Google Cloud

Getting started with Machine Learning

Data Engineering on Google Cloud Platform

Notes:

2 hours lecture + 30 min of interactive quizzes interspersed + 30 min lab. Be smart about the timeline for this section. It will take the full 3 hours for an audience that is new to ML but should take less time if they already know some of this. Before you start diving into the slides, ask questions like:

(1) Do you know what ML is?
(2) Have you built a ML model before? What toolkit did you use?
(3) Do you know what deep neural networks are?

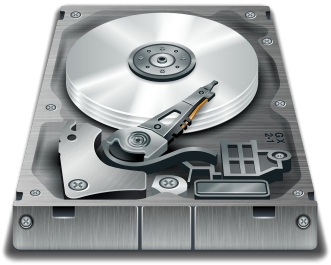
Based on the answers to these questions, pace this material appropriately. For example, you should plan on covering this chapter in 60 minutes if the audience has prior experience with ML -- don't belabor the point and feel free to skip slides.

Be smart about how deep to cover this section:

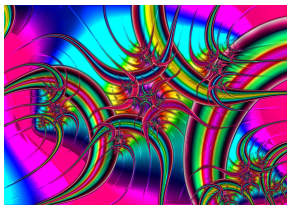
(1) Normal pace
(2) Cover only slides 6 and 17, using them to summarize the main points of section.
(3) Skip this section completely.

Choose option #1 if these are developers new to ML. Choose #2 if these are enthusiasts who have read about ML already but have not built a ML model for their own use cases. Choose #3 if these are people who have used scikit-learn etc.

Machine Learning: Way to derive insights from data



data



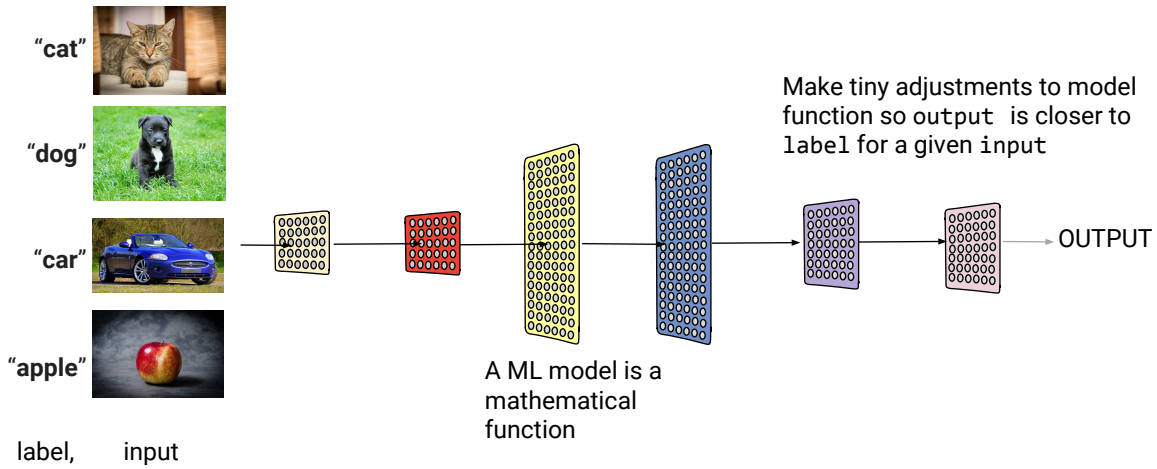
algorithm

insight

Notes:

ML is a way to derive insights from data.
https://pixabay.com/en/hard-disk-storage-computer-159264/ (cc0)
https://pixabay.com/en/fractal-pattern-abstract-form-142748/ (cc0)
https://pixabay.com/en/arrows-inside-pressure-request-2029157/ (cc0)

Step 1: Train a ML Model with examples



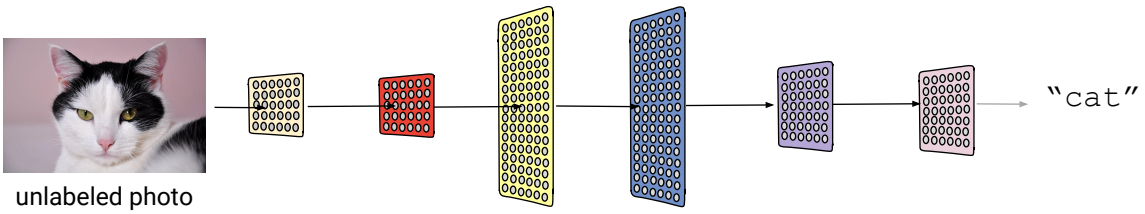
Notes:

- Key words on this slide: "train", "example", "label", "input", "model", "output" -- use these words in a sentence. Do not bother to define these terms formally -- just hearing the words in context will help them understand what they mean … We'll repeat this for a few more examples to help attendees think in a ML manner.
- That ML is a function can be hard for attendees to see. As an example, assume that the first layer returns the number of pixels that are brown/black/blue/red, and the second layer finds the most common color, and the third layer returns "cat" if previous layer had supplied "brown". Based on just these images, that is a perfectly valid ML model. With more data, we'll have to do more finely tuned ML functions.
    - Now, if you know ml, you are probably skeptical that my "function" above is really achievable with a neural network, so consider this. Mathematically, this model would be, for the first layer, [sum(r = 255, g=255, b=255), …, …, sum(r=255, g=0, b=0)] -- this is just a set of appropriately positioned relu functions (okay, for r=234, we'd need two relu functions, so two layers, but you get the idea). The second layer would be a softmax layer. The third layer is simply an identity! [Instructors: don't go into this since it will only confuse a class … if challenged, ask the person

    - doing the challenging to read the slide notes.]
- The idea is that by exposing these neural nets to million of images and comparing their outputs to the desired output, we can make tiny adjustments to the network with the goal of improving its prediction capability -- training is an iterative operation.

https://pixabay.com/en/domestic-cat-cat-adidas-relaxed-726989/ (cc0)
https://pixabay.com/en/dog-young-dog-puppy-280332/ (cc0)
https://pixabay.com/en/sports-car-vehicle-transportation-1317645/ (cc0)
https://pixabay.com/en/apple-education-school-knowledge-256268/ (cc0)

Step 2: Predict with a trained model



Notes:

- And after the neural net is trained, we can use it to label images that it has never seen before.
- In this slide, we are giving the neural network this image and because the network has been trained, it is correctly able to output “cat”.
- Note the cat image on this slide is different from the one before it -- still works; The key to making all this work is data and lots and lots of it.

https://pixabay.com/en/cats-a-normal-cat-pet-796437/ (cc0)

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

Term	Meaning
Label	
Input	
Example	
Model	
Training	
Prediction	

Notes:

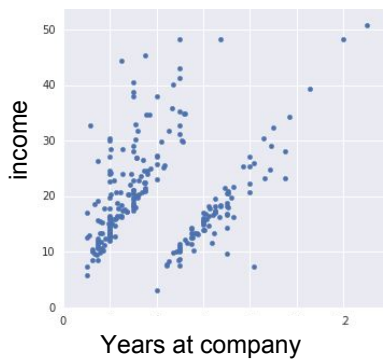
Label = true answer
Input = predictor variable(s), what you can use to predict the label
Example = input + corresponding label
Model = math function that takes input variables and creates approximation to label
Training = adjusting model to minimize error
Prediction = using model on unlabeled data

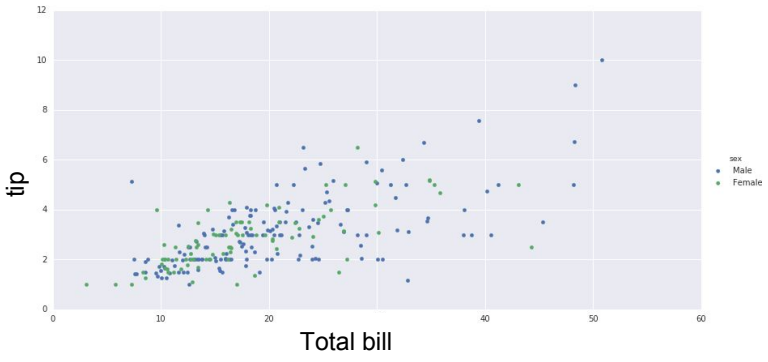
Students are likely to struggle with the idea of a model. So, in the recap to this slide, delve more into models. So explain a ML model as a math function that weights & adds inputs and does mathematical transformations like tanh() to the numbers. And tell them that we will look at it again and again.

- That ML is a function can be hard for attendees to see. As an example, assume that the first layer returns the number of pixels that are brown/black/blue/red, and the second layer finds the most common color, and the third layer returns “cat” if previous layer had supplied “brown”. Based on just these images, that is a perfectly valid ML model. With more data, we’ll have to do more finely tuned ML functions.
- Now, if you know ml you are probably skeptical that my “function” above is really achievable with a neural network, so consider this.

- Mathematically, this model would be, for the first layer, [sum(r = 255, g=255, b=255), …, …, sum(r=255, g=0, b=0)] -- this is just a set of appropriately positioned relu functions (okay, for r=234, we'd need two relu functions, so two layers, but you get the idea). The second layer would be a softmax layer. The third layer is simply an identity! [Instructors: don't go into this since it will only confuse a class … if challenged, ask the person doing the challenging to read the slide notes.]

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

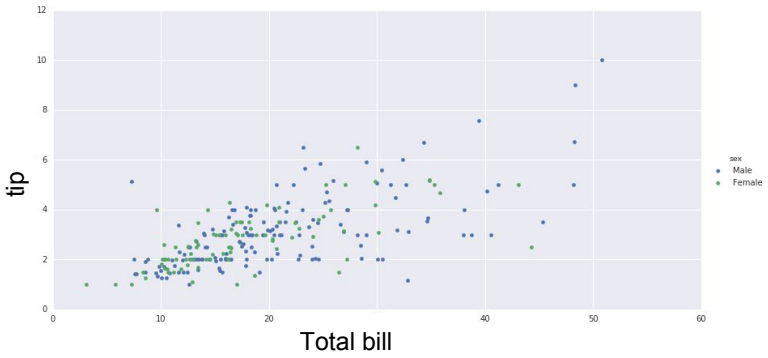
Notes:

The leftmost use case is that we want to look at tenure and income across a company to find out whether someone is on the "fast track" or not. Data not labeled; Clustering attempts to group points close to each other. This is a discovery problem.

Second: based on total-amount and gender of customer, predict the tip-amount.
Third: based on total-amount and tip, predict the gender.

Graphs created by course author using seaborn "tips" dataset.

We'll focus on supervised learning



Regression
Predict the tip amount

Classification
Predict the gender of the customer

Notes:

#2 and #3 both have labels (supervised). That is what we will focus on in this course. In other words, on prediction and not on description.

