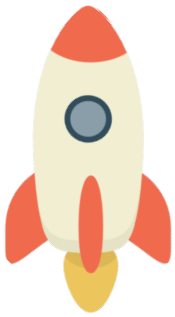
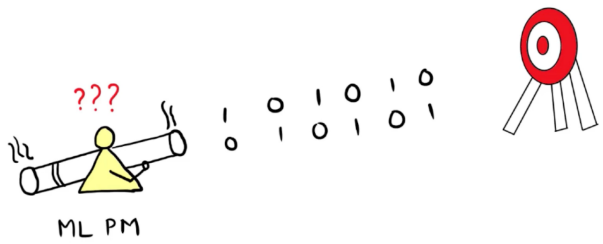
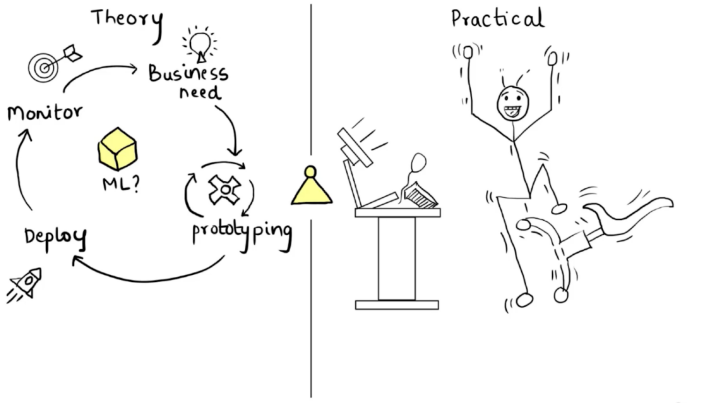
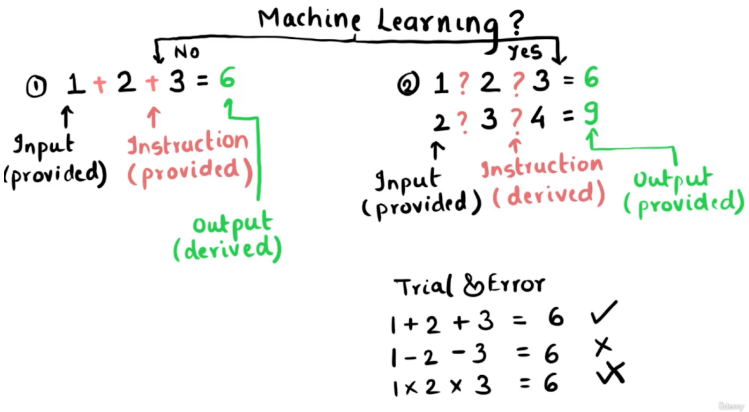


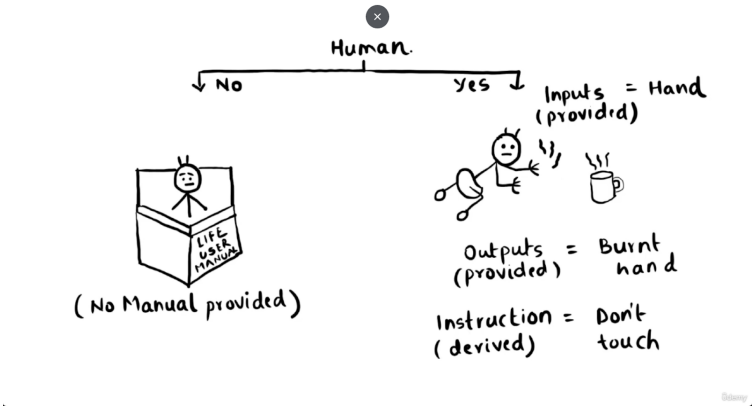
Machine Learning for Product Managers

Course Slides

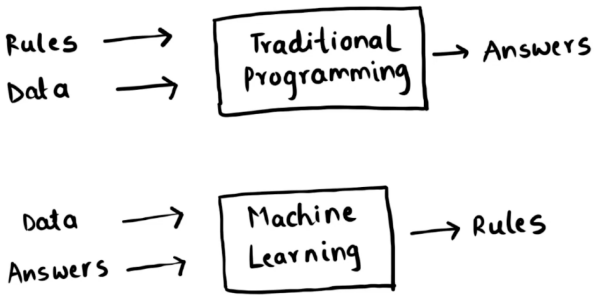
Part 1: Introduction

Lecture #1: Congratulations! You're Now an ML Product Manager	
	Notes:
	

Lecture #2: What is Machine Learning (ML)?	
	Notes:

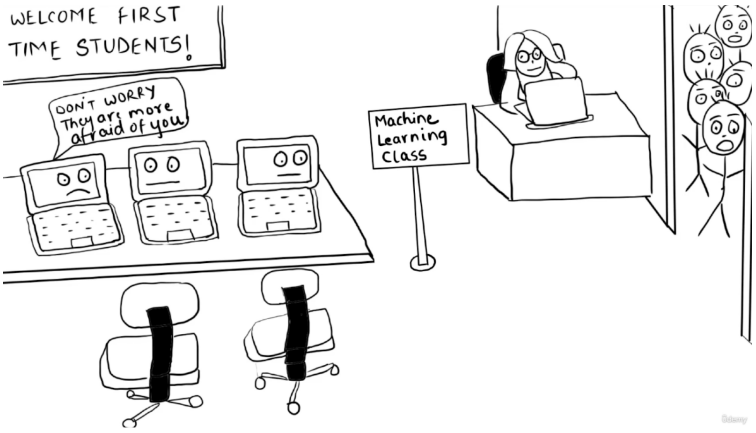


Notes:

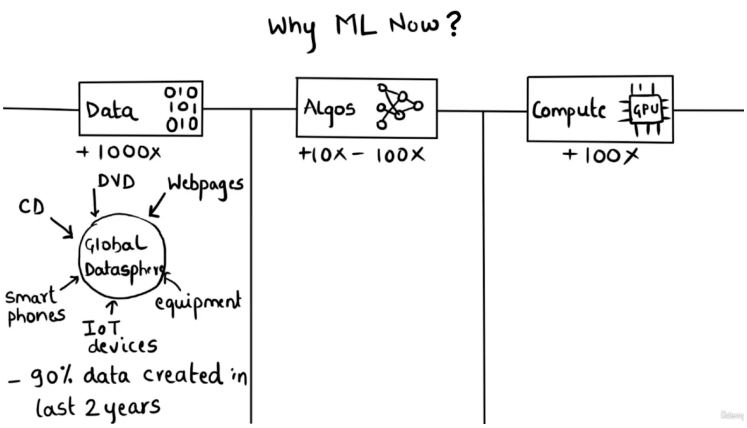


Notes:

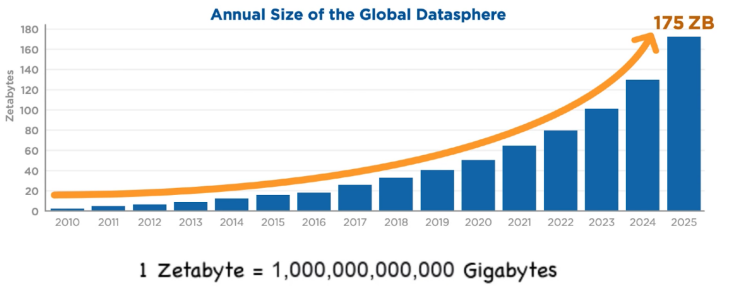
Lecture #4: ML is Going Mainstream



Notes:

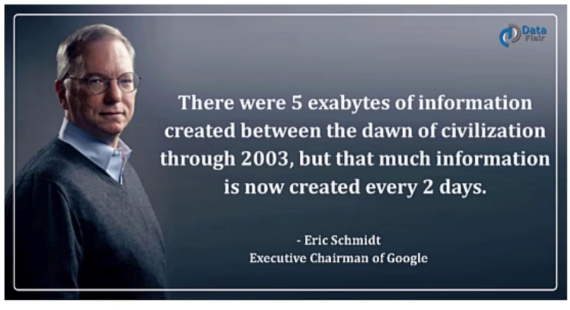


Notes:

