recompute error after each batch of examples (not full dataset)

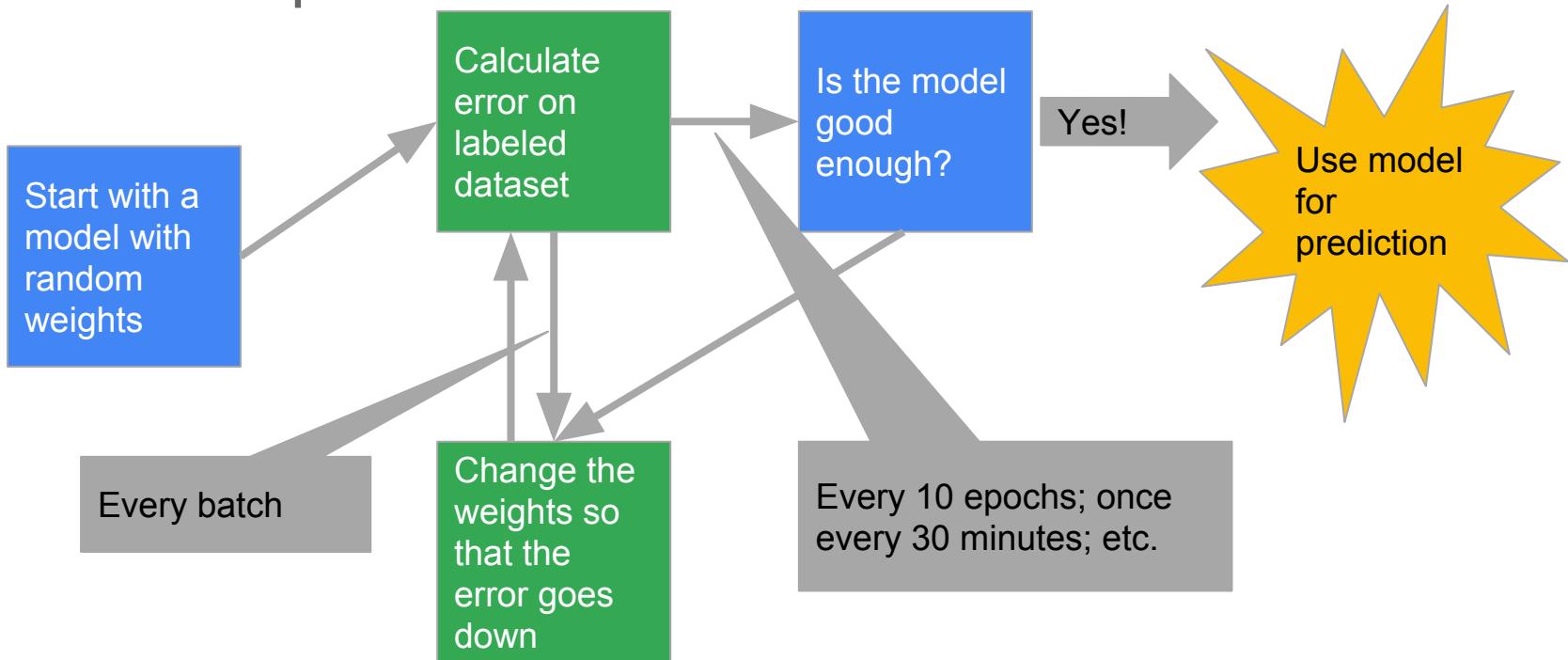


Gradient descent is used to find the best parameters

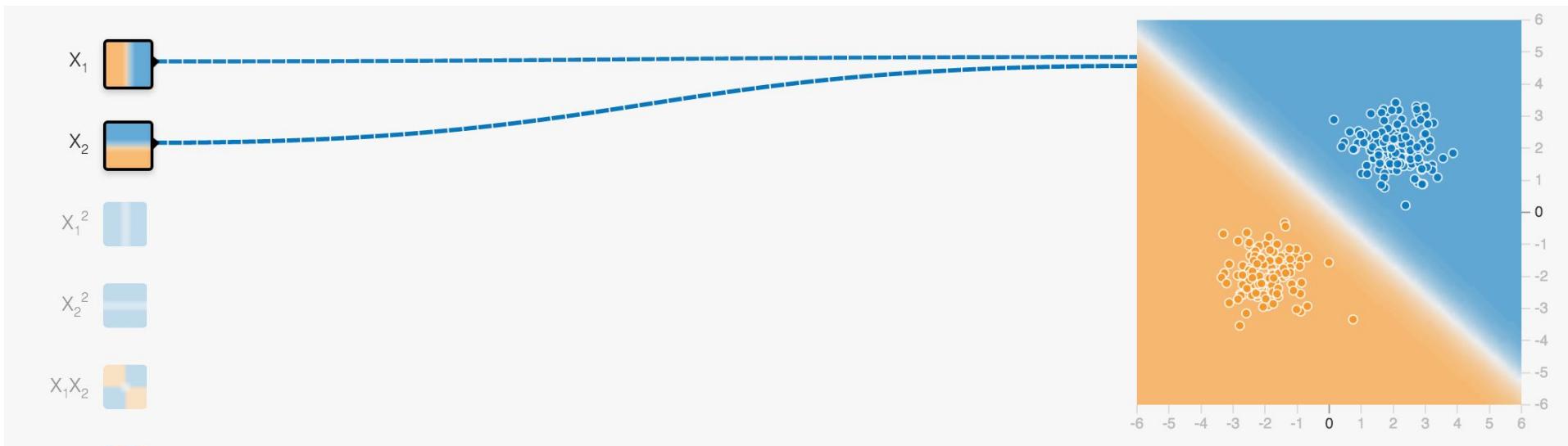


	-5	-4	-3	-2	-1	0	1	2	3	4	5
5	50	41	34	29	26	25	26	29	34	41	50
4	41	32	25	20	17	16	17	20	25	32	41
3	34	25	18	13	10	9	10	13	18	25	34
2	29	20	13	8	5	4	5	8	13	20	29
1	26	17	10	5	2	1	2	5	10	17	26
0	25	16	9	1	0	1	4	9	16	25	
-1	26	17	10	5	2	1	2	5	10	17	26
-2	29	20	13	8	5	4	5	8	13	20	29
-3	34	25	18	13	10	9	10	13	18	25	34
-4	41	32	25	20	17	16	17	20	25	32	41
-5	50	41	34	29	26	25	26	29	34	41	50

Occasionally, evaluate model to decide whether to stop



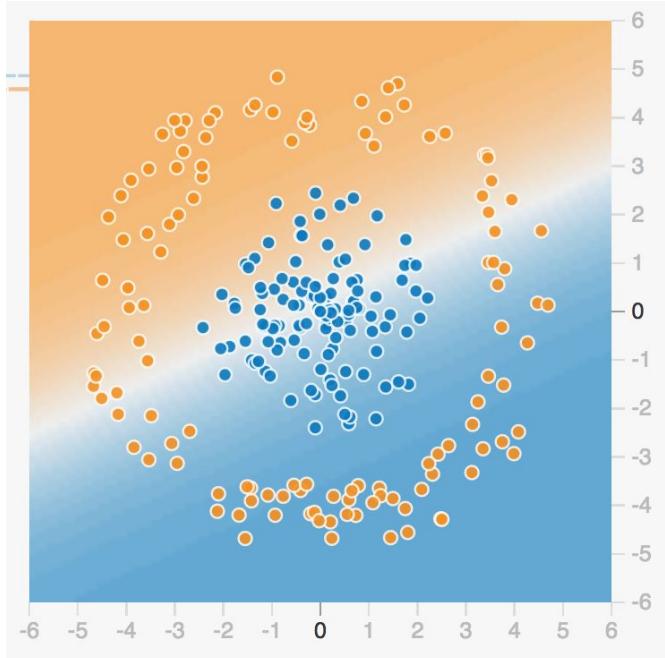
Do Now: <http://goo.gl/5aZjBF>



Do Now: In your own words, write down definitions for these ML terms

Term	Meaning
Weights	
Batch size	
Epoch	
Gradient descent	
Evaluation	
Training	

Do Now: Can you use a single line to separate these? What do you have to do?