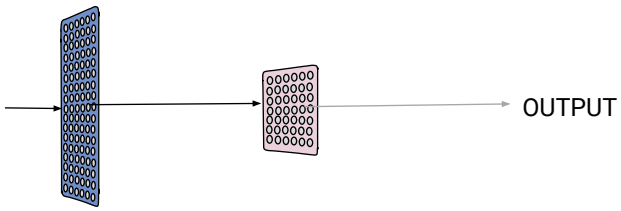
Graphs created by course author using seaborn "tips" dataset.

This regression model predicts a continuous number

gestation_weeks	mother_age	cigarette_use	alcohol_use	weight_gain_pounds
36	23	false	false	70
37	23	false	false	25
38	21	false	false	33
39	16	false	false	26
39	23	true	true	null
41	30	false	false	8

36

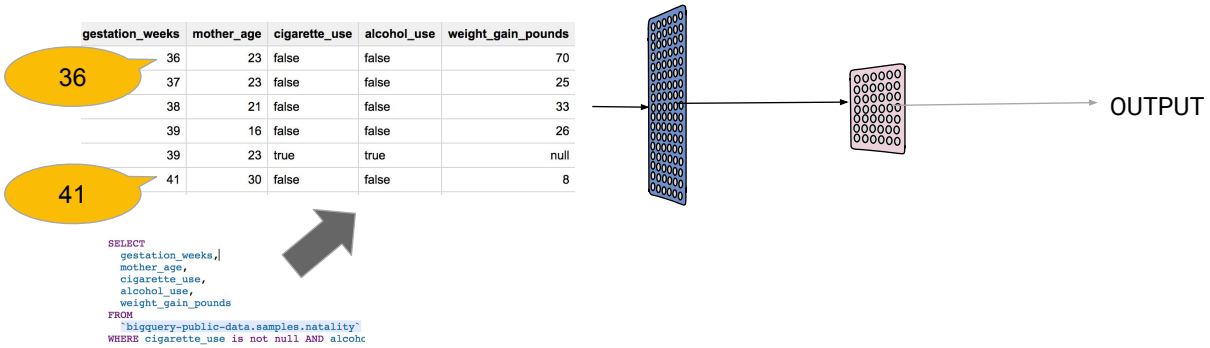
41



Notes:

- Key words on this slide: “regression” and “continuous” -- use these words in a sentence.

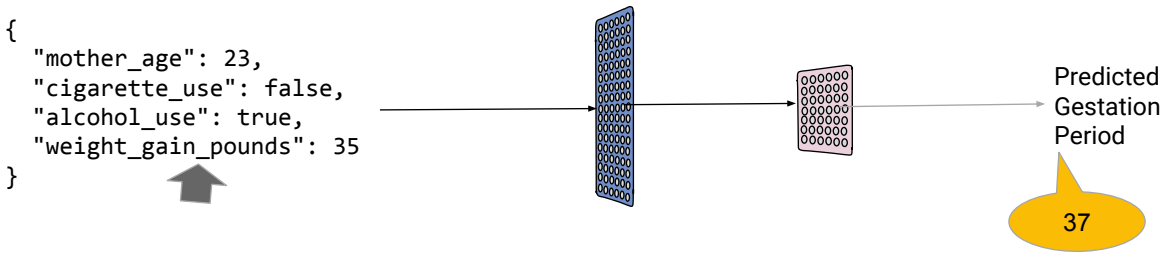
A common source of structured data for ML training is your data warehouse



Notes:

Key word here “structured data” … Here the source is BigQuery.

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

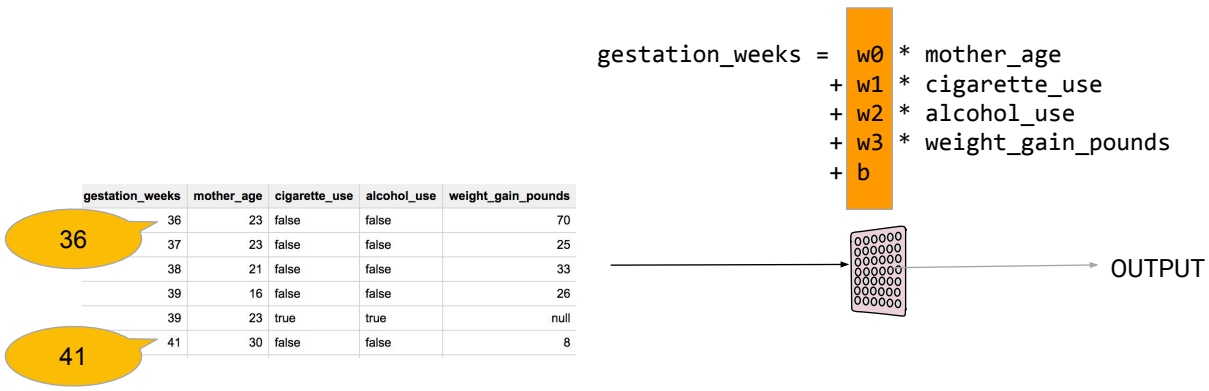


Notes:

- The key point here is that you are predicting from real-time data, for example, from Pub/Sub or your web application. This itself might come from a physician.

https://pixabay.com/en/doctor-medical-medicine-health-563428/ (cc0)

The model may even have only one layer



gestation_weeks	mother_age	cigarette_use	alcohol_use	weight_gain_pounds
36	23	false	false	70
37	23	false	false	25
38	21	false	false	33
39	16	false	false	26
39	23	true	true	null
41	30	false	false	8

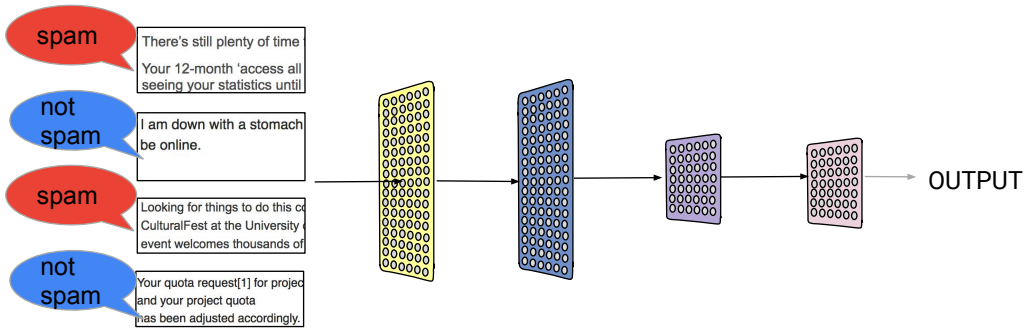
Notes:

Training involves finding weights w0, w2, w2, w3 such that the predicted gestation_weeks is really, really close to the labels.

A neural network with no hidden layers is a linear model …

- Point out that the model here is a lot simpler than the one on the previous slides -- more layers in a DNN makes it more complex. Models on structured data are typically only a few layers deep whereas image classification DNNs can have hundreds of layers.

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



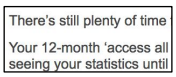
The input here is text

Notes:

- Key words on this slide: “classification” -- use these words in a sentence.
- Say that is another classification problem, similar to the dog/cat/ example. In this case, we have only two categories or labels.
- The input here does not seem numeric ...

The inputs for unstructured data are still ultimately just numbers

N-dimensional array of pixel values



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

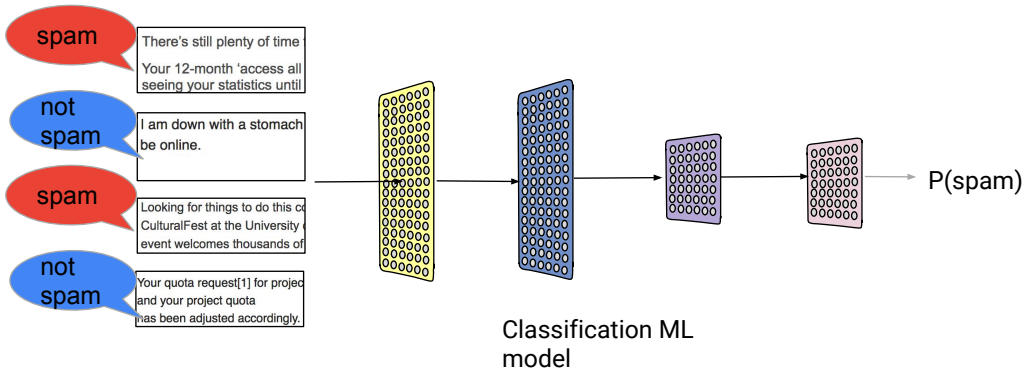
Notes:

Pixel values here are [r, g, b, a] and the image itself is 2D, so it could be a width x height x 4 array.

word2vec

https://pixabay.com/en/rocket-launch-smoke-rocket-take-off-67723/ (cc0)

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



Notes:

- Note the probabilistic output -- we would have to threshold the probability if we wanted “spam” or “not spam”. The previous model returned a probability for each label (cat, dog, etc.) and we picked the most likely (usually, we also threshold; i.e., we don’t return “cat” if the P(cat) is only 0.01).

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

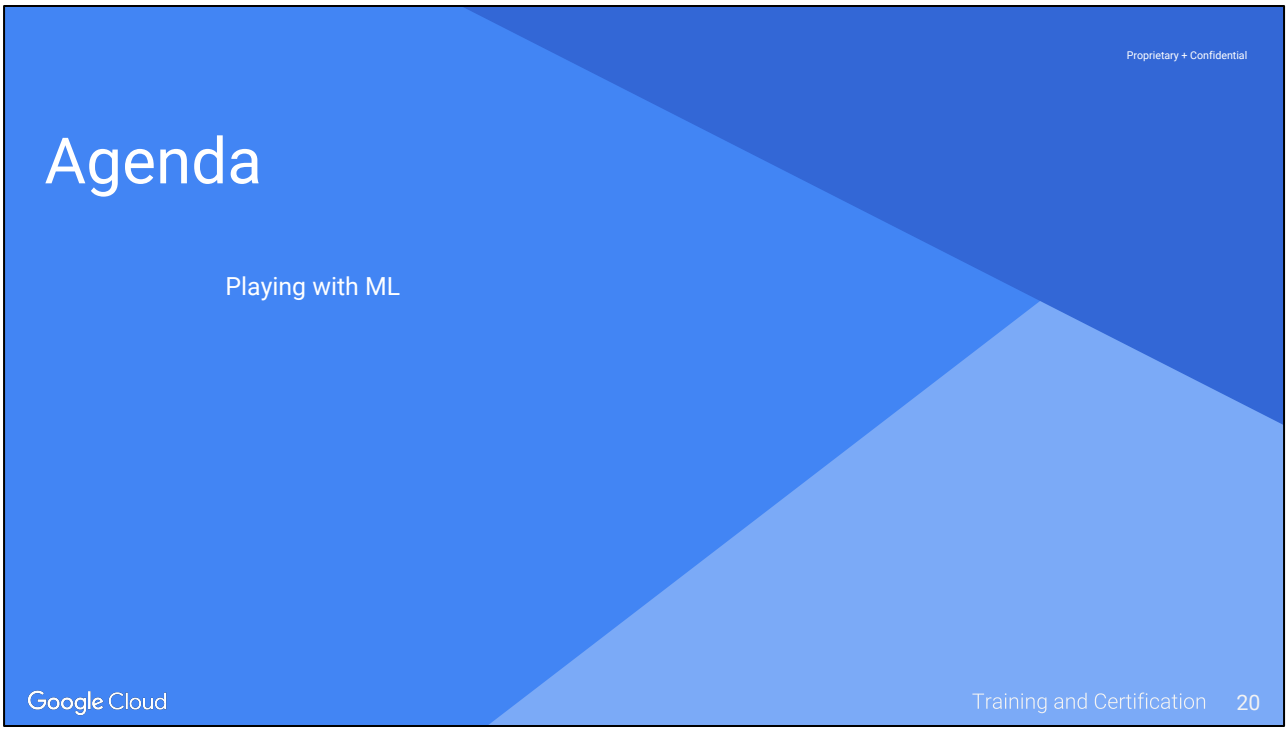
- Risk analytics and regulation
- Customer Segmentation
- Cross-selling and upselling
- Sales and marketing campaign management
- Credit worthiness evaluation