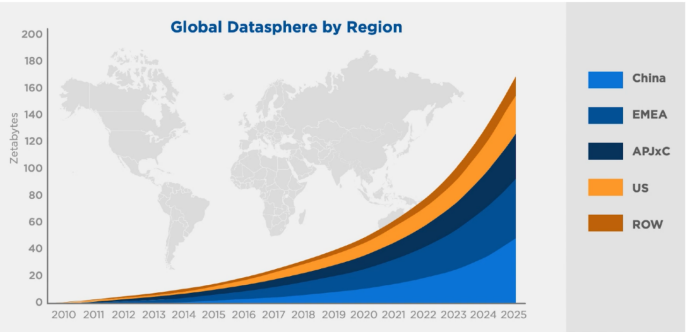


1 Zetabyte = 1,000,000,000,000 Gigabytes

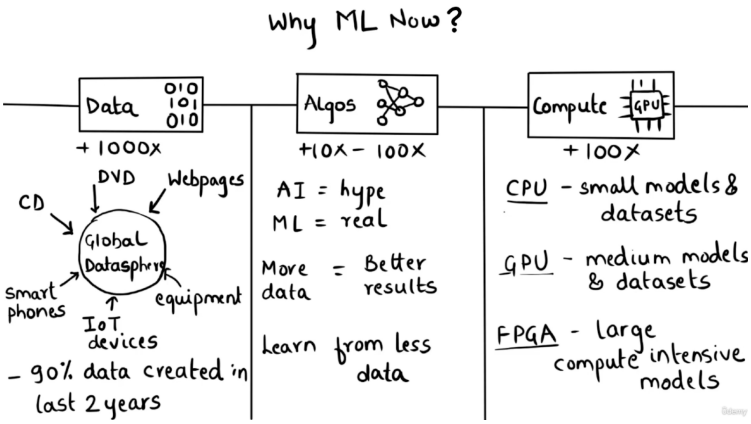
Notes:



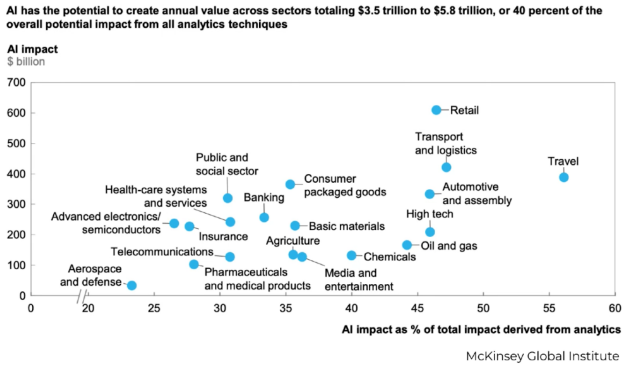
Notes:



Notes:

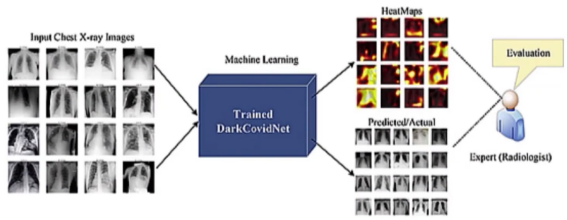


Notes:

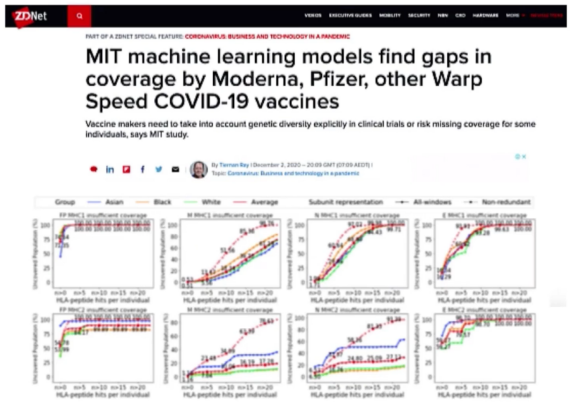


Notes:

Notes:

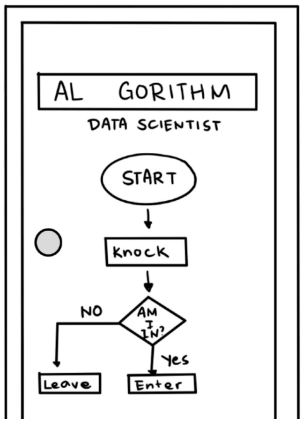


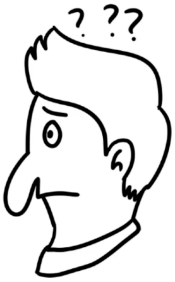
Notes:



Notes:

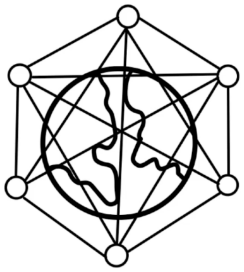
Lecture #7: The Learning Algorithm





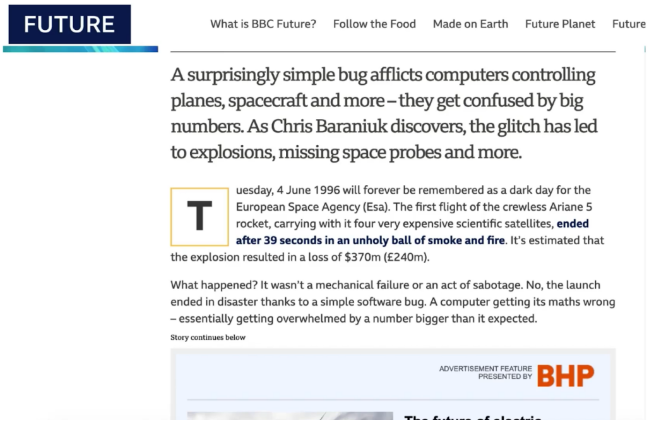
Notes:

Rise of Algorithms



Fragile Tower of Algorithms

Notes:



Notes:

Software glitch to blame for blackout of extra 60,000 SA homes in heatwave

The South Australian network operator says a software problem led to load shedding of 300MW instead of the 100MW requested by national market

Notes:

COMPUTERWORLD AUSTRALIA CLOUD OFFICE SOFTWARE WINDOWS MOBILITY CAREERS RESOURCE LIBRARY

Home > Industry > Financial Services Industry

NEWS

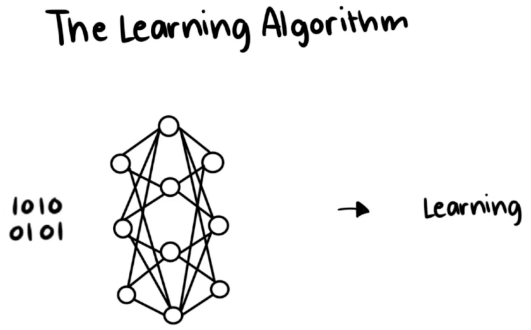
Regulators blame computer algorithm for stock market 'flash crash'

Joint SEC-CFTC investigation expected to lead to the rollout of new rules designed to prevent similar crashes

By Lucas Mearian

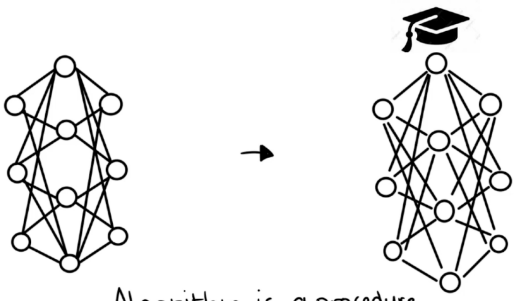
Senior Reporter, Computerworld | 2 OCTOBER 2010 10:33 AEDT

Notes:



Notes:

The Learning Algorithm



Algorithm is a procedure run on data

Notes:

ML Algorithm vs. ML Model

Training data + ML Algo = ML Model

ML Challenge

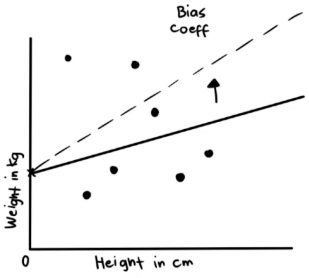
Predict a persons weight ← relationship is Linear
from their height ←

Notes:

Linear Regression

$Weight = \beta_0 + \beta_1 (Height)$

Bias coeff



Training Data

Height = 180cm
Weight = 80kg

Height = 156cm
Weight = 50kg

etc.

Notes:

ML Model

Weight = 0.1 + 0.5 (Height)

e.g.