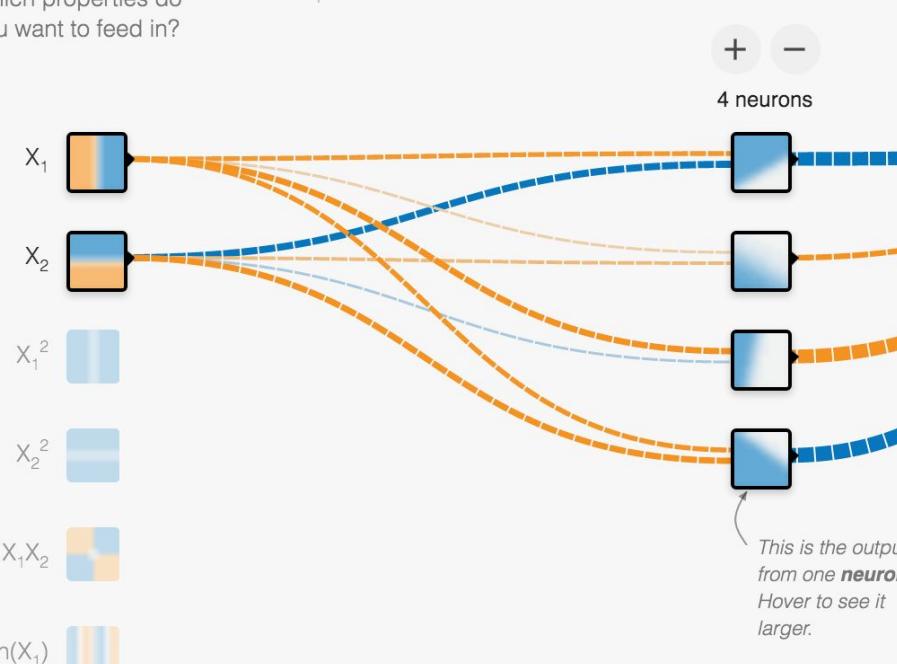
<http://goo.gl/v7qM4Q>

More neurons \Rightarrow more input combinations (features)

INPUT

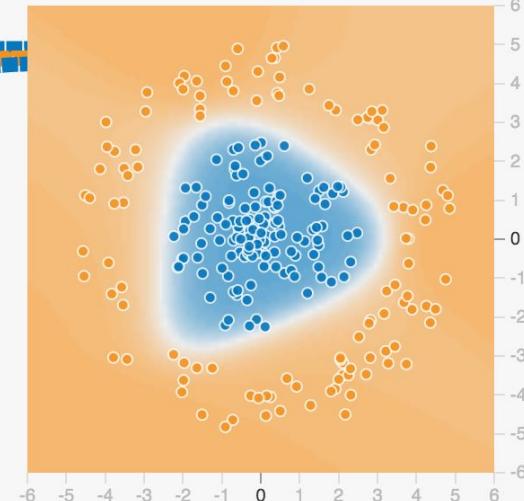
Which properties do you want to feed in?

+ - 1 HIDDEN LAYER

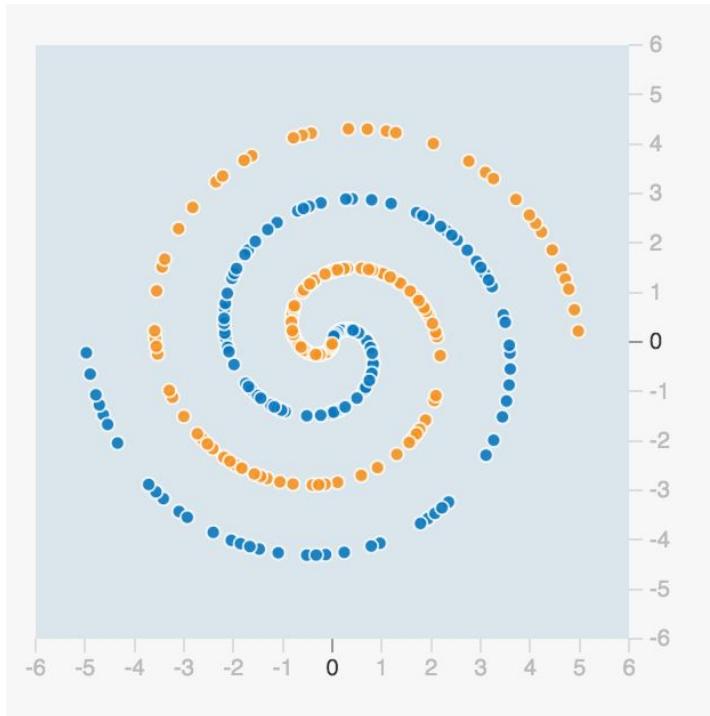


OUTPUT

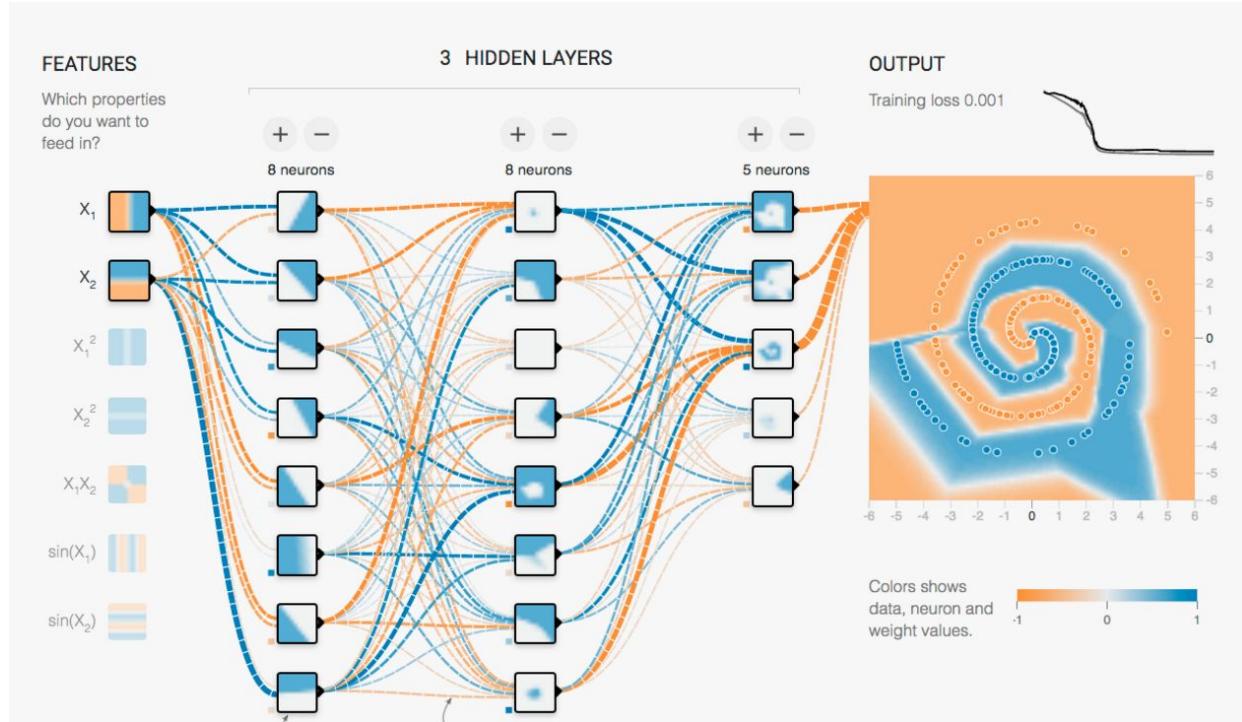
Test loss 0.026
Training loss 0.024



How about this? Will a set of lines work?

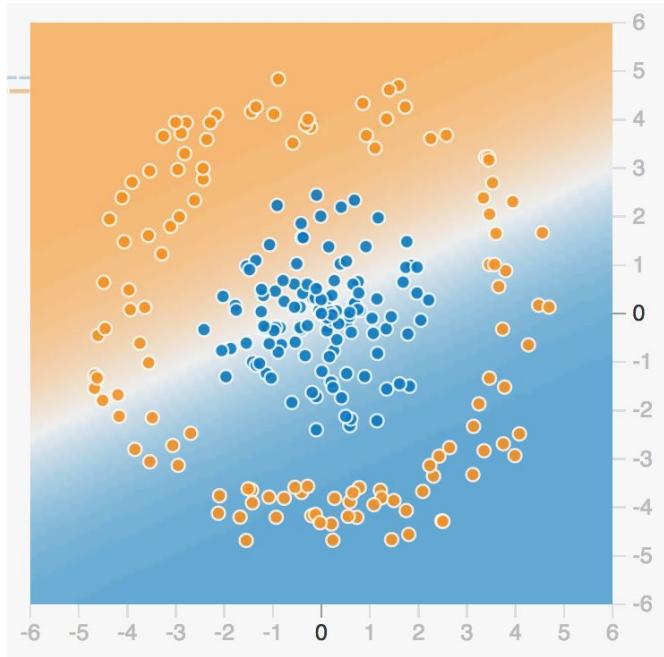


More hidden layers \Rightarrow more hierarchies of features



<http://goo.gl/fymPMI>

Do Now: Can you use a single line to separate these without adding layers?