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

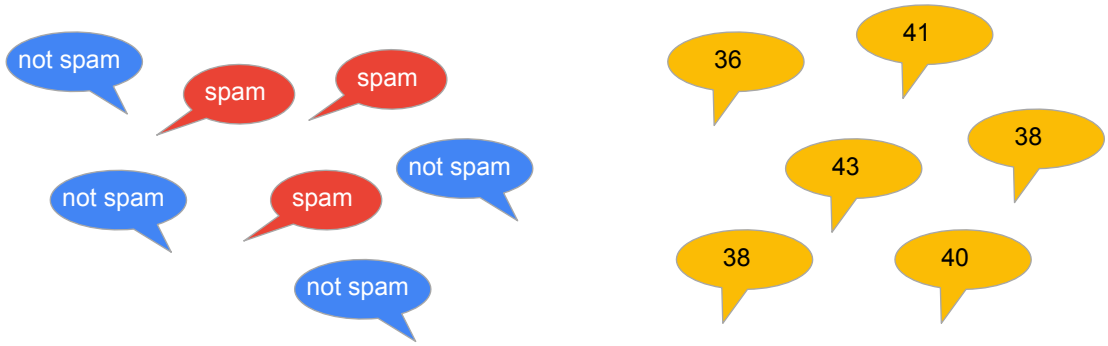
Do Now: Pick 5 use cases from previous slide and fill out this table

Use case	Label	Input(s)	Classification or regression?



This section should take 20-40 minutes based on audience. In my experience, even people who know ML get something out of this section because it introduces neural networks, deep networks, and feature engineering in an intuitive way. However, the more sophisticated the audience, the faster you should cover this.

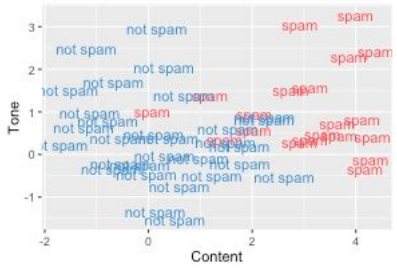
Machine Learning is an approach to making many similar decisions based on data

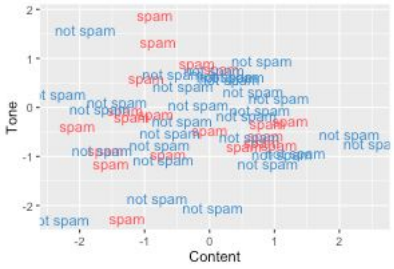


Notes:

You train the computer using lots of examples. On the left is spam-filtering. On the right, the gestation model.
This slide is here to make more obvious the relationship between the “real” examples in previous section and the “toy” example in this one.

ML = Pattern recognition from examples



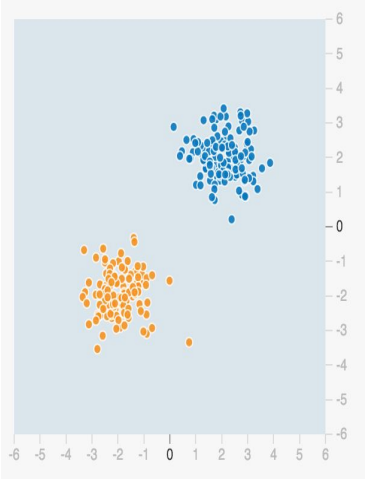


Notes:

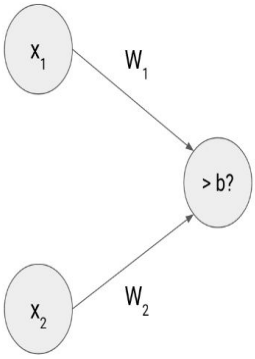
The right-hand side is a more difficult problem than the left-hand side, but with non-linear models and transformations, the ML model can usually manage to find a way to separate the two classes to varying levels of accuracy. Use the right-hand side to lead into a discussion on accuracy, which leads into next slide. Also, tie in the blue-red here to the dots on the next slide.

Graphs created by Cassie Kozyrkov, Google

How do you classify these points?



We could create a ML model consisting of a single neuron



$$w_1 x_1 + w_2 x_2 > b$$

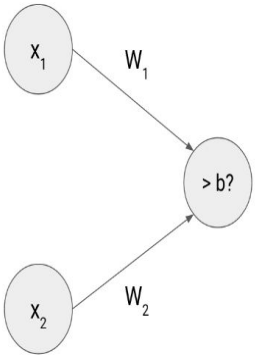
Graphically, that translates to: find a line that separates the two sets of points

bias
(threshold)

$$w_1 x_1 + w_2 x_2 > b$$

weights

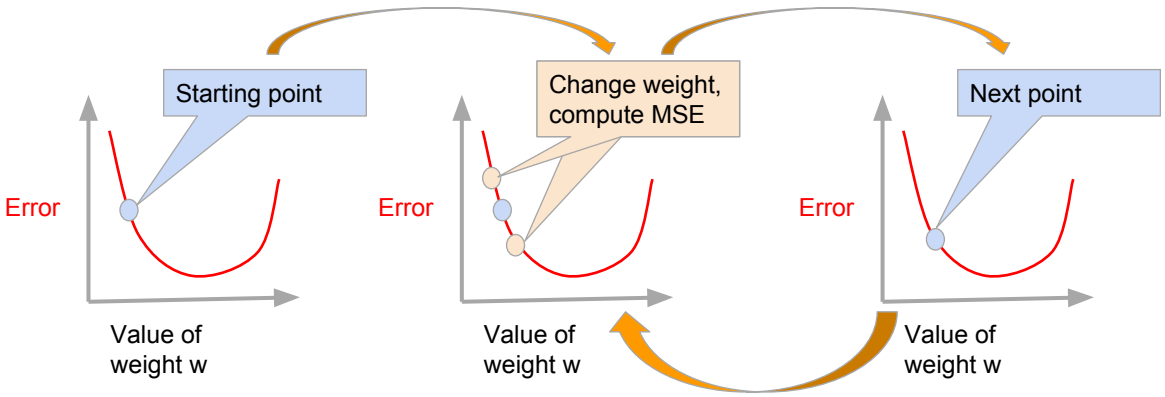
We can use gradient descent to find the best weights and bias



$$w_1 x_1 + w_2 x_2 > b$$

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

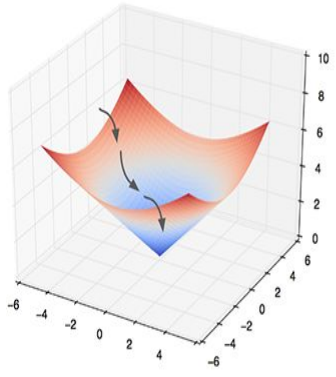
Recompute error after each batch of examples (not full dataset)



Notes:

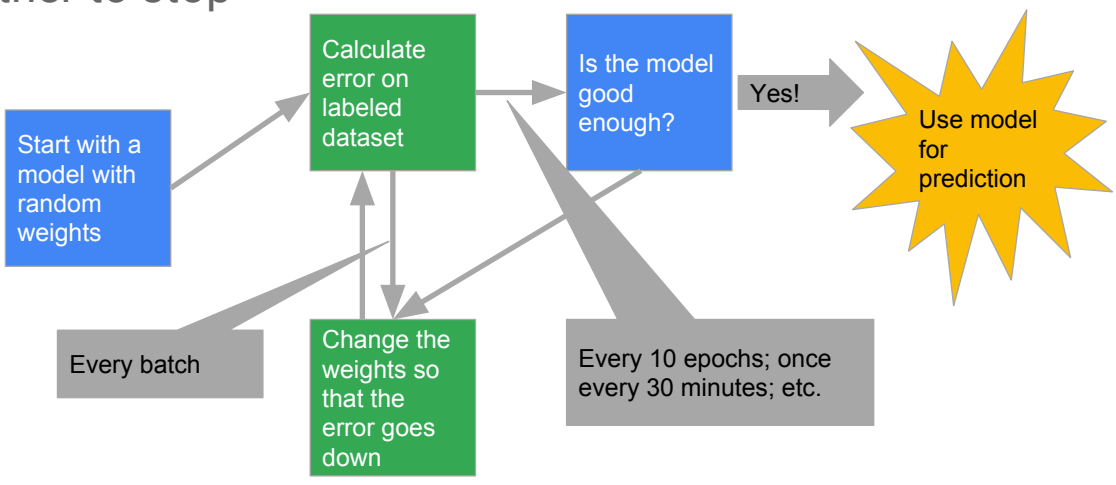
Typical batch size = 100-500 samples.