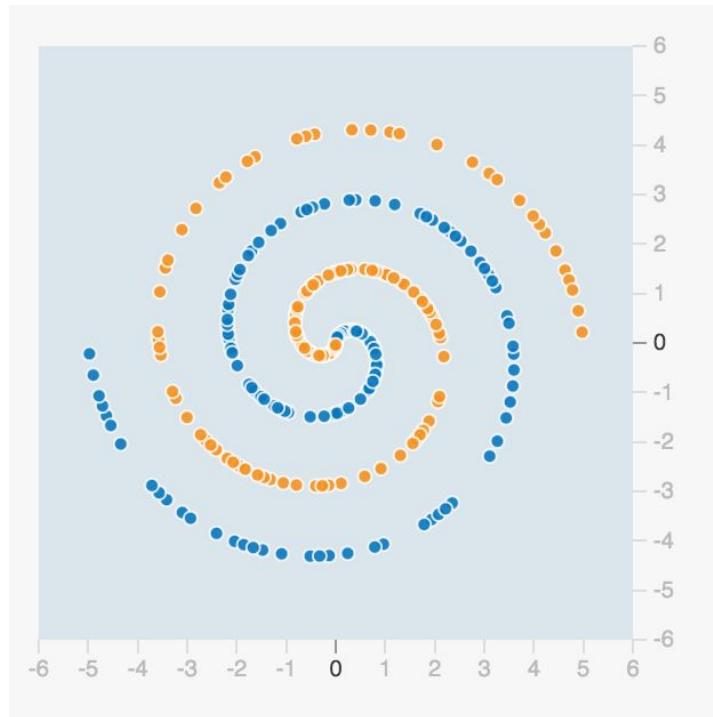


<http://goo.gl/v7qM4Q>

Engineer some extra features



These features help in the spiral case too...



<http://goo.gl/jdvKga>

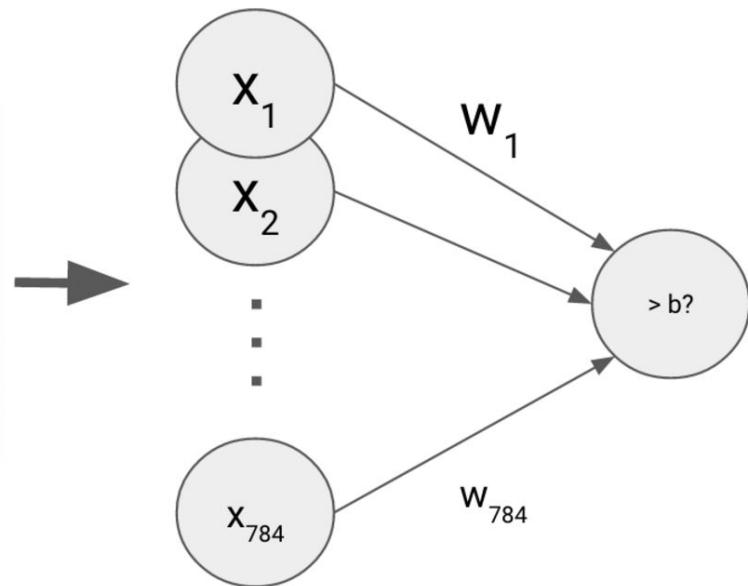
Do Now: In your own words, write down definitions for these ML terms

Term	Meaning
Neurons	
Hidden layer	
Inputs	
Features	
Feature Engineering	

What about images? How does it work?



Each pixel value is an input

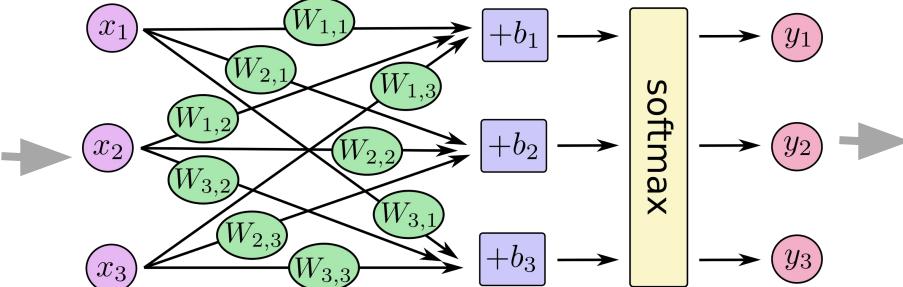
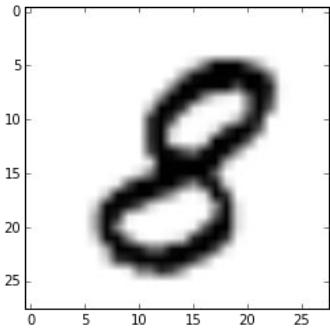


28 x 28 gray scale image =
784 numbers

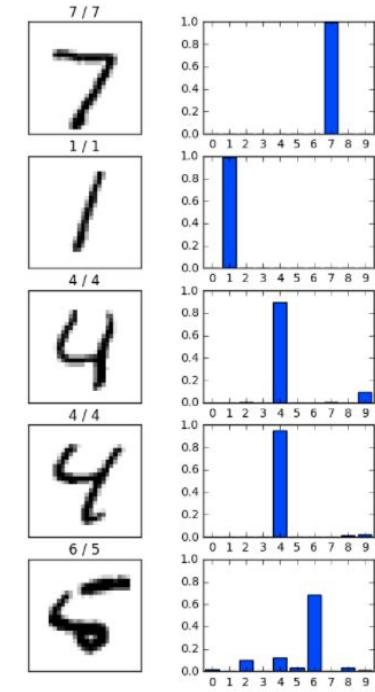
$$\sum_i^n w_i x_i > b$$

The softmax helps deal with multiple labels

input vector
(pixel data)



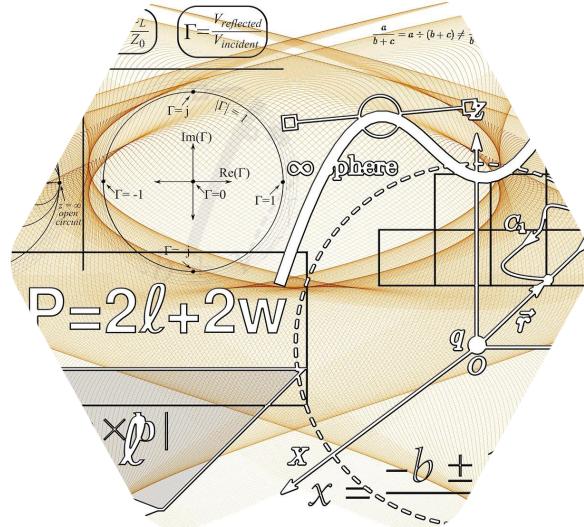
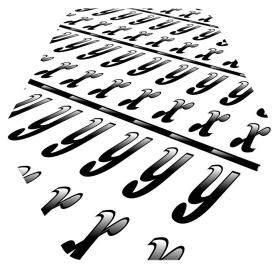
output vector
(probability)



Agenda

Effective ML

The popular imagination of what ML is



Lots of data

Complex mathematics in multidimensional spaces

Magical results

In reality, ML is...



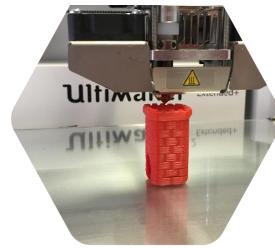
Collect
data



Organize
data



Create
model

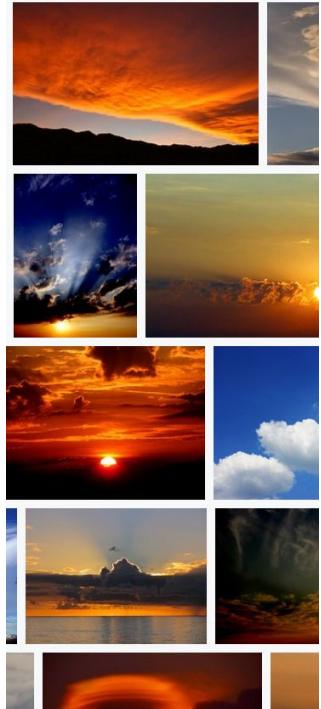


Use machines to
flesh out the
model from data



Deploy fleshed
out model

The Dataset should cover all cases

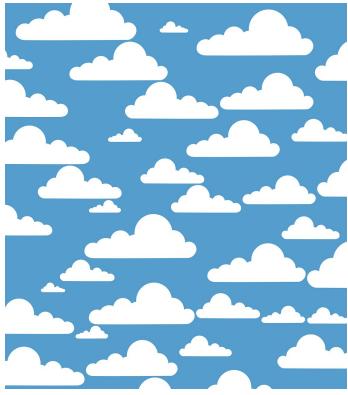


All types covered?