Gradient descent is used to find the best parameters



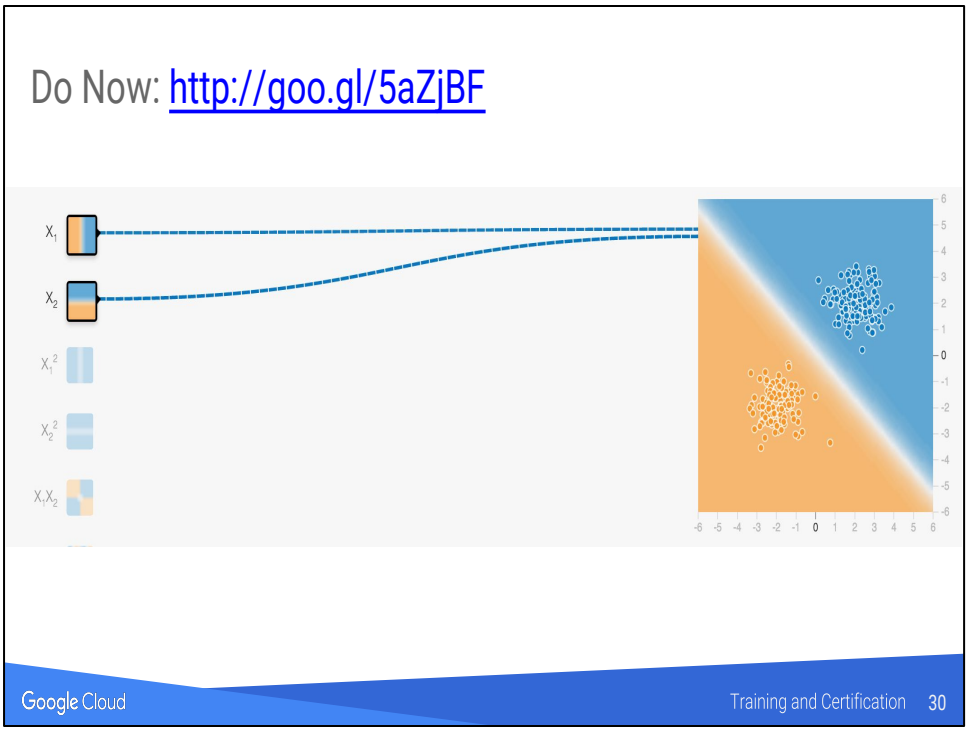
	-5	-4	-3	-2	-1	0	1	2	3	4	5
5	50	41	34	29	26	25	26	29	34	41	50
4	41	32	25	20	17	16	17	20	25	32	41
3	34	25	18	13	10	9	10	13	18	25	34
2	29	20	13	8	5	4	5	8	13	20	29
1	26	17	10	5	2	1	2	5	10	17	26
0	25	16	9		1	0	1	4	9	16	25
-1	26	17	10	5	2	1	2	5	10	17	26
-2	29	20	13	8	5	4	5	8	13	20	29
-3	34	25	18	13	10	9	10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



Notes:

Note that after we calculate error on batch, we can either keep going or we can evaluate the model.
Evaluating the model needs to happen on full dataset, not just a small batch!



Notes:

Playground URL for this pattern: http://goo.gl/5aZjBF

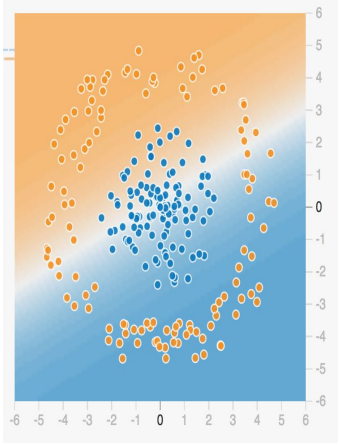
Do Now: In your own words, write down definitions for these ML terms

Term	Meaning
Weights	
Batch size	
Epoch	
Gradient descent	
Evaluation	
Training	

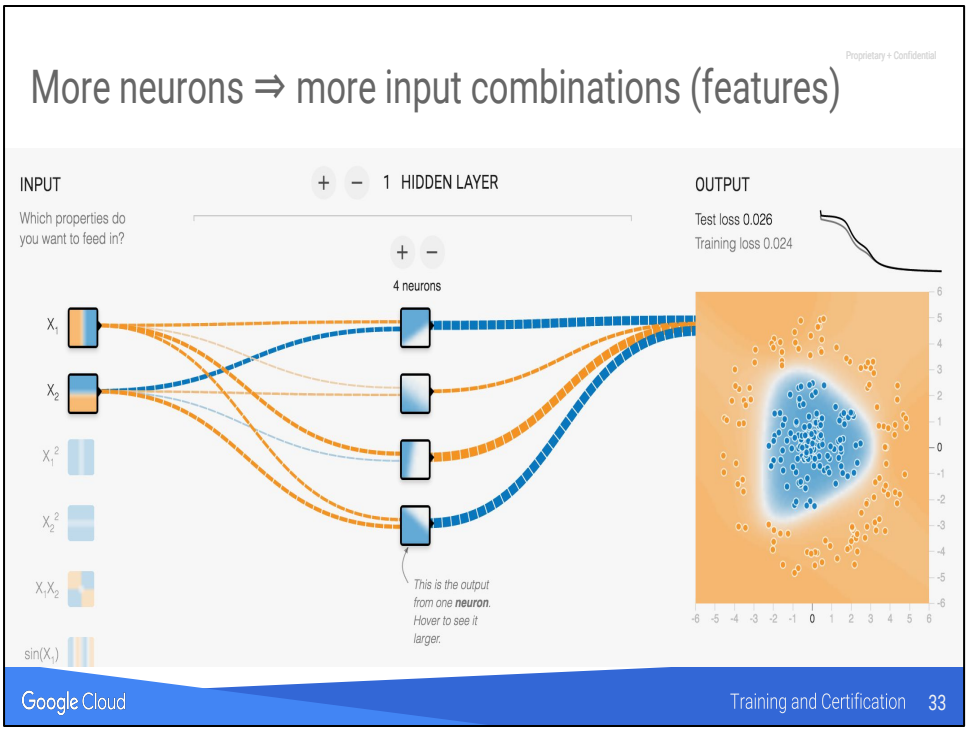
Notes:

Weights/bias = parameters we optimize
Batch size = the amount of data we compute error on
Epoch = one pass through entire dataset
Gradient descent = process of reducing error
Evaluation = is the model good enough? Has to be done on full dataset
Training = process of optimizing the weights; includes gradient descent + evaluation

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



http://goo.gl/v7qM4Q

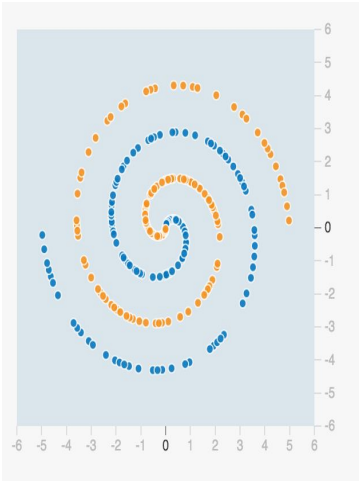


Notes:

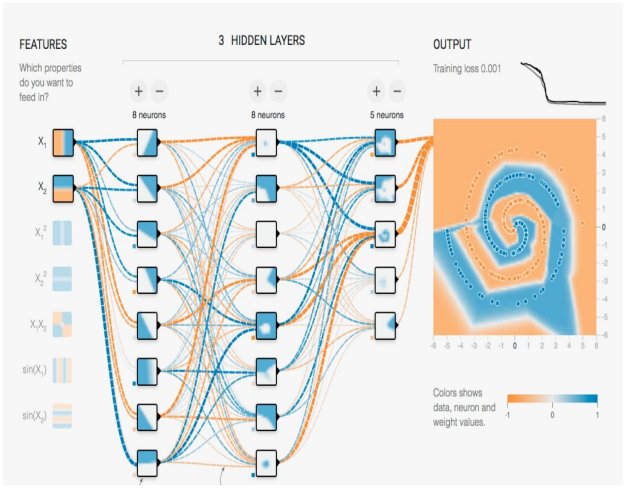
Note that the separation boundary is a polygon … each side is a linear combination of features …. We should have gotten a 4-sided polygon, but in the diagram above two of the sides merged (i.e., their weights were linearly correlated).

Playground URL for this pattern: http://goo.gl/rSX7Ve

How about this? Will a set of lines work?



More hidden layers ⇒ more hierarchies of features

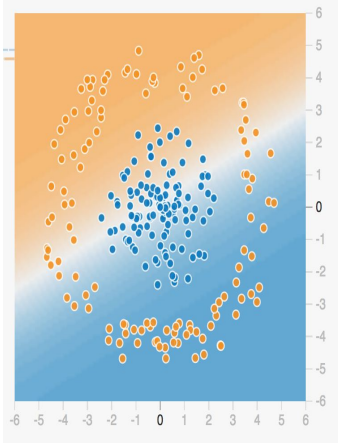


http://goo.gl/fymPMI

Notes:

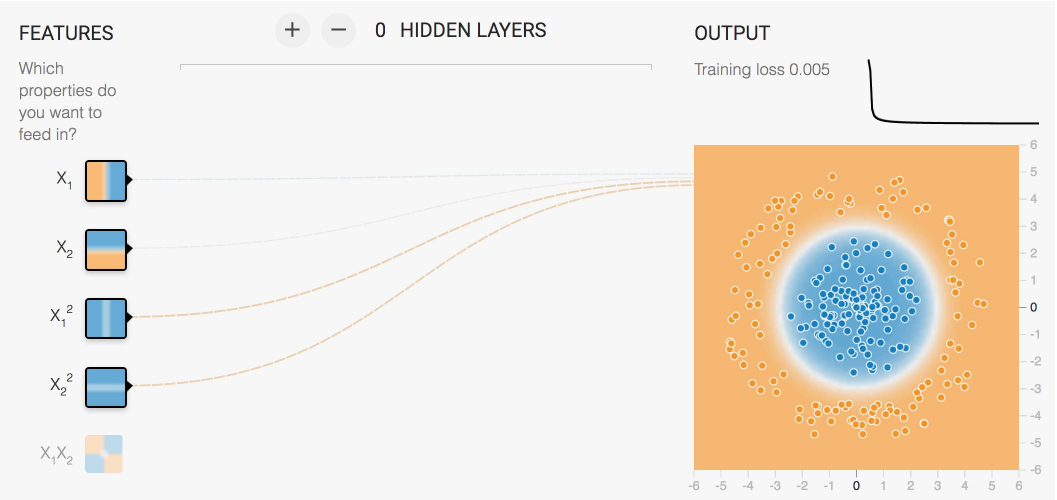
Even a complex non-linear pattern such as the double spiral could be classified by the simple neural network.

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



http://goo.gl/v7qM4Q

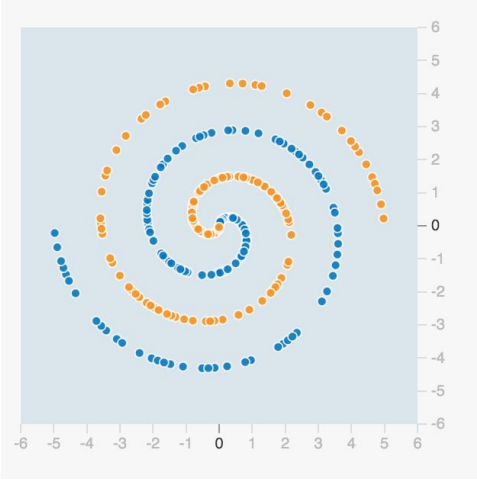
Engineer some extra features



Notes:

The reason we know that x^2 and y^2 will work is because we realize that points close to the origin are blue and those far away are orange. This indicates that distance would be a good feature, hence x^2 + y^2 ...

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

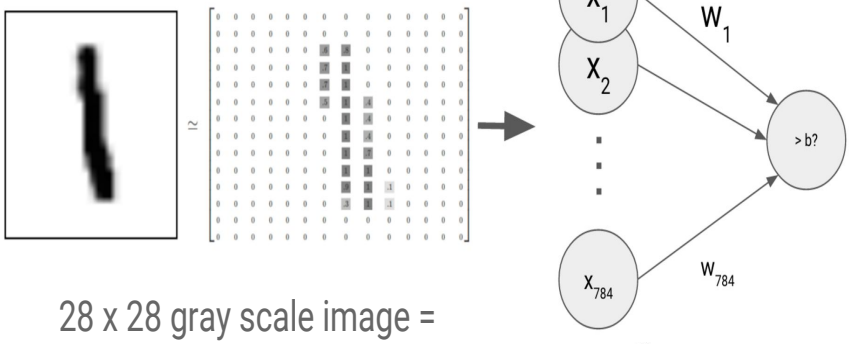
Term	Meaning
Neurons	
Hidden layer	
Inputs	
Features	
Feature Engineering	

Notes:

Neuron = one unit of combining inputs
Hidden layer = set of neurons that operate on the same set of inputs
Features = transformations of inputs, such as x^2
Feature engineering = coming up with what transformations to include

What about images? How does it work?

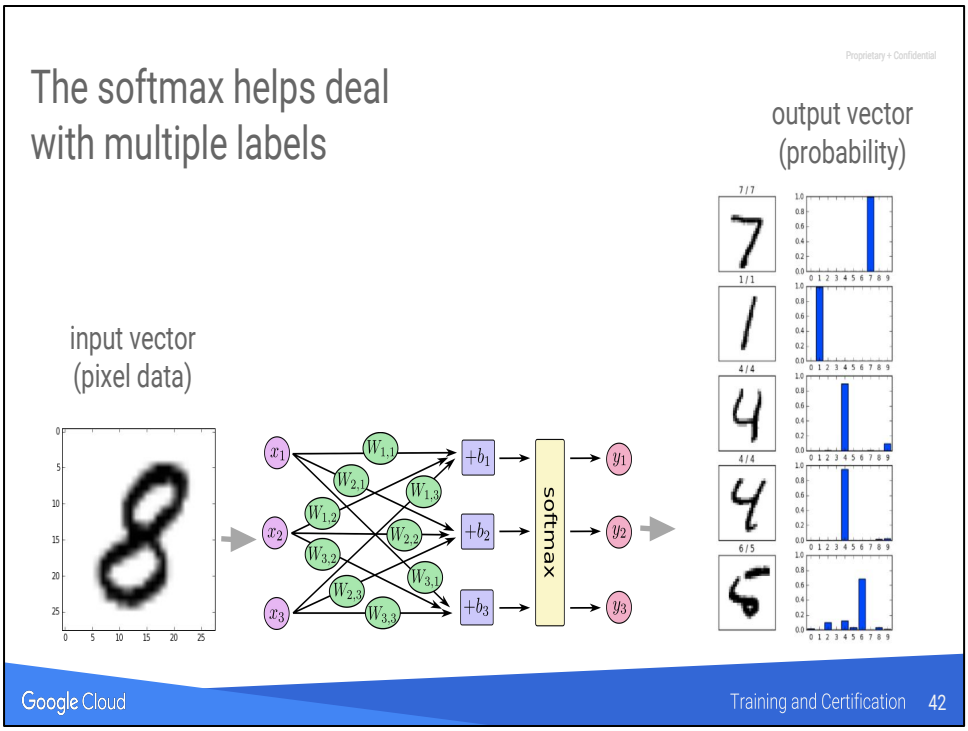
Each pixel value is an input



28 x 28 gray scale image =

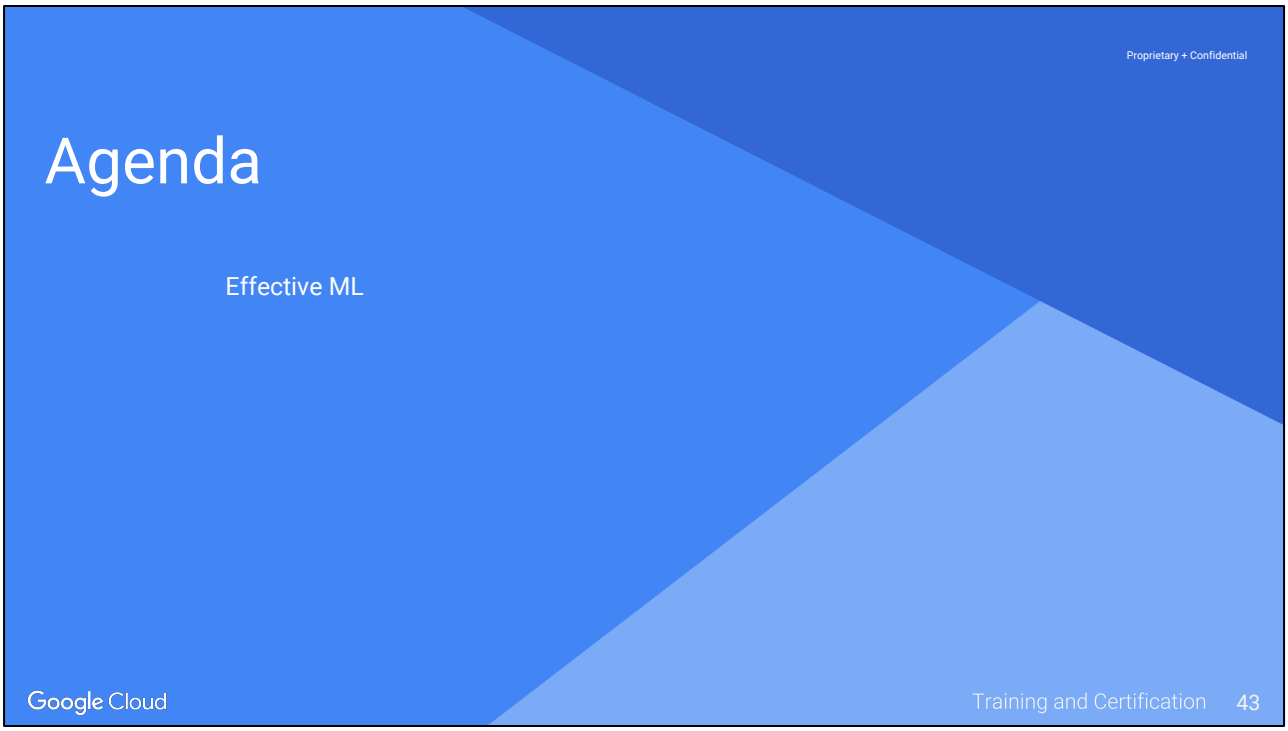
784 numbers

$$\sum_{i}^{n} w_i x_i > b$$



Notes:

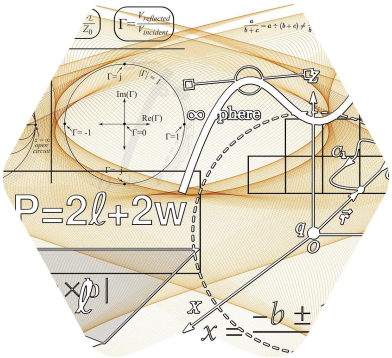
The softmax essentially normalizes the labels so that total probability is 1.0. It also emphasizes the peaks more than just a simple divide by sum.



20-40 minutes based on audience.

The popular imagination of what ML is

Lots of data



Complex mathematics in multidimensional spaces

Magical results

Notes:

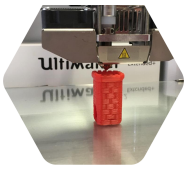
https://pixabay.com/en/x-y-mathematics-equation-937883/ (cc0)
https://pixabay.com/en/mathematics-formula-physics-school-1233876/ (cc0)
https://pixabay.com/en/deer-statue-silhouette-animal-stag-732122/ (cc0)

In reality, ML is...

Collect data

Organize data

Create model



Use machines to flesh out the model from data

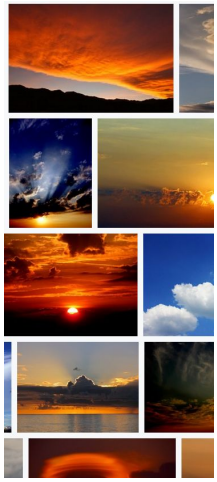
Deploy fleshed out model

Notes:

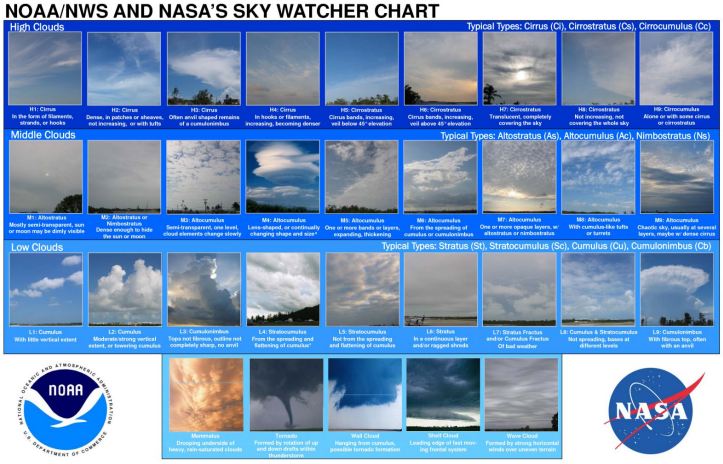
https://pixabay.com/en/ant-brown-carrying-egg-white-44588/ (cc0)
https://pixabay.com/en/tile-organization-exterior-materials-846016/ (cc0)
https://pixabay.com/en/deer-dream-animal-fantasy-1333814/ (cc0), cropped
https://pixabay.com/en/printer-3d-pressure-3d-printing-1455169/ (cc0)
https://pixabay.com/en/deer-statue-silhouette-animal-stag-732122/ (cc0)

On GCP, we can use:
Logging APIs, Cloud Pub/Sub, etc. and other real-time streaming to collect the data.
BigQuery, Dataflow and ML preprocessing SDK to organize the data [different types of organization].
TensorFlow to create the model.
Cloud ML to train, deploy the model.
The magic is still there …

The Dataset should cover all cases



All types covered?



Notes:

If you are going to recognize clouds, make sure you know what all types of clouds there are, and collect enough images of each type. Domain experts have to be involved in creating the dataset for ML. But this isn't enough … (see next slide)

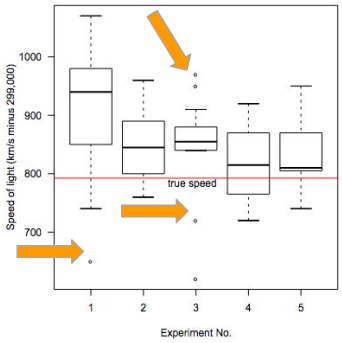
Image source:
The left-hand side is a snapshot of several cc0 images from Pixabay.
Right-hand side is from NOAA/NWS and NASA. so public domain.