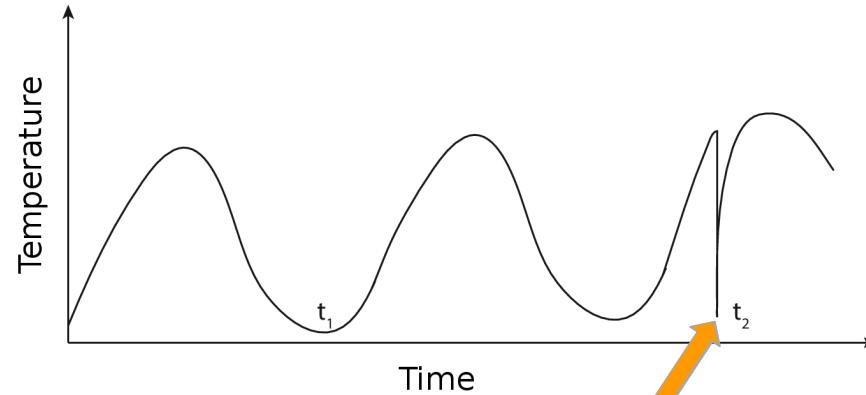
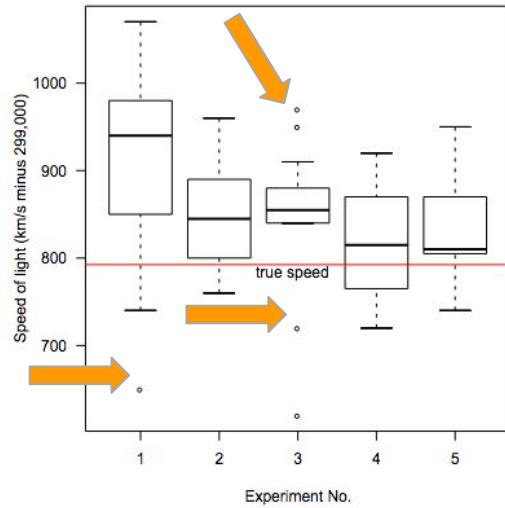
NOAA/NWS AND NASA'S SKY WATCHER CHART

High Clouds			Typical Types: Cirrus (C), Cirrostratus (Cs), Cirrocumulus (Cc)					
H1: Cirrus In the form of filaments, strands, or hooks	H2: Cirrus Dense, in patches or sheaves, not increasing, or with tufts	H3: Cirrus Often an unexpanded remains of a cumulonimbus	H4: Cirrus In tufts or segments, increasing, becoming denser	H5: Cirrostratus Cloud bands or veils, veil below 45° elevation	H6: Cirrostratus Cirrus bands or veils, veil above 45° elevation	H7: Cirrostratus Thin cirrus completely covering the sky	H8: Cirrostratus Not thin cirrus, not covering the whole sky	H9: Cirrocumulus Alone or with some cirrus or cirrostratus
Middle Clouds			Typical Types: Altostratus (As), Altocumulus (Ac), Nimbostratus (Ns)					
M1: Altostratus Mostly semi-transparent, sun or moon may be dimly visible	M2: Altostratus or Nimbostratus Dense enough to hide the sun or moon	M3: Altocumulus Semi-transparent, one level, cloud elements change slowly	M4: Altocumulus Lens-shaped, or continually changing shape and size*	M5: Altocumulus One or more bands or layers, expanding, thickening	M6: Altocumulus From the spreading of cumulus or cumulonimbus	M7: Altocumulus One or more opaque layers, w/ altostratus or nimbostratus	M8: Altocumulus With cumulus-like tufts or furlers	M9: Altocumulus Chaotic sky, usually at several levels, maybe w/ dense cirrus
Low Clouds			Typical Types: Stratus (St), Stratocumulus (Sc), Cumulus (Cu), Cumulonimbus (Cb)					
L1: Cumulus With little vertical extent	L2: Cumulus Moderately strong vertical extent, or towering cumulus	L3: Cumulonimbus Tops not fibrous, outline not completely sharp, no anvil	L4: Stratocumulus From the spreading and flattening of cumulus*	L5: Stratocumulus Not from the spreading and flattening of cumulus	L6: Stratus In a continuous layer and/or ragged shreds	L7: Stratus Fractus and Cumulus Fractus Of great weather	L8: Cumulus & Stratocumulus Not spreading bases at different levels	L9: Cumulonimbus With fibrous top, often with an anvil
 NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION U.S. DEPARTMENT OF COMMERCE			 Mammatus Drooping underside of heavy, rain-saturated clouds					
			Tornado Formed by rotation of up and down drafts within thunderstorms	Wall Cloud Hanging from cumulus, possible tornado formation	Shelf Cloud Leading edge of fast moving frontal system	Wave Cloud Formed by strong horizontal winds over uneven terrain		

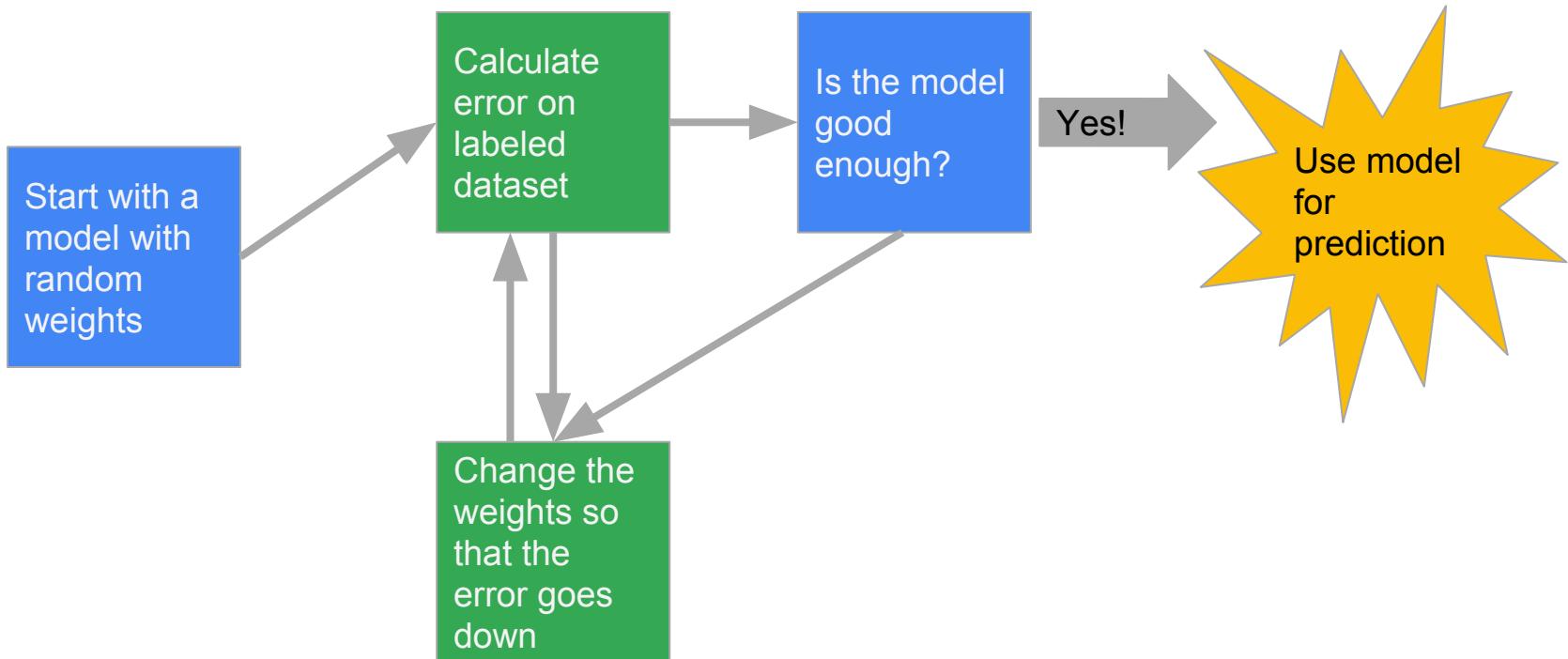
Negative examples and near-misses



Explore the data you have; fix problems



How do we compute error metrics?