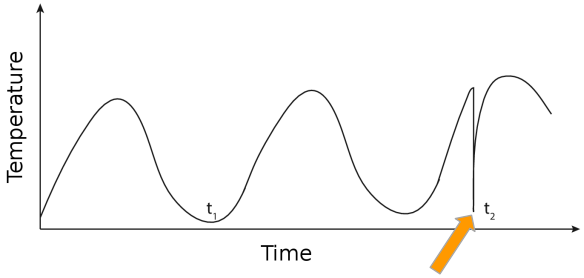
Negative examples and near-misses

Notes:

https://pixabay.com/en/clouds-sky-blue-cumulus-white-34027/ (cc0) cartoon cloud
https://pixabay.com/en/sheep-agriculture-animals-17482/ (cc0) flock of sheep
https://pixabay.com/en/scheuchzers-cottongrass-175409/ (cc0) field of cotton
https://pixabay.com/en/marble-background-backdrop-1006628/ (cc0) floor tile texture is cloud-like
https://pixabay.com/en/rocket-launch-steam-smoke-trail-693270/ (cc0) is a steam from a rocket
https://pixabay.com/en/chemtrail-conspiracy-theory-contrail-822000/ (cc0) is a chemtrail from plane

Explore the data you have; fix problems



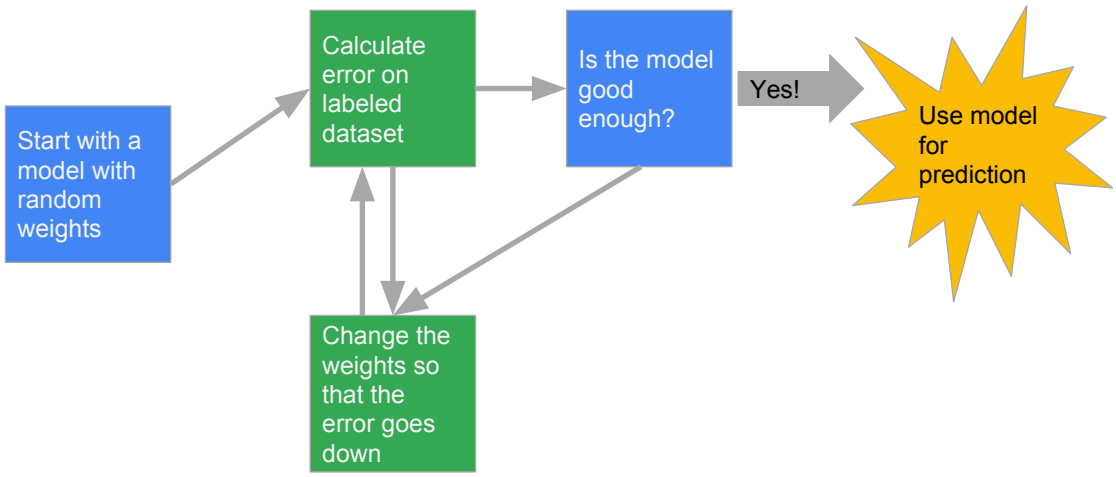


Notes:

Figure out what happened at the marked data points. It might be instrument errors, or worse.

Images from wikipedia page on outliers.

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.

Notes:

Image: https://pixabay.com/en/gallop-horse-horse-racing-1117183/

Regression error: How far from true outcome?

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

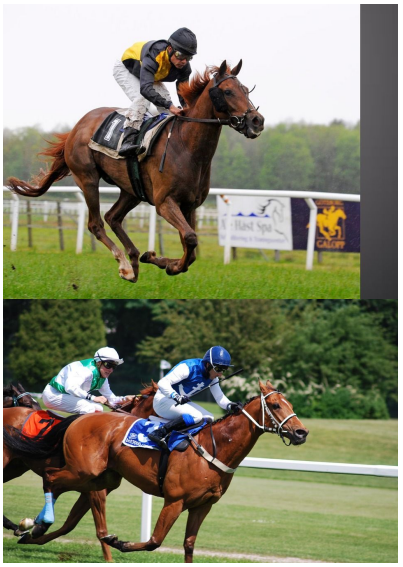
Error = truth - prediction
= -$14 - $89 = -$103

Notes:

Image: https://pixabay.com/en/gallop-horse-horse-racing-1117183/

Calculating regression error—Outcomes

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



Notes:

Top Right Image: https://pixabay.com/en/gallop-horse-horse-racing-1117183/
Bottom Right Image: https://pixabay.com/en/race-horses-racecourse-744105/

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

Notes:

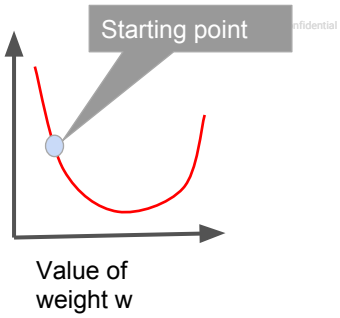
Image: https://pixabay.com/en/gallop-horse-horse-racing-1117183/

Mathematically...

Y-cap is the model estimate
Y is the labeled value
Mean Square Error (MSE) is:

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

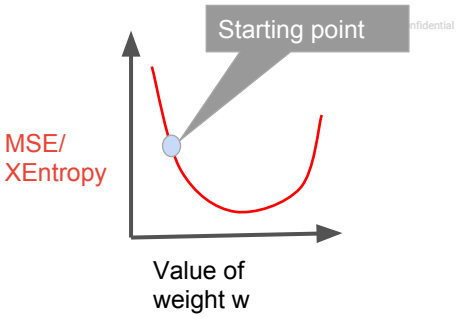
MSE



Notes:

Mathematically speaking, we prefer differentiable error measures so that we can do gradient descent. Mean-square error is differentiable.

For classification problems, we use cross-entropy



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

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

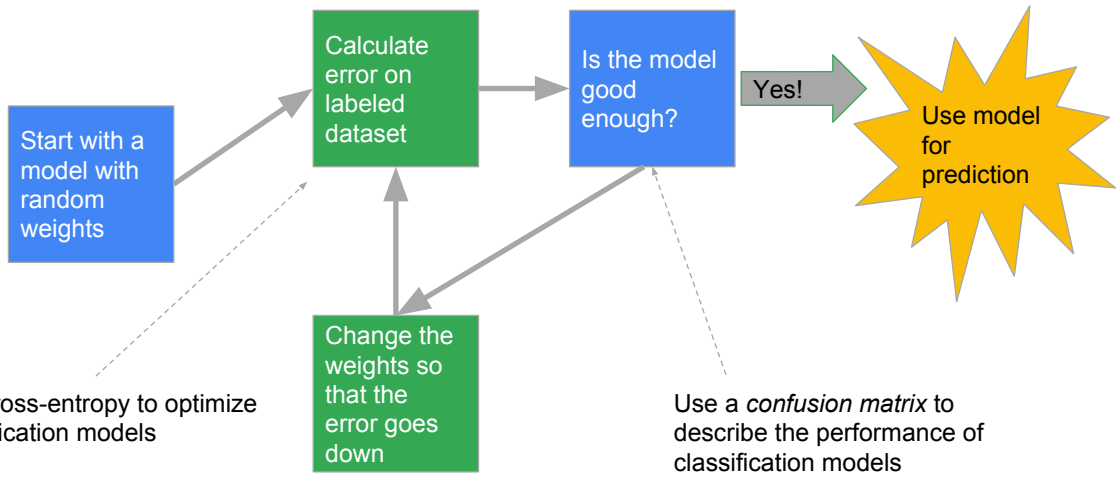
Notes:

Mathematically speaking, we prefer differentiable error measures so that we can do gradient descent. So, for classification, we'll use cross-entropy.

Why cross-entropy for categorical variables? See: https://en.wikipedia.org/wiki/Cross_entropy#Cross-entropy_error_function_and_logistic_regression.

In non-mathematical terms: You use a neural network whose output node is a soft-max (so that outputs are restricted to [0,1]) and whose objective function is cross-entropy. The result of the NN can be treated as a probability. So if you train using data on whether a customer bought your product or did not buy the product and you train on that data using the right NN structure, the trained NN can be used to estimate the probability that a visitor to your website will buy your product! This is hugely useful.

Cross-entropy is not intuitive to business users



Use cross-entropy to optimize classification models

Use a *confusion matrix* to describe the performance of classification models

Confusion matrix

1) Get the outcomes

TN

FP

TP

FN

TP

FN

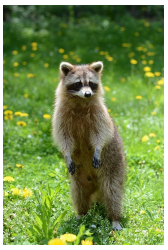
		ML System Says	
		Cat	No Cat
Truth	Cat	True Positive #TP	False Negative #FN
	No Cat	False Positive #FP	True Negative #TN

Notes:

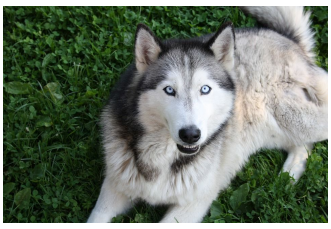
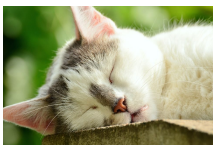
Get the outcome for each image in dataset, then add them up
Image: clipart https://pixabay.com/en/cartoon-cat-cute-1292872/
Confusion matrix is great, but it is 4 numbers and business decision makers want to see only one. Which one?

Accuracy, precision, and recall

Classify the cats!




Notes:

Image: cat with shoes courtesy of course author, Cassie Kozyrkov
Images: https://pixabay.com/en/cat-cat-face-sleep-exhausted-1551810/
https://pixabay.com/en/cat-pet-mirror-697113/
https://pixabay.com/en/tiger-sumatran-sumatran-tiger-164905/
https://pixabay.com/en/cat-wink-funny-fur-animal-red-1333926/
https://pixabay.com/en/huskies-dog-animal-blue-eyes-view-1370230/
https://pixabay.com/en/goldhamster-hamster-animal-nuts-943373/
https://pixabay.com/en/racoon-animal-garden-summer-1453600/

Hypothetical results from ML

Notes:

Image: cat with shoes courtesy of course author, Cassie Kozyrkov
Images: https://pixabay.com/en/cat-cat-face-sleep-exhausted-1551810/
https://pixabay.com/en/cat-pet-mirror-697113/
https://pixabay.com/en/tiger-sumatran-sumatran-tiger-164905/
https://pixabay.com/en/cat-wink-funny-fur-animal-red-1333926/
https://pixabay.com/en/huskies-dog-animal-blue-eyes-view-1370230/
https://pixabay.com/en/goldhamster-hamster-animal-nuts-943373/
https://pixabay.com/en/racoon-animal-garden-summer-1453600/
https://pixabay.com/en/cartoon-cat-cute-1292872/

Accuracy is fraction correct

Accuracy = 3 / 8
= 0.375

Notes:

Image: cat with shoes courtesy of course author, Cassie Kozyrkov
Images: https://pixabay.com/en/cat-cat-face-sleep-exhausted-1551810/
https://pixabay.com/en/cat-pet-mirror-697113/
https://pixabay.com/en/tiger-sumatran-sumatran-tiger-164905/
https://pixabay.com/en/cat-wink-funny-fur-animal-red-1333926/
https://pixabay.com/en/huskies-dog-animal-blue-eyes-view-1370230/
https://pixabay.com/en/goldhamster-hamster-animal-nuts-943373/
https://pixabay.com/en/racoon-animal-garden-summer-1453600/
https://pixabay.com/en/cartoon-cat-cute-1292872/
https://pixabay.com/en/check-mark-tick-mark-check-correct-1292787/
https://pixabay.com/en/incorrect-delete-remove-cancel-red-294245/

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!

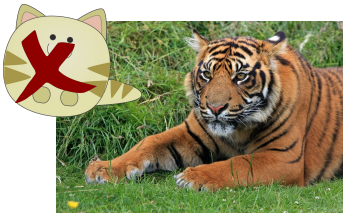
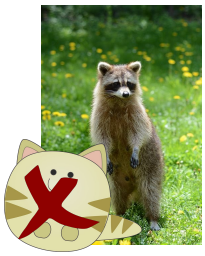
Notes:

Accuracy is fine for well-balanced datasets, but the dataset has a lot more of one category than another, we need to dig a little deeper into the performance. Use precision and recall for that. Suppose you have 1000 people and only 10 thieves. The ML can achieve 99.9% accuracy by classifying everything as “nice-people”. This is where it is helpful to know that recall=0.

https://pixabay.com/en/parking-autos-vehicles-traffic-825371/ (cc0)

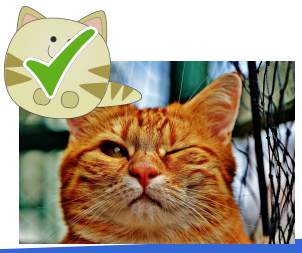
Precision = Positive Predictive Value

Accuracy when ML says “cat”


TP + FP = 5

Precision = TP / (TP + FP)
= 2 / 5 = 0.40



Notes:

Ask them what the precision is for the parking-lot example.
Answer: Depends on what you define as “positive”.
We have to define “available” as positive since that’s what we want to predict
...
Answer: Precision = 1/1 = 1 because all our predictions of “available” are correct.
When we are sure, we are sure.

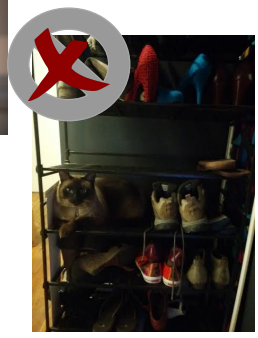
Image: cat with shoes courtesy Cassie Kozyrkov, Google
Images: https://pixabay.com/en/cat-cat-face-sleep-exhausted-1551810/
https://pixabay.com/en/cat-pet-mirror-697113/
https://pixabay.com/en/tiger-sumatran-sumatran-tiger-164905/
https://pixabay.com/en/cat-wink-funny-fur-animal-red-1333926/
https://pixabay.com/en/huskies-dog-animal-blue-eyes-view-1370230/
https://pixabay.com/en/goldhamster-hamster-animal-nuts-943373/
https://pixabay.com/en/racoon-animal-garden-summer-1453600/
https://pixabay.com/en/cartoon-cat-cute-1292872/
https://pixabay.com/en/check-mark-tick-mark-check-correct-1292787/
https://pixabay.com/en/incorrect-delete-remove-cancel-red-294245/

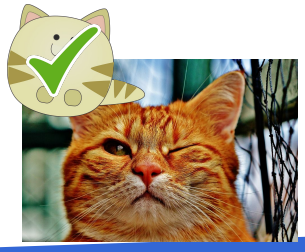
Recall is true positive rate

Recall = fraction of cats ML finds

TP + FN = 4

Recall = TP / (TP + FN)
= 2 / 4 = 0.50





Notes:

Ask them what the recall is for the parking-lot example.
Answer: Recall = 1/10 = 0.1

Image: cat with shoes courtesy Cassie Kozyrkov, Google
Images: https://pixabay.com/en/cat-cat-face-sleep-exhausted-1551810/
https://pixabay.com/en/cat-pet-mirror-697113/
https://pixabay.com/en/tiger-sumatran-sumatran-tiger-164905/
https://pixabay.com/en/cat-wink-funny-fur-animal-red-1333926/
https://pixabay.com/en/huskies-dog-animal-blue-eyes-view-1370230/
https://pixabay.com/en/goldhamster-hamster-animal-nuts-943373/
https://pixabay.com/en/racoon-animal-garden-summer-1453600/
https://pixabay.com/en/cartoon-cat-cute-1292872/
https://pixabay.com/en/check-mark-tick-mark-check-correct-1292787/
https://pixabay.com/en/incorrect-delete-remove-cancel-red-294245/

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

Term	Meaning
MSE	
Cross-entropy	
Accuracy	
Precision	
Recall	

Notes:

Mean Square Error: the loss measure for regression problems
Cross-entropy: the loss measure for classification problems
Accuracy: A more intuitive measure of skill for classifiers
Precision: Accuracy when classifier says “yes” (useful for unbalanced classes where there are many more yes-es than no-es)
Recall: Accuracy when the truth is “yes” (useful for unbalanced classes where there are very few yes-es)
ROC curve: a way to pick the threshold (of the probability that is output by the classifier) at which a specific precision or recall is reached. The area under the curve (AUC) is a threshold-independent measure of skill.

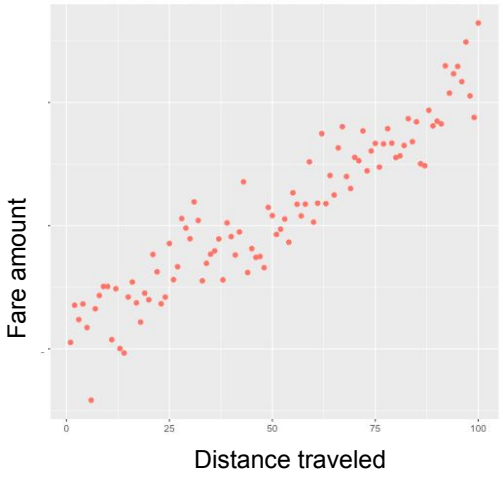
Agenda

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?



Notes:

Simulation by Cassie Kozyrkov, Google.

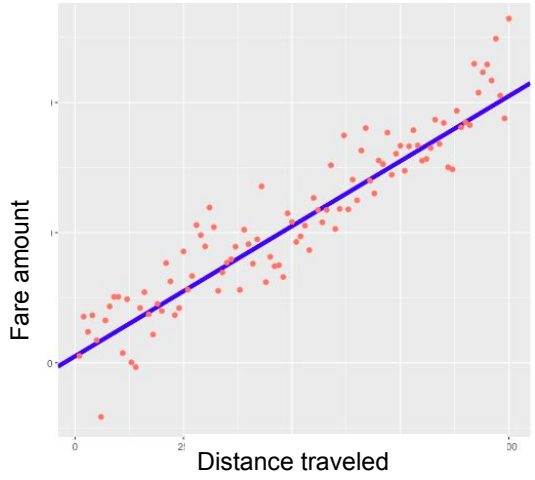
Regression problem, so typically MSE or RMSE.

Model 1

Red = training data

Blue = model prediction for each distance traveled

RMSE = 22.24



Notes:

Simulation by Cassie Kozyrkov, Google.

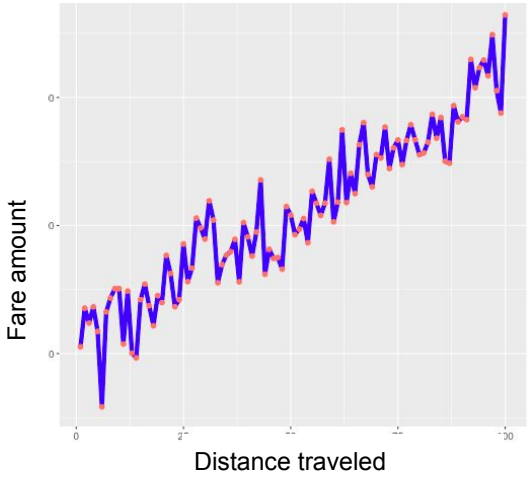
This is a linear model, so linear regression.

Model 2 has more free parameters

RMSE = 0

Which model is better?

How can we tell?



Notes:

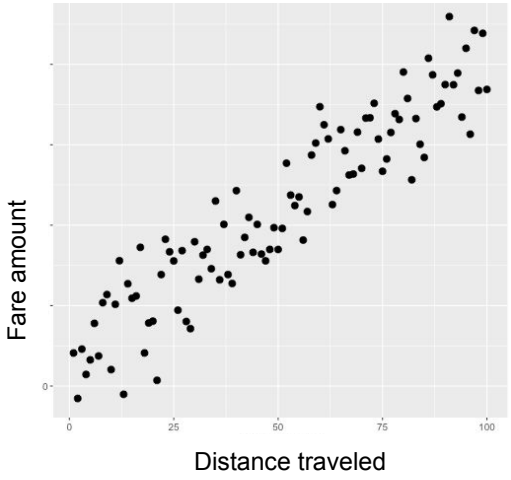
Simulation by Cassie Kozyrkov, Google.

This might be a polynomial of 100th order or a neural network with lots of nodes. A more complex model has more parameters that can be optimized. This can help is fit more complex data. But it might also help it memorize simpler datasets.

People intuitively feel that there is something fishy about Model 2.
But how can we tell? In ML, we often have lots of data, and no such intuition. Is a NN with 8 nodes better than a NN with 12 nodes? The 12 nodes has lower RMSE … so should we pick it? But what if we try 16 nodes, and it has even lower RMSE? At what point do we stop and say a model is now simply memorizing?

Does the model generalize to new data?

Need data that were not used in training



Notes:

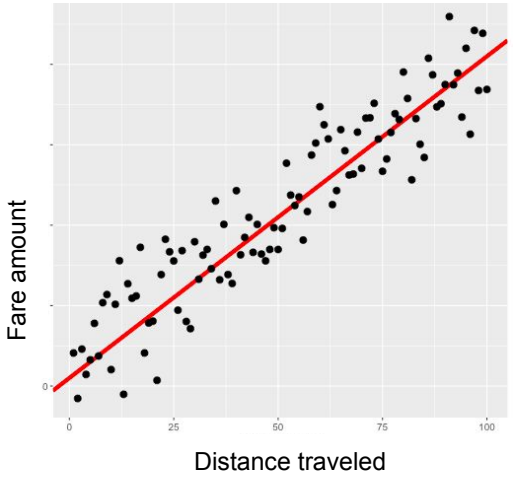
Simulation by Cassie Kozyrkov, Google.