Start from the outcome for a single example

ML system says you'll **win \$89**.

Truth: You **lose \$14**.



Regression error: How far from true outcome?

ML system says you'll **win \$89**.
Truth: You **lose \$14**.
Your outcome is the error.

$$\begin{aligned}\text{Error} &= \text{truth} - \text{prediction} \\ &= \textcolor{red}{-\$14} - \textcolor{green}{\$89} = -\$103\end{aligned}$$



Calculating regression error—Outcomes

-\$103
+\$75
-\$10
+\$99
-\$113
+\$82
+\$56



Calculating regression error

1) Get the **error**:

-\$103

+\$75

-\$10

+\$99

-\$113

+\$82

+\$56

2) **Square** the error:

10609

5625

100

9801

12769

6724

3136



3) Calculate the **mean**:
6966

MSE

mean squared error

Mathematically...

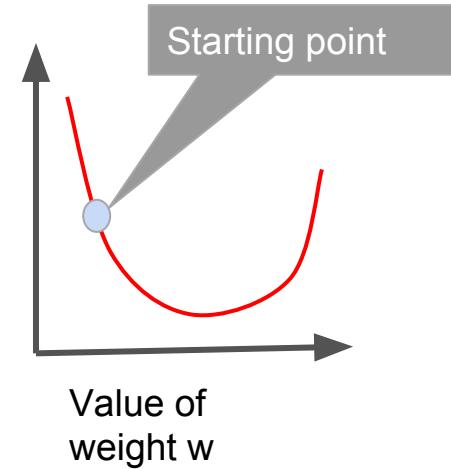
\hat{Y} -cap is the model estimate

Y is the labeled value

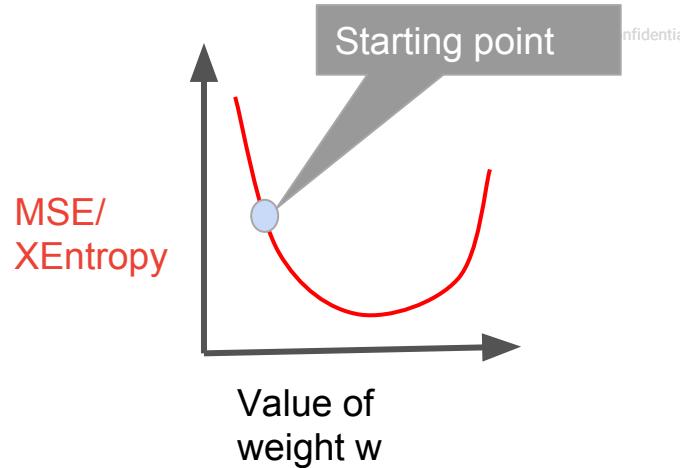
Mean Square Error (MSE) is:

MSE

$$\frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2$$



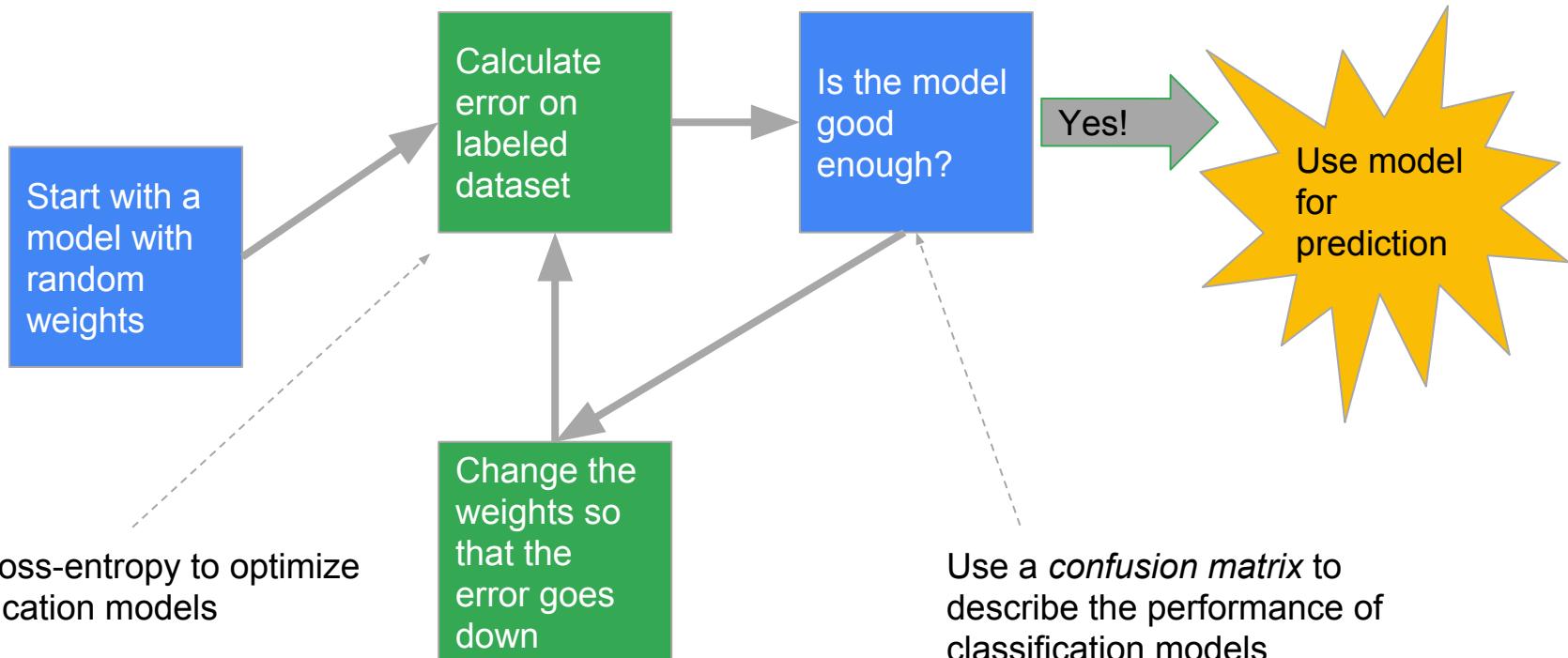
For classification problems, we use cross-entropy



For classification problems, the most commonly used error measure is *cross-entropy*—because it is differentiable:

$$-\frac{1}{N} \sum_{n=1}^N \left[y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n) \right]$$

Cross-entropy is not intuitive to business users



Confusion matrix

1) Get the outcomes

TN

FP

TP

FN

TP

FN



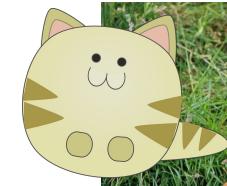
		ML System Says	
		Cat	No Cat
Truth	Cat	True Positive #TP	False Negative #FN
	No Cat	False Positive #FP	True Negative #TN

Accuracy, precision, and recall

Classify the cats!

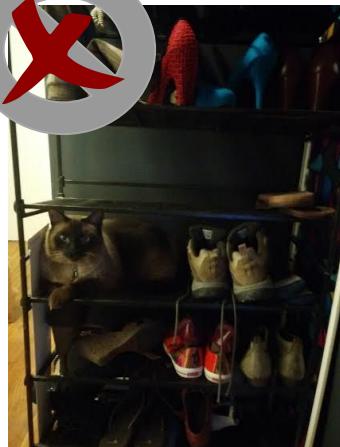
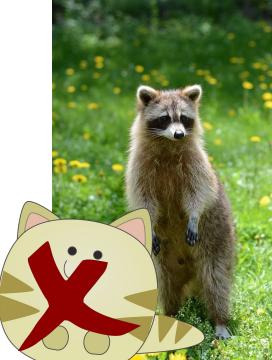
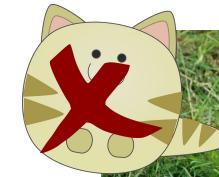


Hypothetical results from ML



Accuracy is fraction correct

Accuracy = 3 / 8
= 0.375



Accuracy fails if dataset unbalanced



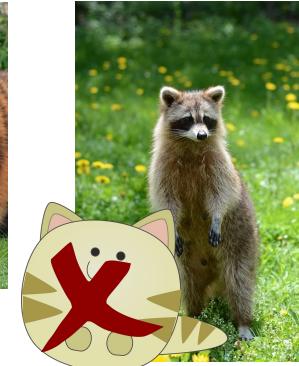
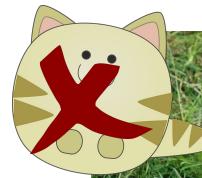
1000 parking spaces
990 of them are **taken**
10 are **available**

A ML model that
identified only **one** of the
ten **available** spaces:

Accuracy = $991/1000$
= 0.991!