Model 1 generalizes well

Old RMSE = 22.24

New RMSE = 21.98

Pretty similar = good



Notes:

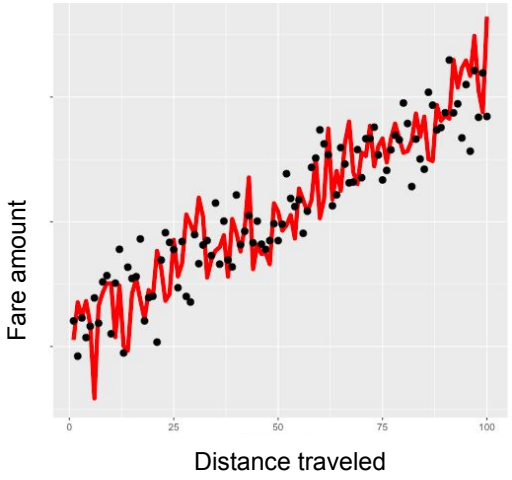
Simulation by Cassie Kozyrkov, Google.

Model 2 does not generalize well

Old RMSE = 0

New RMSE = 32

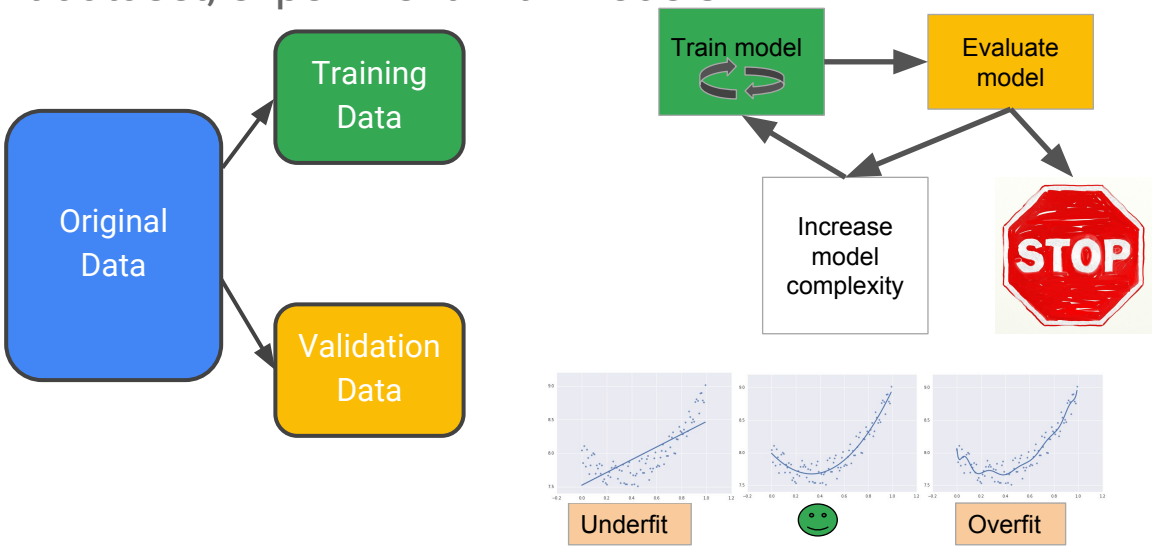
This is a red flag



Notes:

Simulation by Cassie Kozyrkov, Google.

Split dataset, experiment with models

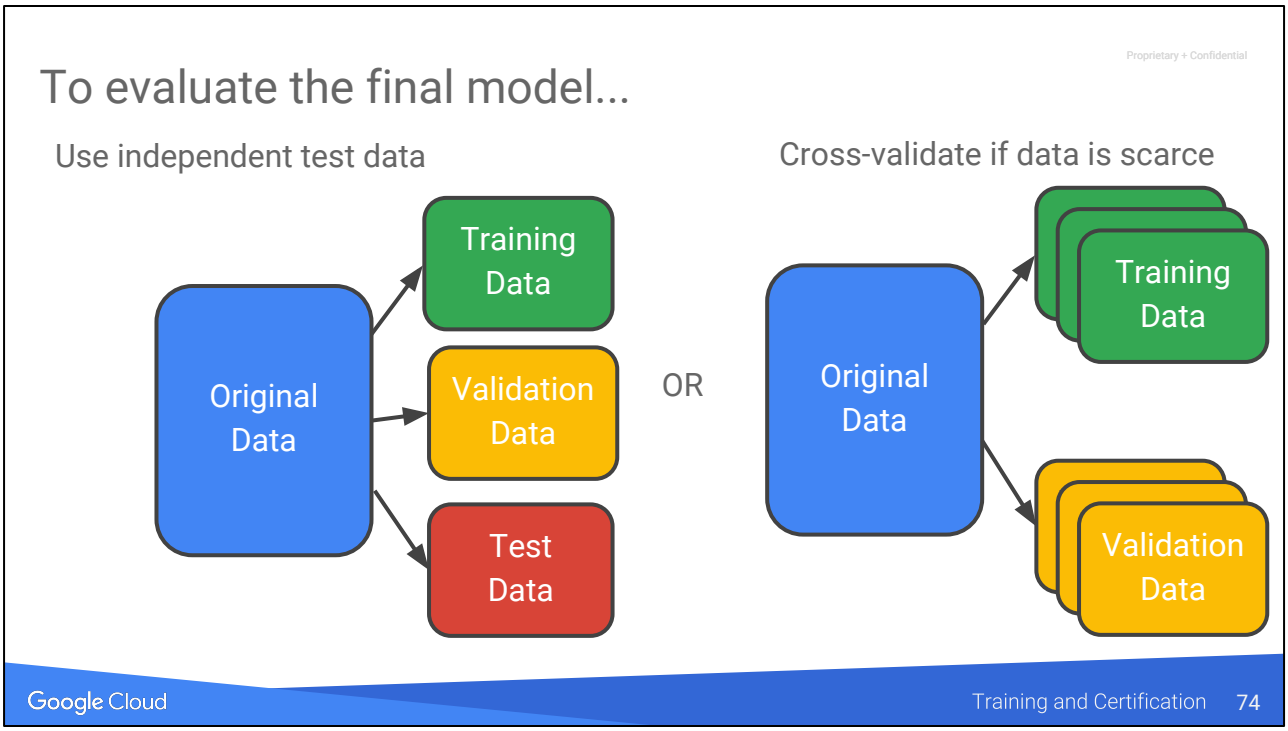


Notes:

https://pixabay.com/en/red-stop-sign-road-road-sign-1315030/ (cc0)

Use the validation data to tell you when to stop increasing model complexity. Graphs by course author in Python

Remember that “Train model” involves iterative optimization over training dataset, so this experimentation is akin to an outer loop. This point is useful to get the point across of why Cloud ML is needed.

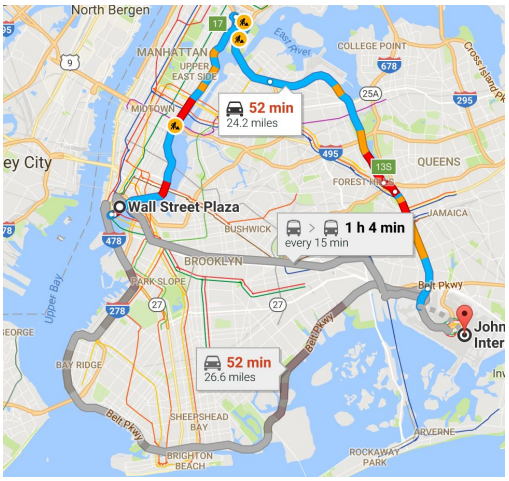


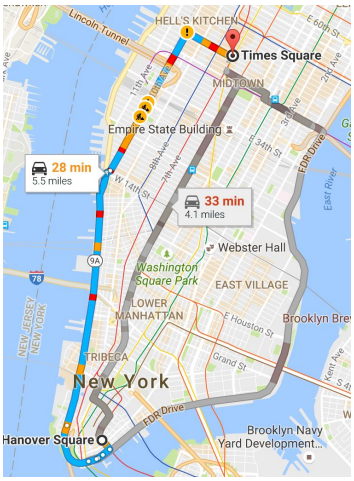
Notes:

Cannot just use the original validation dataset. That is no longer truly independent.
Cross-validation could be done in both cases on the ice cream cone example. If we do random sampling, it's just a matter of choosing a different seed. If we divide by days/month, it's just a matter of varying the cutoff day (it was the first 25 days in our example, but we could cycle through and take the middle 25 days). This also ensures that we don't throw away any data completely due to leakage.

Included in Option 2 is the idea of going ahead and using all of our data and allowing the passage of time to get us an independent test data.

Goal: To estimate taxi fare





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

Taxi fares:

$2.50 initial charge
+
50c per ⅕ mile
(or)
50c per minute if stopped
+
Passenger pays tolls
+
Various special charges

Notes:

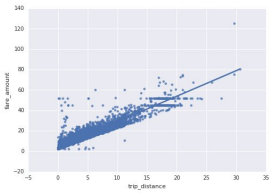
Left: a trip from Hanover Square to Time Square (downtown to midtown). Two routes -- the shorter one takes longer.

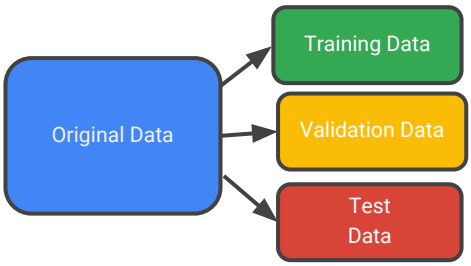
Right: a trip from JFK to Manhattan offers 2 routes, both very roundabout as compared to what a bus would take. You pay a toll to take the triborough bridge

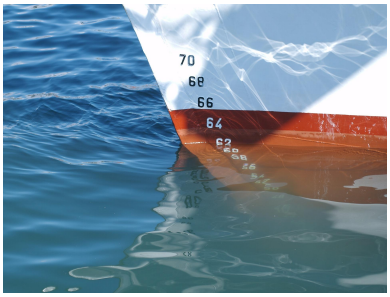
But assume we don't know any of these things. Instead of hardcoding a bunch of rules, let's try to infer the fare amount simply from the data. The point of ML is to program with data instead of rules.
https://pixabay.com/en/question-mark-question-faq-ask-1421013/ (cc0)

Lab: Serverless Machine Learning

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







1. Explore

2. Create Datasets

3. Benchmark

Notes:

These should form the first steps for any ML project that you undertake. You will often spend weeks just exploring the dataset to gain intuition into the problem -- this is crucial to creating good ML models. The benchmark phase should not be neglected; if you don't have a benchmark, you won't know what kind of performance you should seek to attain. Many times, errors can be detected simply by realizing that the performance of the ML model is nowhere near the benchmark.

https://pixabay.com/en/ship-sea-water-blue-water-level-1518522/ (cc0)


cloud.google.com
Image by Connie Zhou