Precision = Positive Predictive Value

Accuracy when
ML says “cat”



TP + FP = 5



$$\begin{aligned}\text{Precision} &= \text{TP} / (\text{TP} + \text{FP}) \\ &= 2 / 5 = 0.40\end{aligned}$$



Recall is true positive rate

Recall = fraction of cats ML finds

$$TP + FN = 4$$



$$\begin{aligned} \text{Recall} &= TP / (TP + FN) \\ &= 2 / 4 = 0.50 \end{aligned}$$



Do Now: In your own words, write down definitions for these ML terms

Term	Meaning
MSE	
Cross-entropy	
Accuracy	
Precision	
Recall	

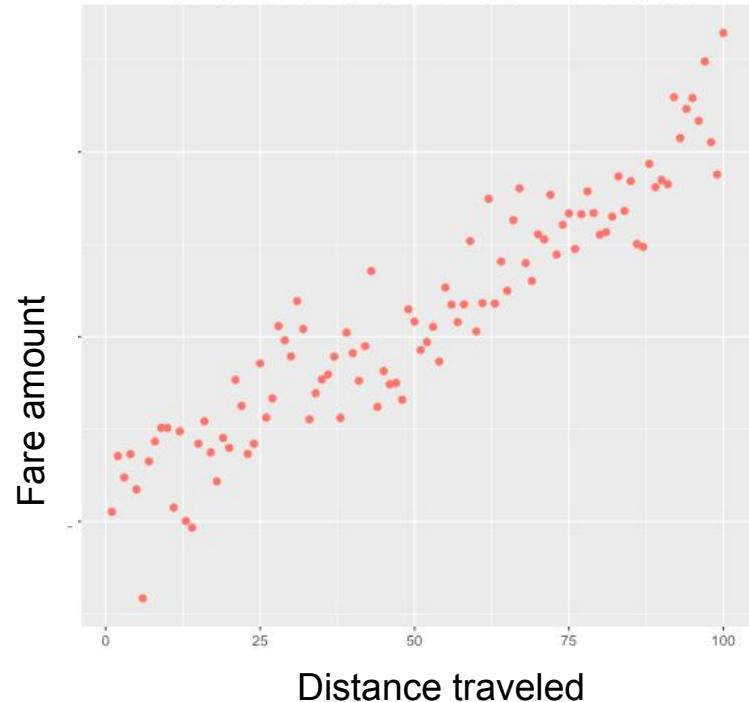
Agenda

Creating ML Datasets + Lab

Regression problem: Predict taxi fare

Problem: predict taxi fare amount based on distance traveled

What is the error measure to optimize?

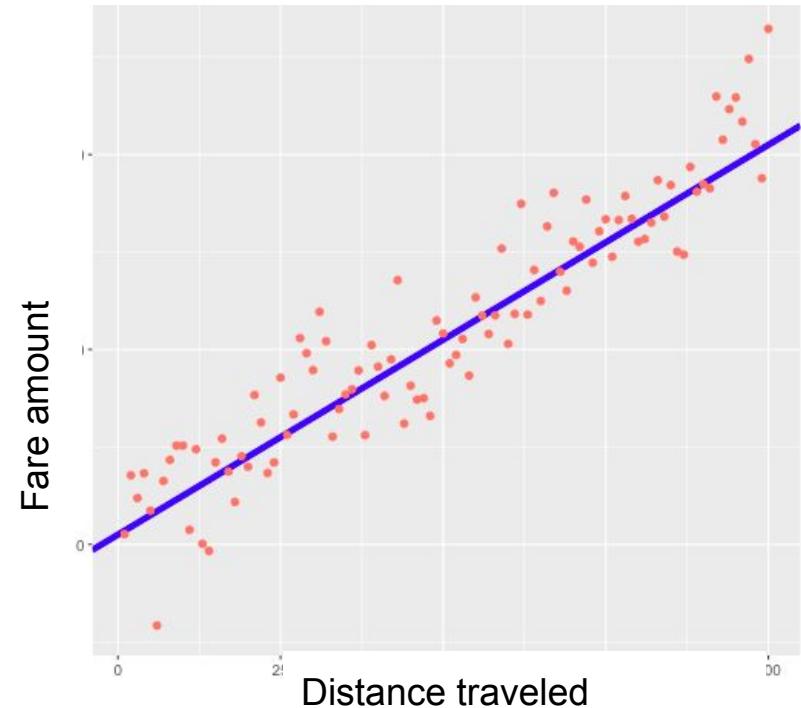


Model 1

Red = training data

Blue = model prediction for
each distance traveled

RMSE = **22.24**

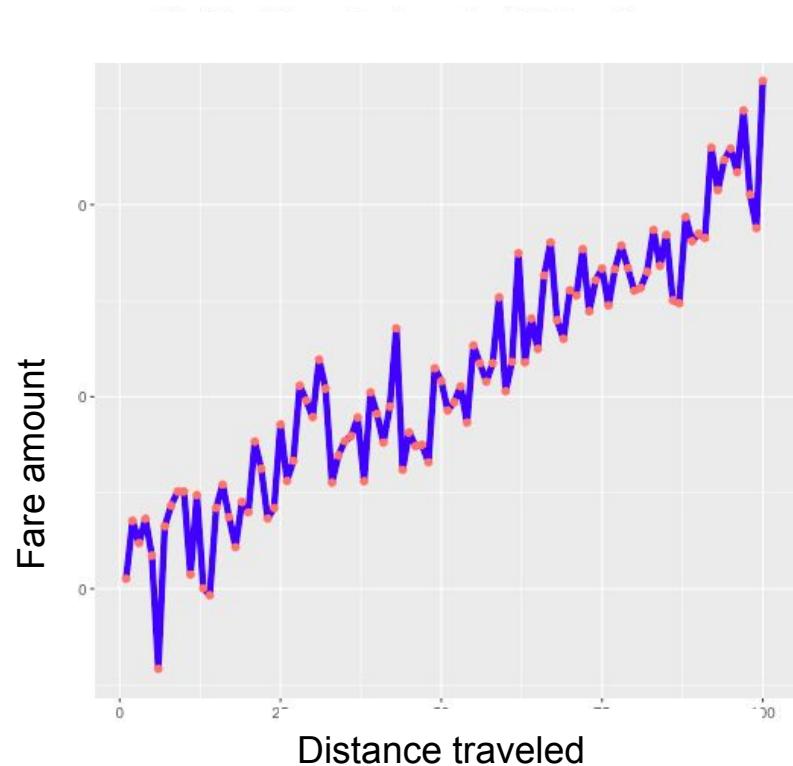


Model 2 has more free parameters

RMSE = 0

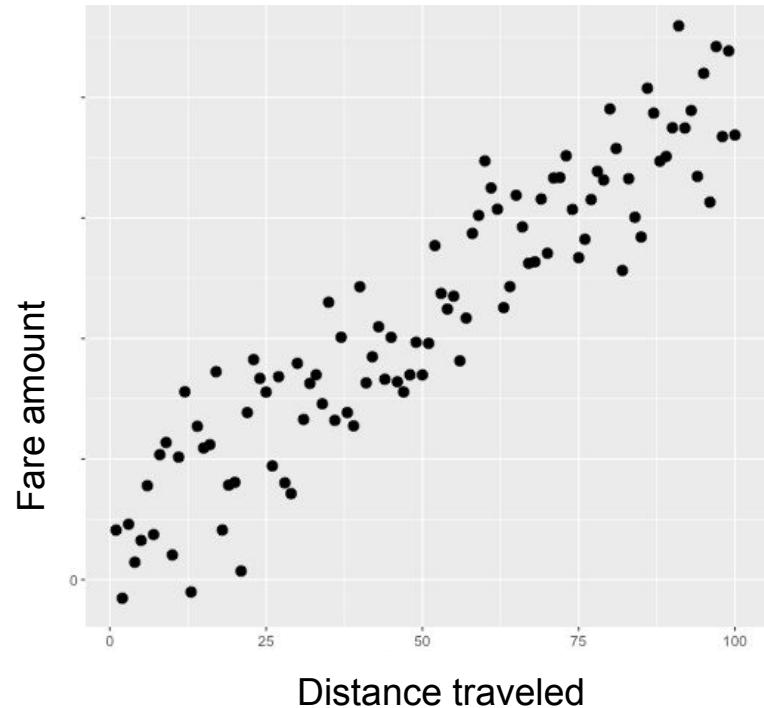
Which model is better?

How can we tell?



Does the model generalize to new data?

Need data that were not used
in training

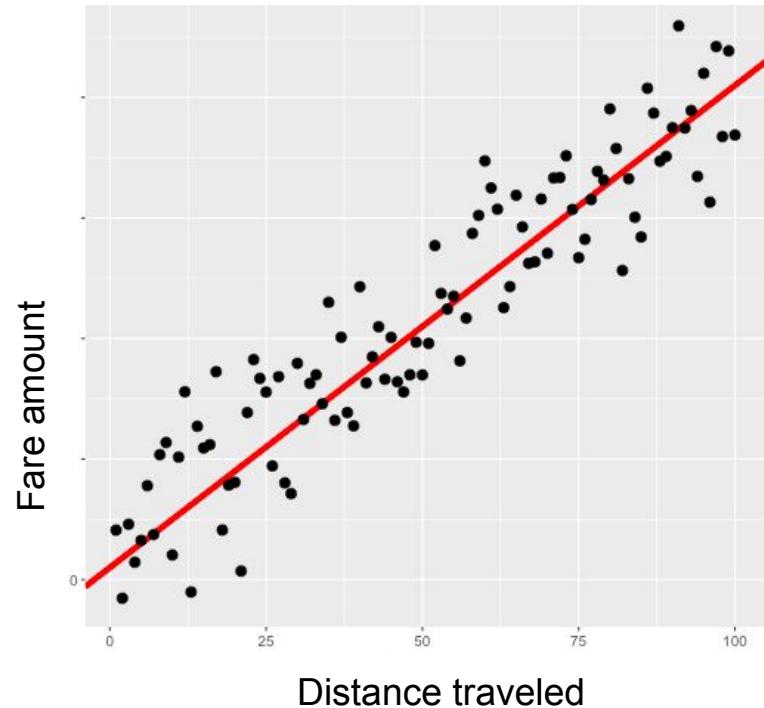


Model 1 generalizes well

Old RMSE = **22.24**

New RMSE = **21.98**

Pretty similar = good

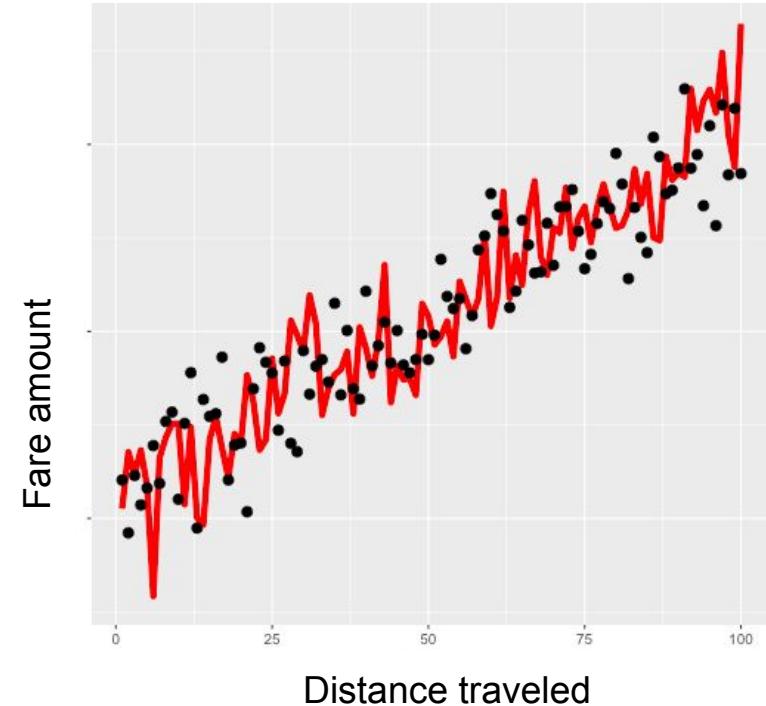


Model 2 does not generalize well

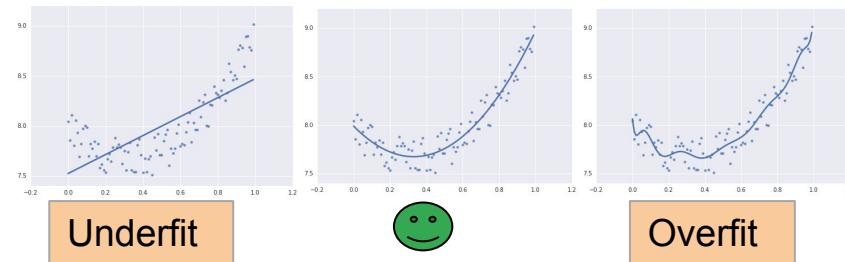
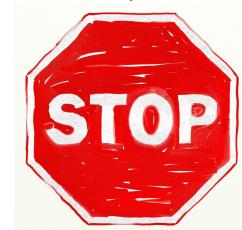
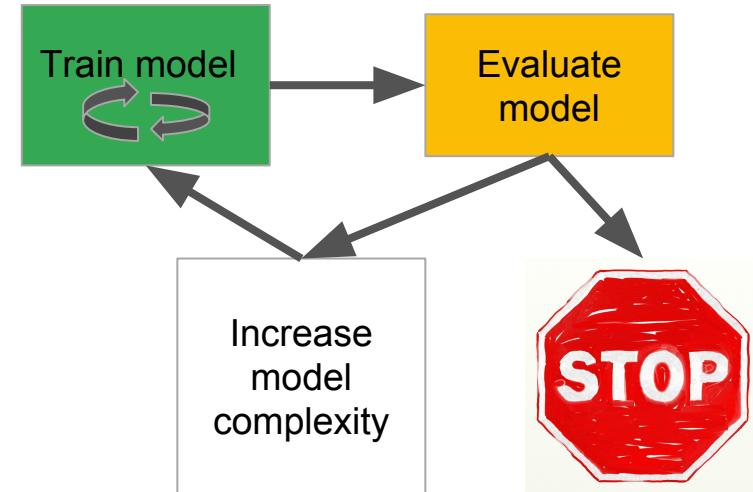
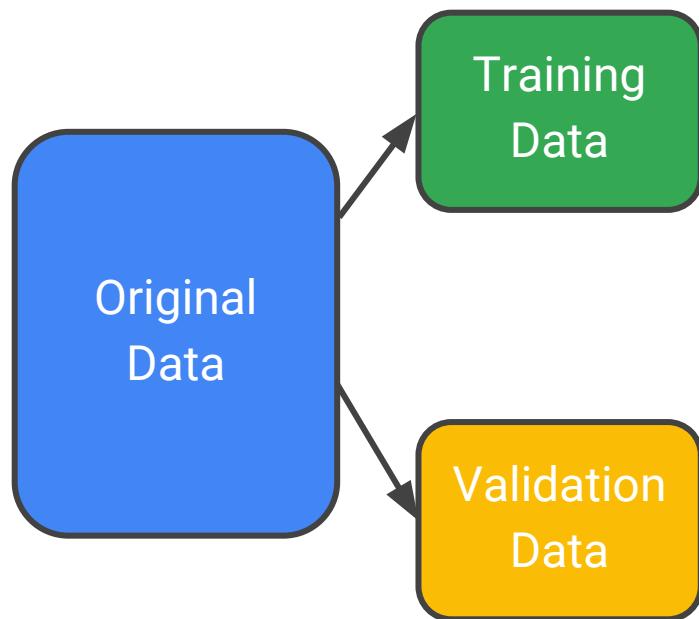
Old RMSE = 0

New RMSE = 32

This is a red flag

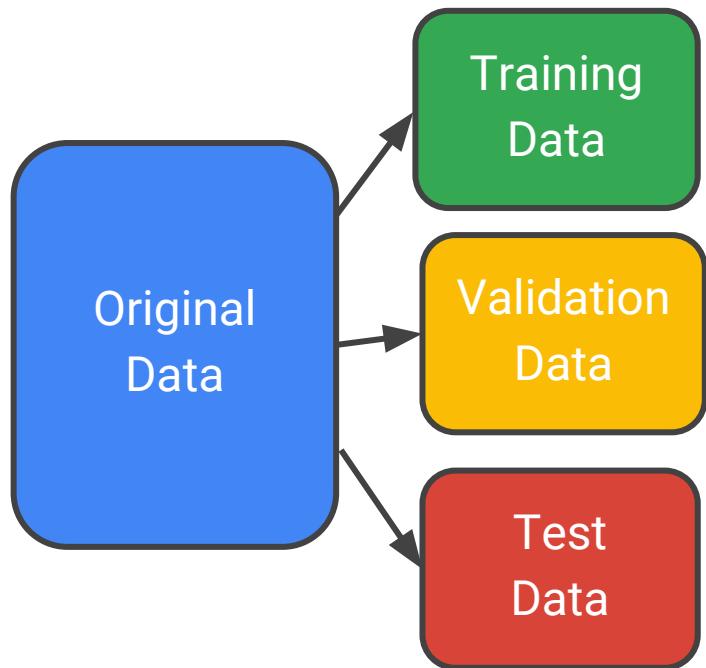


Split dataset, experiment with models



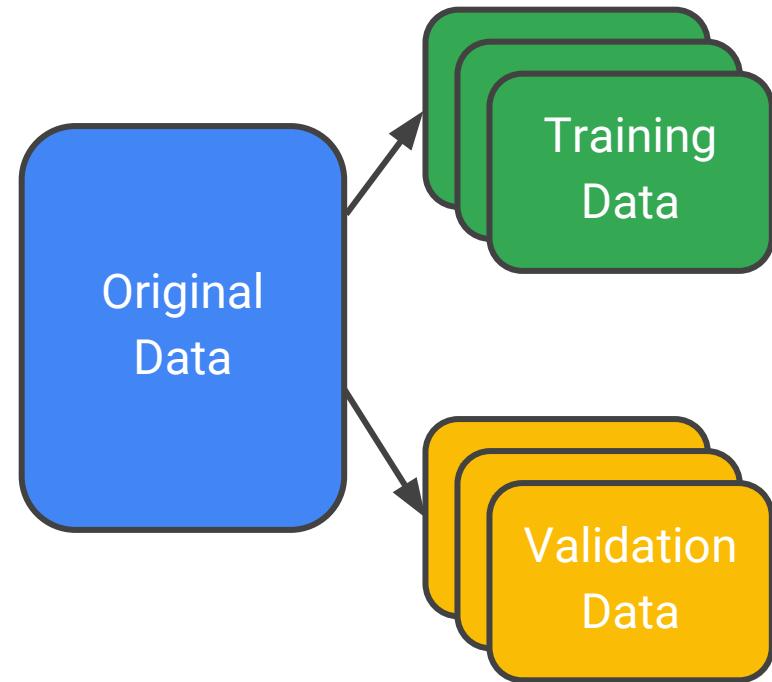
To evaluate the final model...

Use independent test data

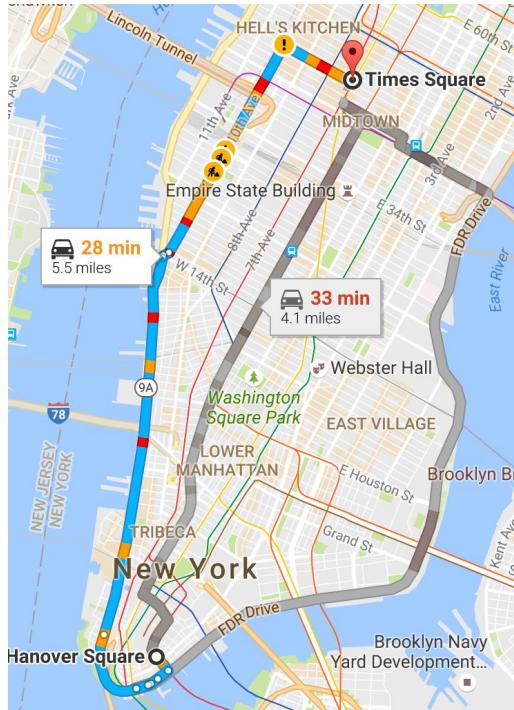
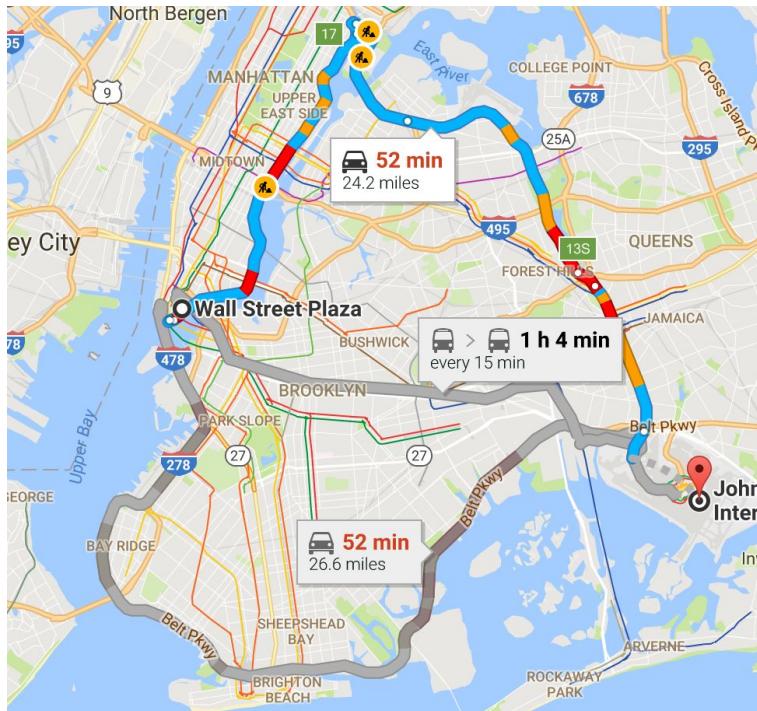


OR

Cross-validate if data is scarce

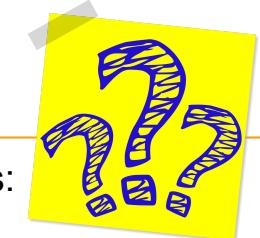


Goal: To estimate taxi fare



http://www.nyc.gov/html/tlc/html/passenger/taxicab_rate.shtml

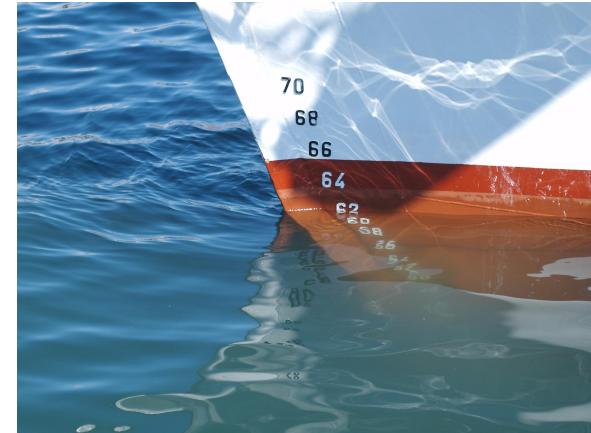
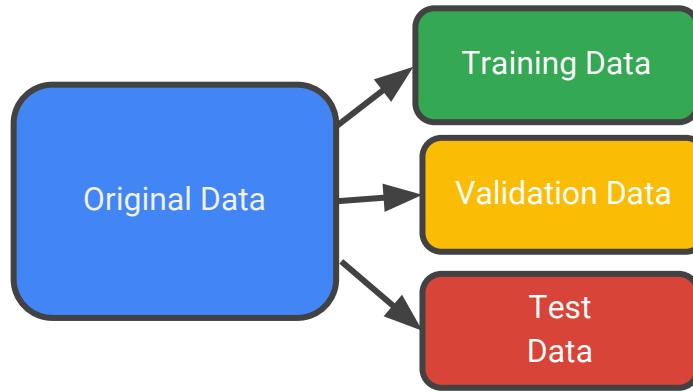
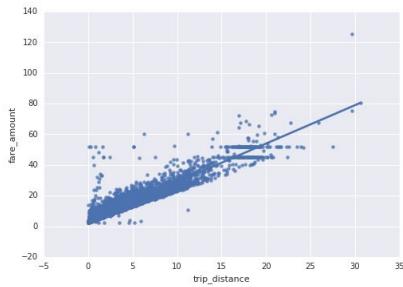
Taxi fares:



\$2.50 initial charge
+
50c per $\frac{1}{5}$ mile
(or)
50c per minute if stopped
+
Passenger pays tolls
+
Various special charges

Lab: Serverless Machine Learning

Part 1. Explore datasets, create ML datasets, create benchmark



1. Explore

2. Create Datasets

3. Benchmark

cloud.google.com