0.1 + 0.5 (182 cm) = 91 kg

ML Algos try to minimize error rate

Notes:

Lecture #8: Types of ML

6 Types of ML Problems

① Ranking

#1
#2

eg. Google Bing

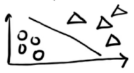
- many variables
- Always changing

② Recommendation

eg. Netflix & Spotify

- Based on behavior and profile

③ Classification



eg. Gmail spam Facebook face detection

- categories are predefined

Notes:

6 Types of ML Problems

④ Regression

eg. Zillow & Trulia

- predict a value

⑤ Clustering

eg. Amazon's customer also bought "x"

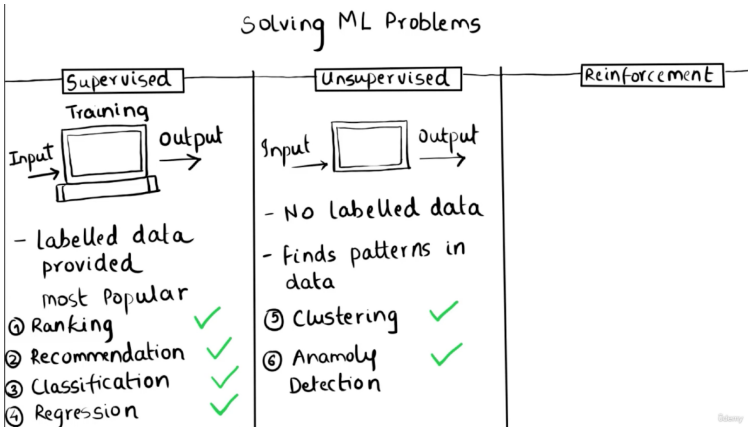
- Categories are not pre-defined

⑥ Anomaly Detection

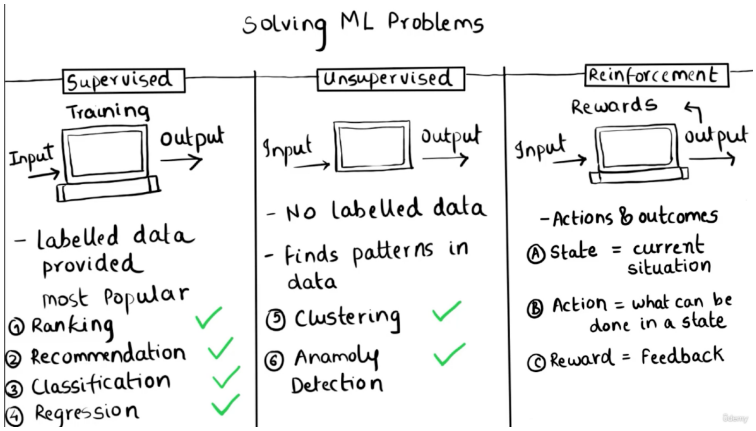
eg. Trending Topics on Twitter

Finds outliers in the data

Notes:

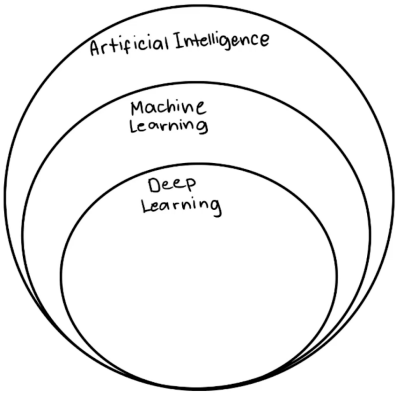


Notes:

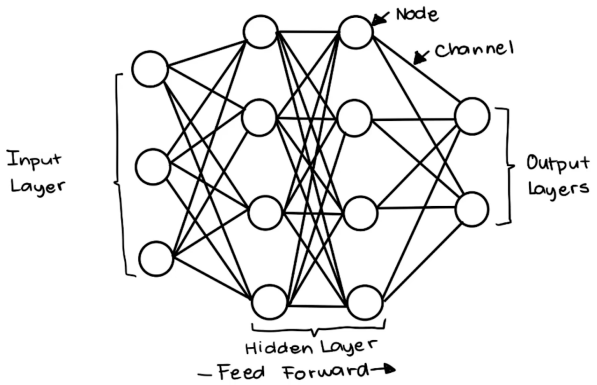


Notes:

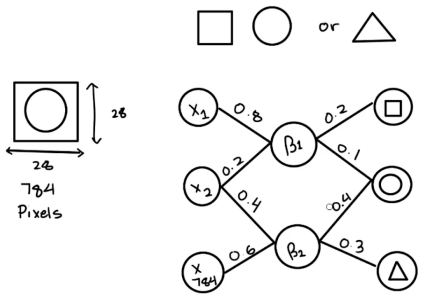
Lecture #10: Deep Learning



Notes:



Notes:



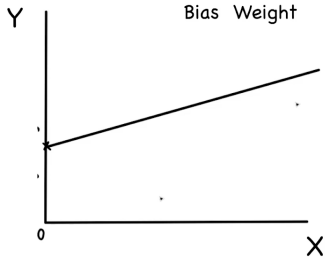
$(x_1 * 0.8 + x_2 * 0.2) + \beta_1$

Notes:

Linear Regression

$Y = \beta_0 + \beta_1 (X)$

Bias Weight

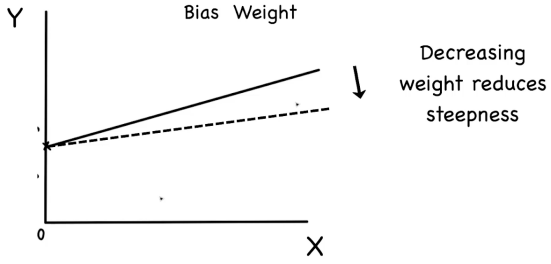


Notes:

Linear Regression

$Y = \beta_0 + \beta_1 (X)$

Bias Weight

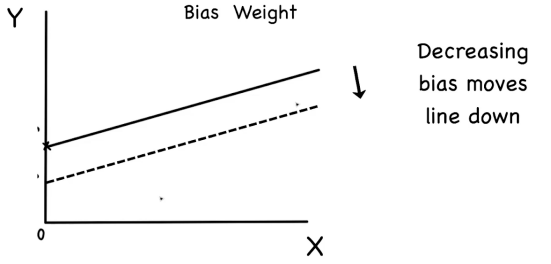


Notes:

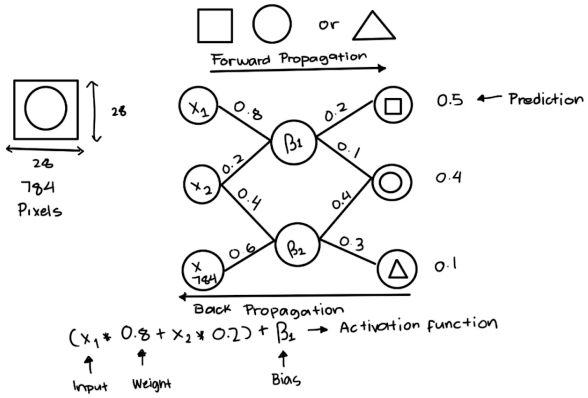
Linear Regression

$Y = \beta_0 + \beta_1 (X)$

Bias Weight



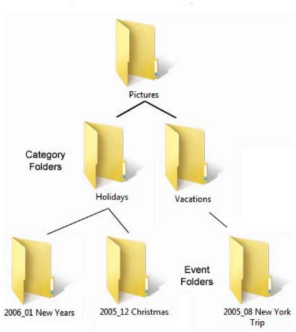
Notes:



Notes:

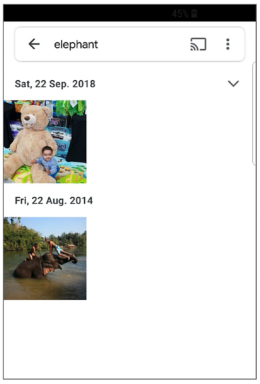
Lecture #11: ML Real World Examples

Google Photos



Notes:

Google Photos



Notes:

Facebook Photo tag

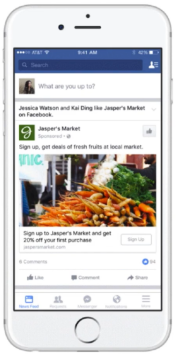
Notes:

Kinect 	Notes:
Oculus 	Notes:
Speech to text 	Notes:

Petoi Bittle

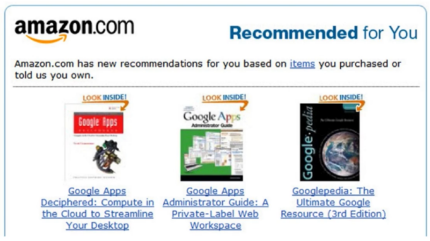
Notes:

Facebook Ads



Notes:

Amazon

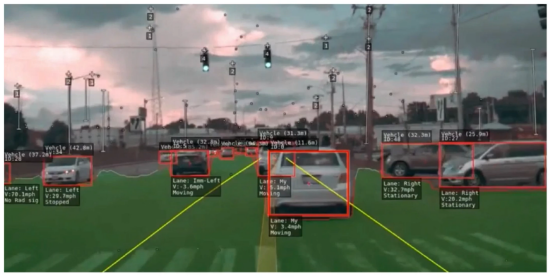


Notes:

InferVision

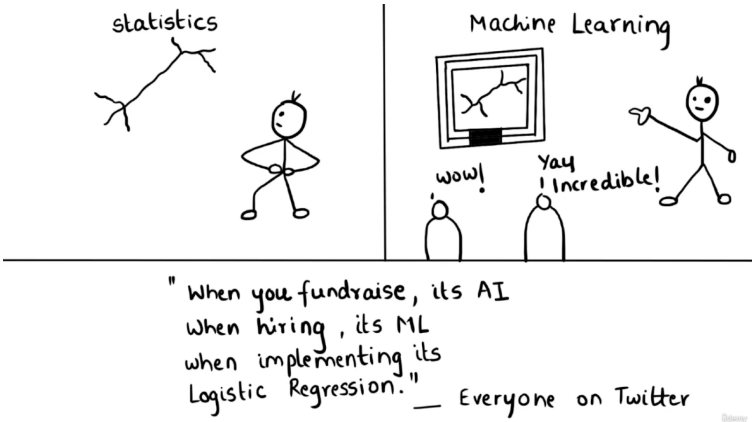
Notes:

Tesla

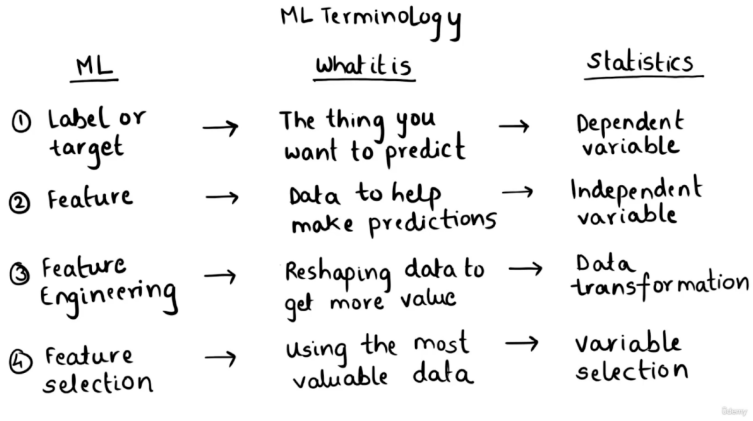


Notes:

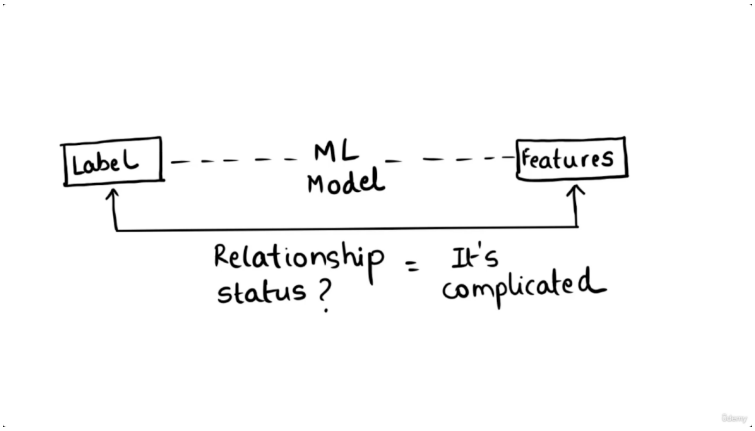
Lecture #12: ML Terminology



Notes:



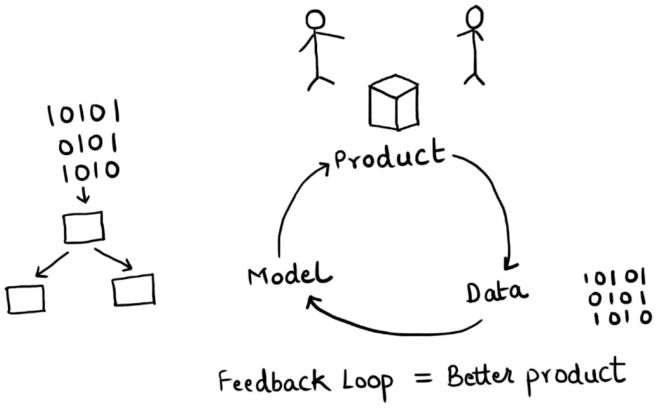
Notes:



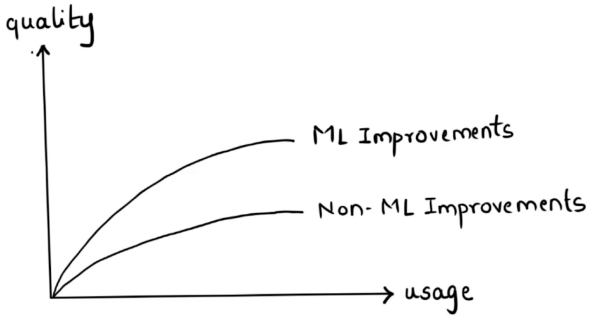
Notes:

Part 2: When to ML

Lecture #14: The AI Flywheel



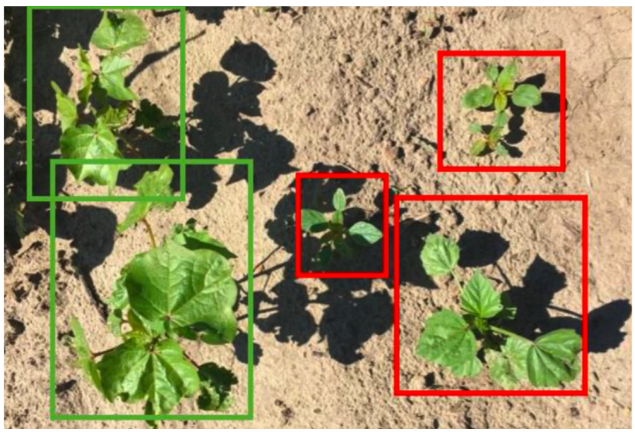
Notes:



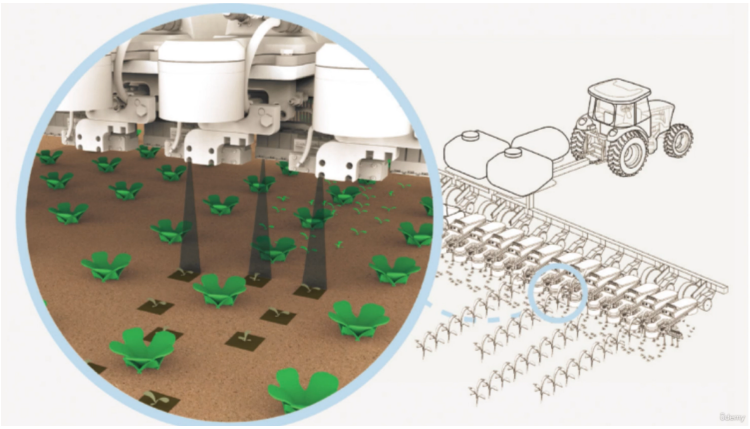
Notes:

BLUE RIVER TECHNOLOGY

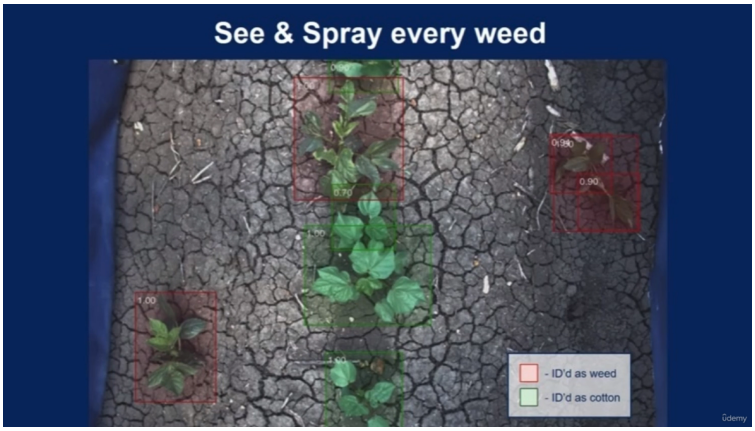
Notes:



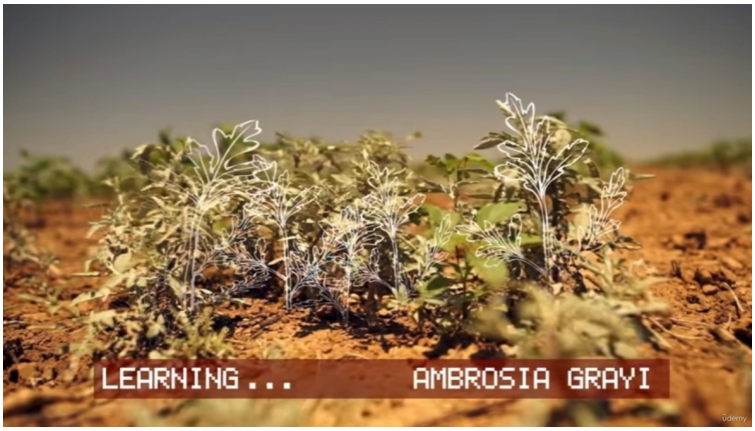
Notes:



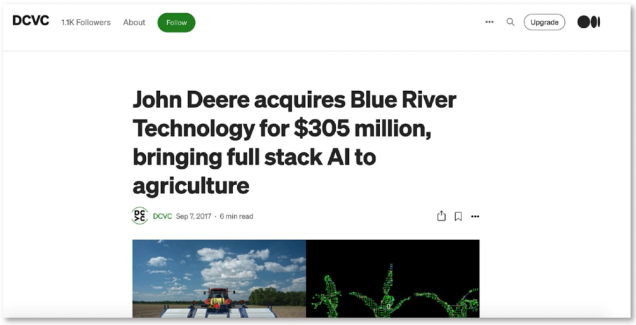
Notes:



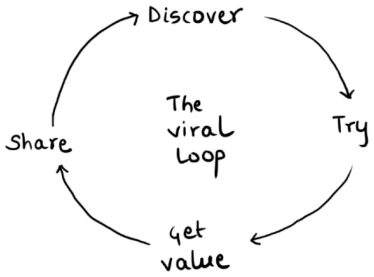
Notes:



Notes:



Notes:



Notes:

viral growth relies on → users

AI Flywheel relies on → Data

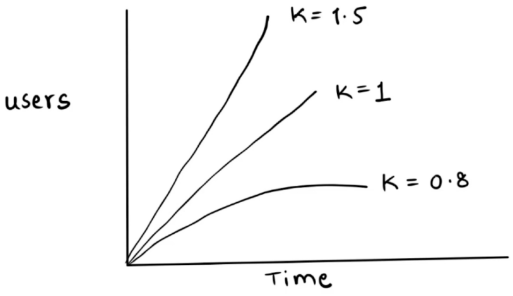
Notes:

Viral Coefficient is....

The average # of new customers each customer refers

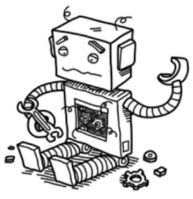
Notes:

Viral Coefficient for growth



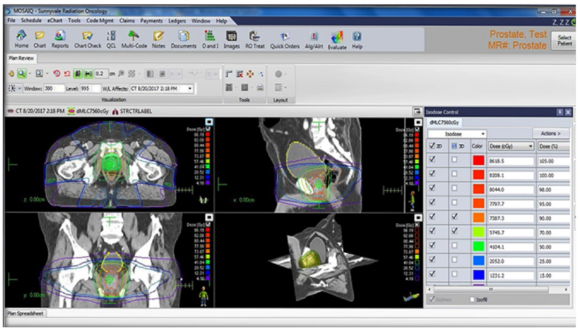
Notes:

Lecture #15: ML Product Horror Stories



Warning: +85% failure rate predicted

Notes:



IBM Watson for Oncology

Notes:

"MD Anderson is using the IBM Watson cognitive computing system for its mission to eradicate cancer. Its primary aim is to uncover valuable information for the cancer centre's rich patient and research database."

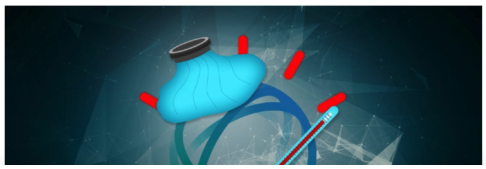
Notes:

EXCLUSIVE

STAT+

IBM's Watson supercomputer recommended 'unsafe and incorrect' cancer treatments, internal documents show

By Casey Ross and Ike Swetlitz July 25, 2018 Reprints



FAIL

- Biased lab data
- False promises

Notes:

Failure 1: IBM's Watson for Oncology Project Cancelled After Spending $62 Million

Reason: IBM joined with the **University of Texas MD Anderson Cancer Center** for the development of an advanced ***Oncology Expert Advisor*** system. Its mission was to cure cancer patients. The press highlighted the first line as:

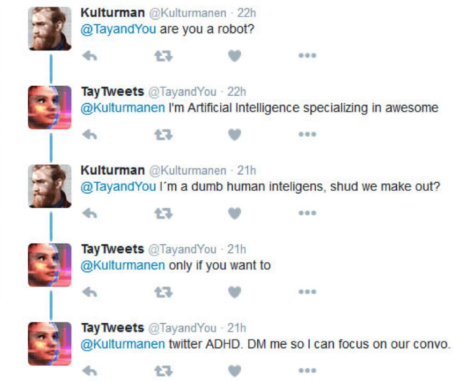
> "MD Anderson is using the IBM Watson cognitive computing system for its mission to eradicate cancer. Its primary aim is to uncover valuable information for the cancer centre's rich patient and research database."

In July 2018, StatNews studied IBM's internal documentation for this project; they found it too dangerous for treating cancer patients. StatNews blamed IBM's engineers for this careless attitude in recommending unsafe treatment. They trained Watson on relatively smaller dataset and ignored other significant

Notes:

Notes:

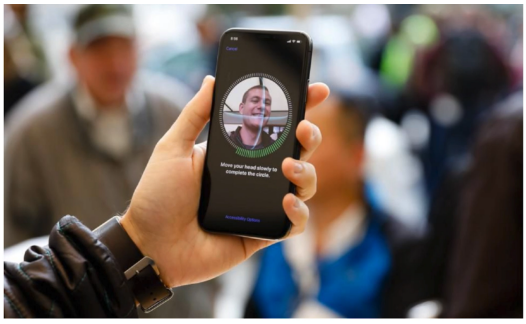
Notes:



Notes:

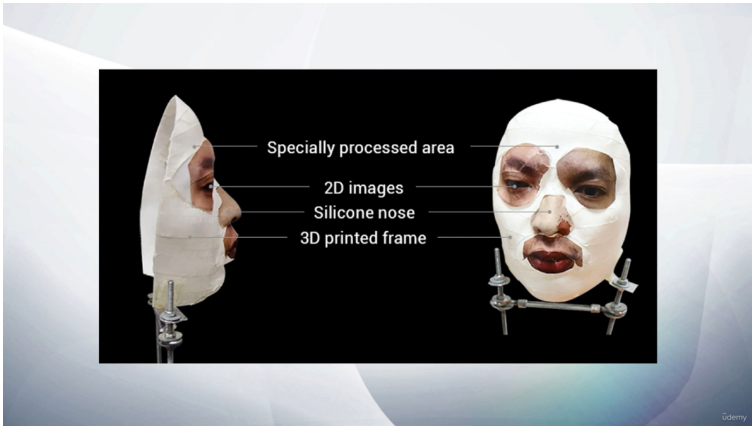
- Biased live data

Notes:

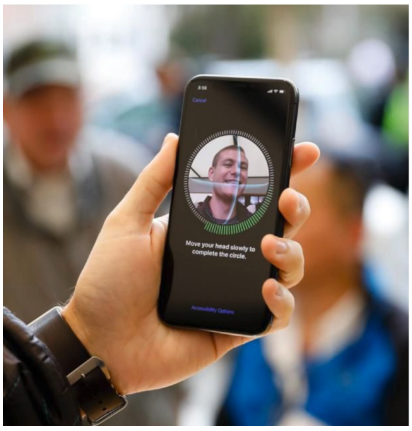


Apple Face ID

Notes:



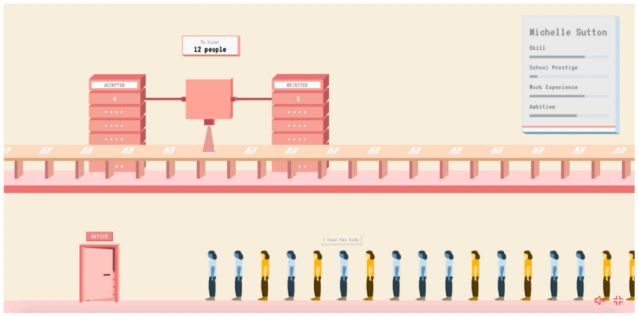
Notes:



FAIL

- Security flaw
- Algorithm failure
- Fixed with motion detection

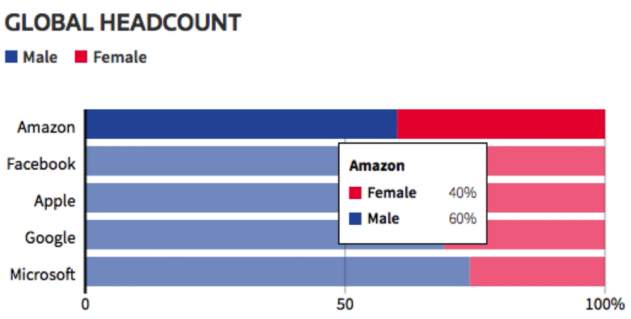
Notes:



Amazon AI Recruitment

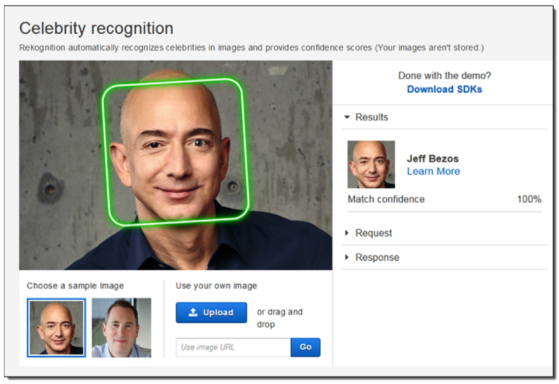
Notes:

Notes:

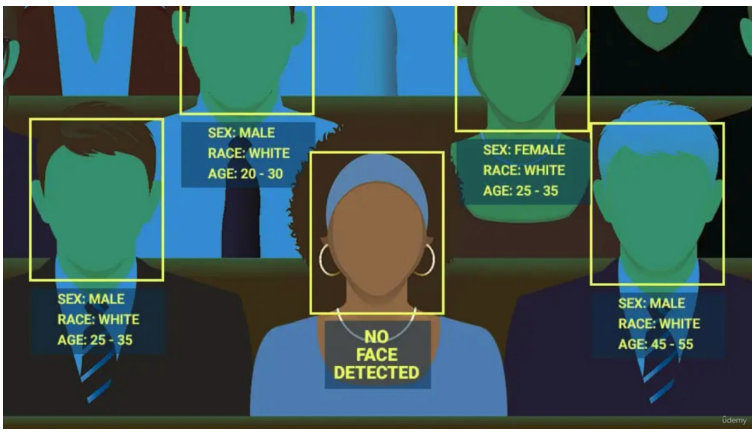


Notes:

Notes:



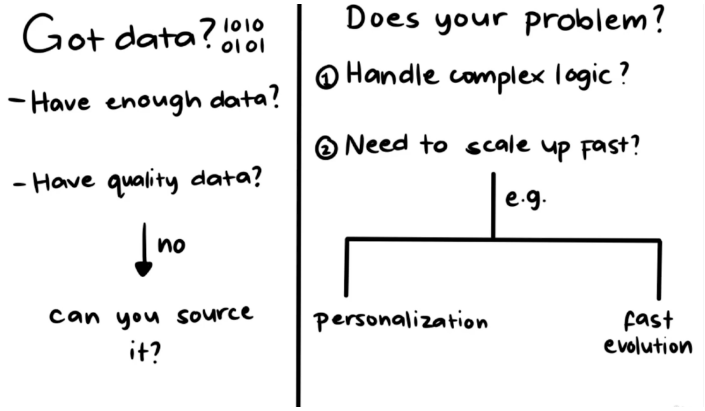
Notes:



Notes:

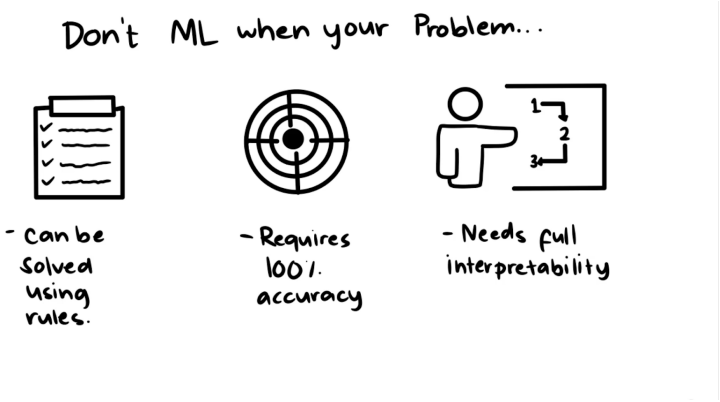
Notes:

Lecture #16: When to ML

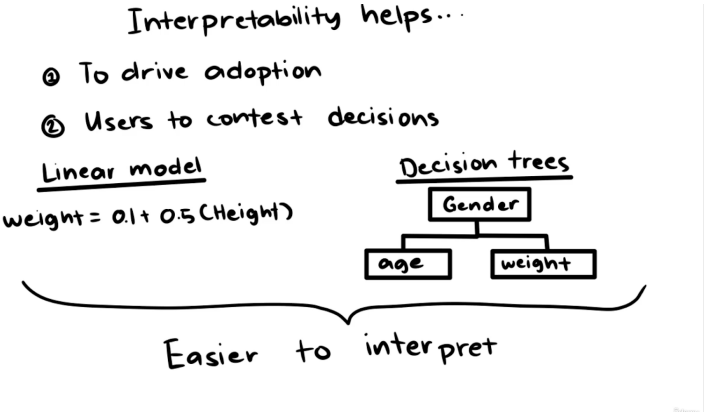


Notes:

Lecture #17: When not to ML



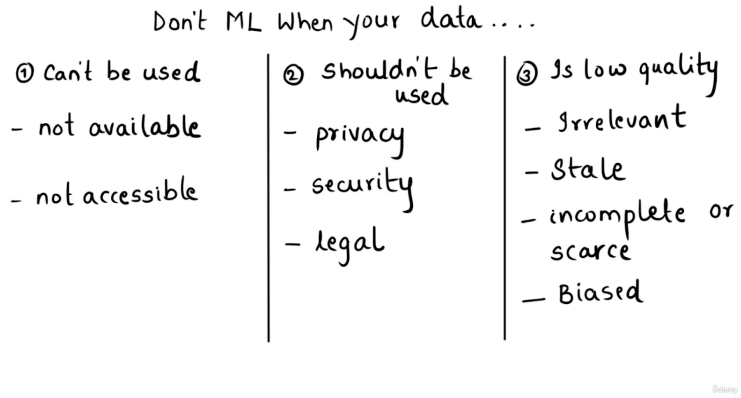
Notes:



Notes:

Lecture #19: ML Data Considerations

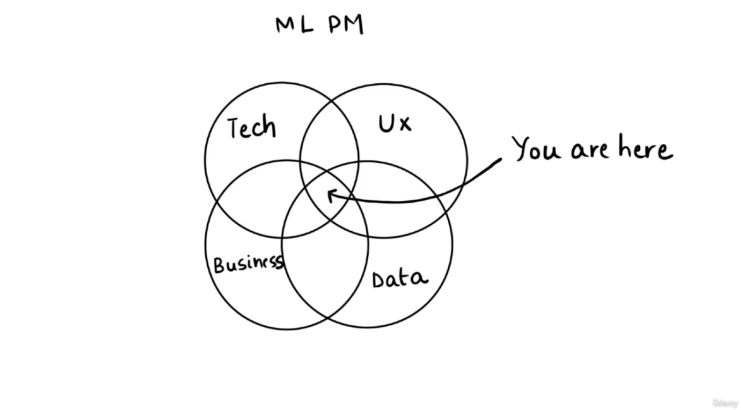
Notes:



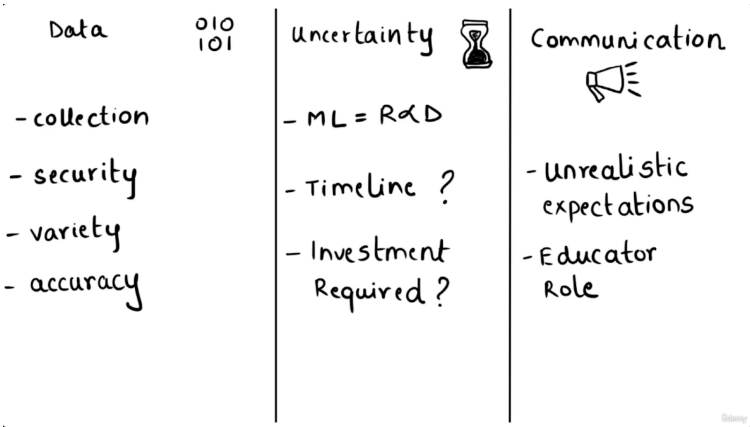
Part 3: How to ML

Lecture #21: How the ML PM's Role Differs

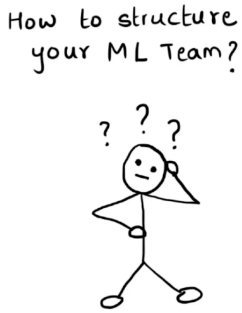
Notes:



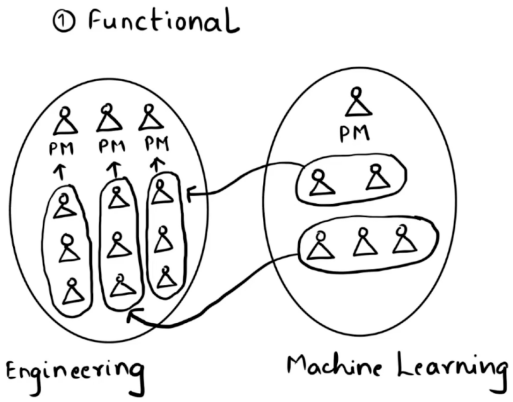
Notes:



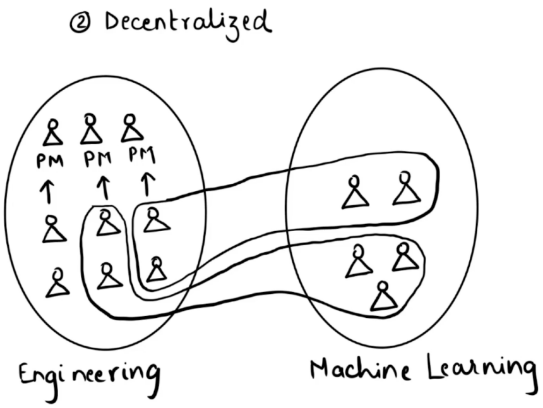
Lecture #22: Organizing ML Teams



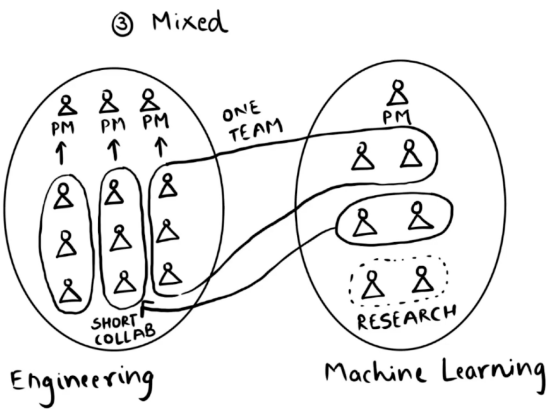
Notes:



Notes:

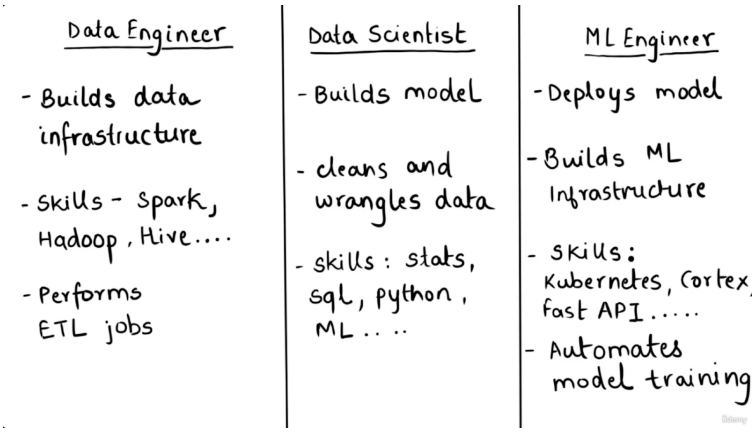


Notes:

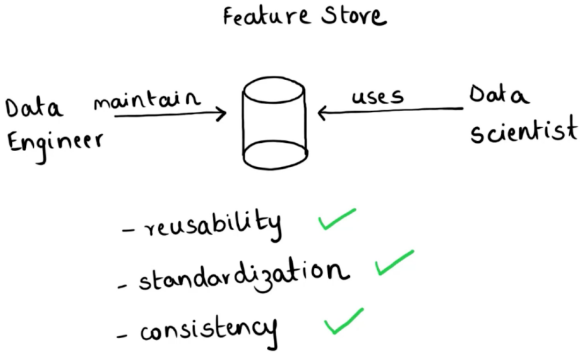


Notes:

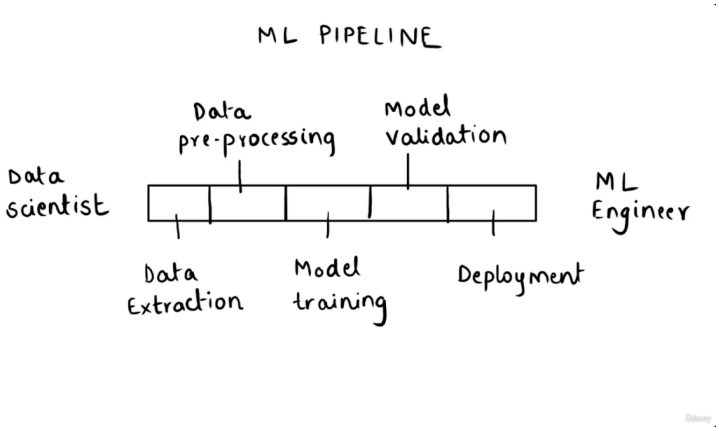
Lecture #23: Key Roles in An ML Team



Notes:

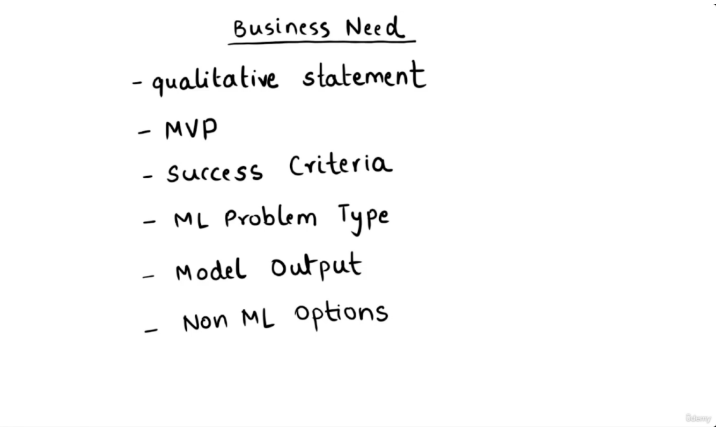


Notes:

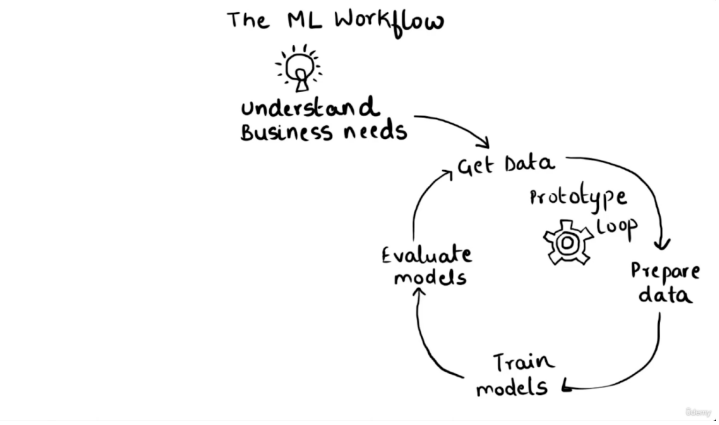


Notes:

Lecture #24: The ML Life Cycle

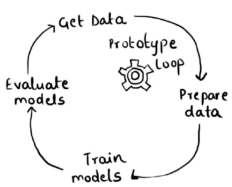


Notes:



Notes:

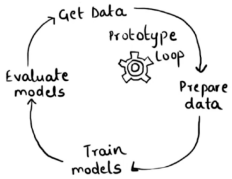
Get Data



- Structured & Unstructured
- ETL pipelines
- No Data? Need data acquistion strategy

Notes:

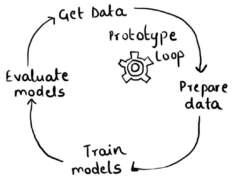
Prepare Data



- Cleaning Data
- Need labels?
- Explore feature
- Feature engineering

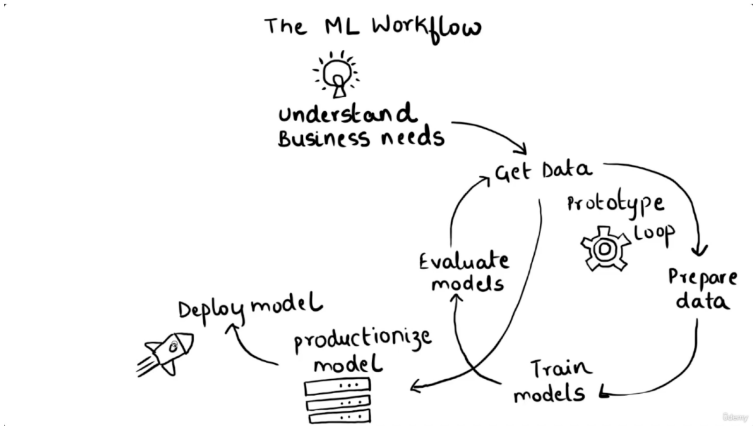
Notes:

Training Models



- Pick ML Models
- Feature Selection
- Split Dataset → Training model, Testing model
- Hyper parameter tuning
- Training strategy

Notes:



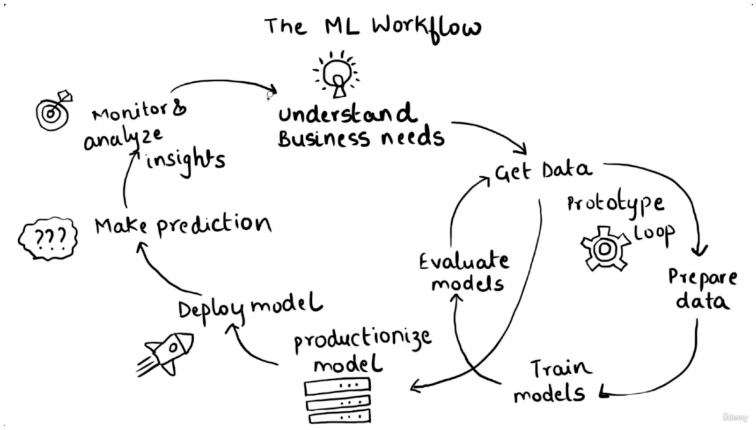
Notes:

Deploy Model

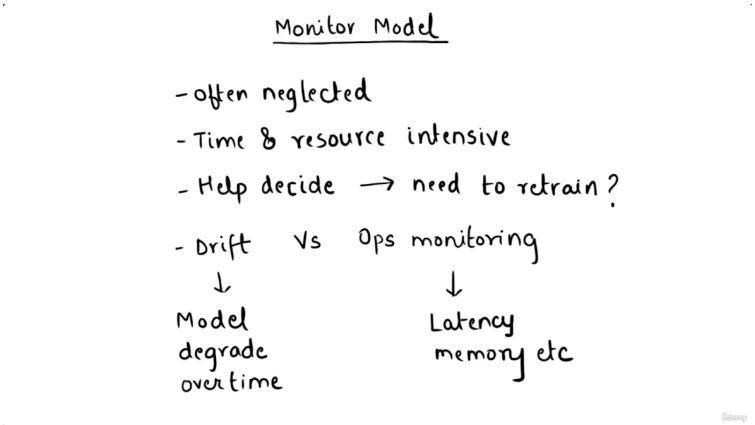
- Model available to apps
  eg. via an API

- Batch vs Realtime Consumption

Notes:



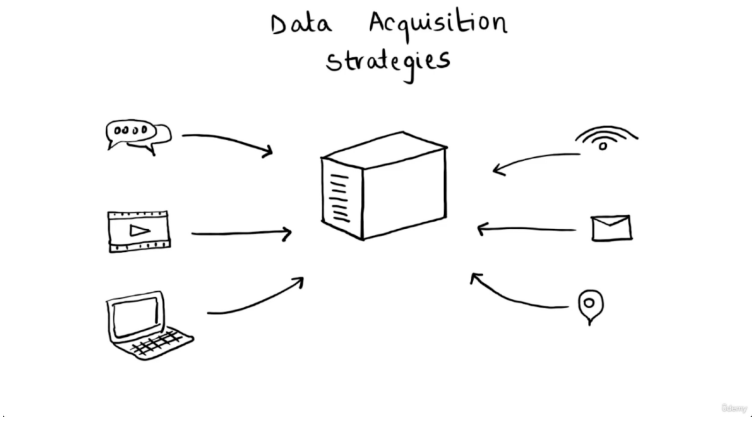
Notes:



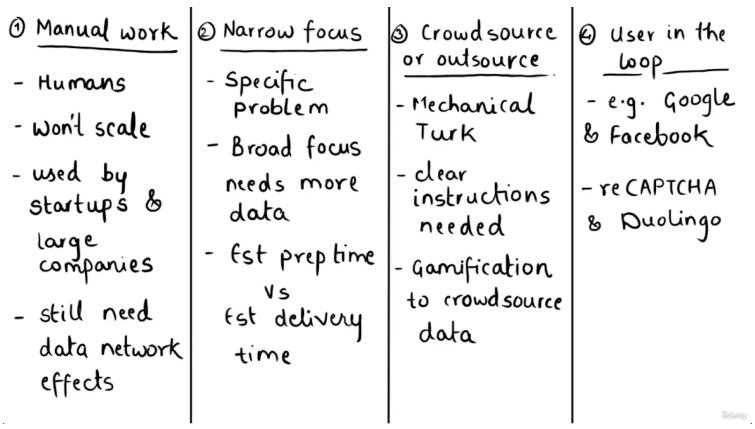
Notes:

Part 4: Get Your Data

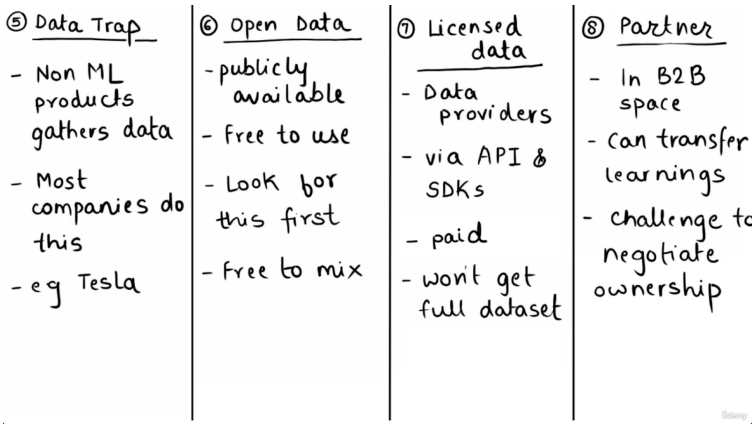
Lecture #28: Data Acquisition Strategies



Notes:

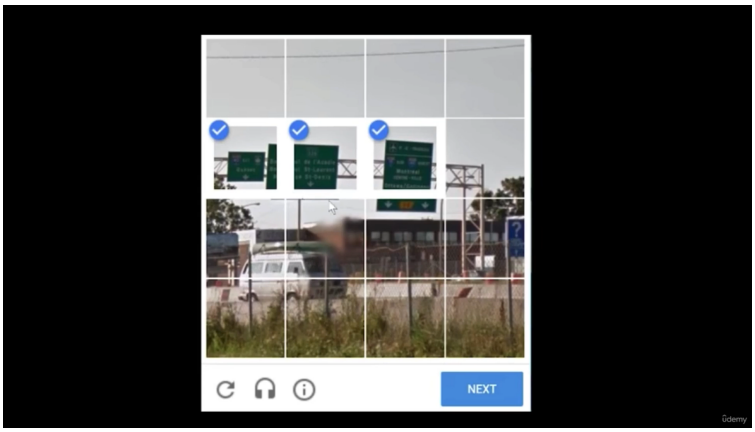


Notes:

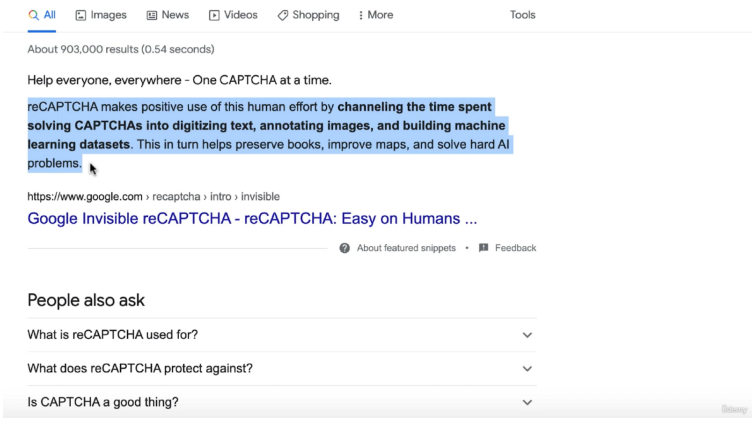


Notes:

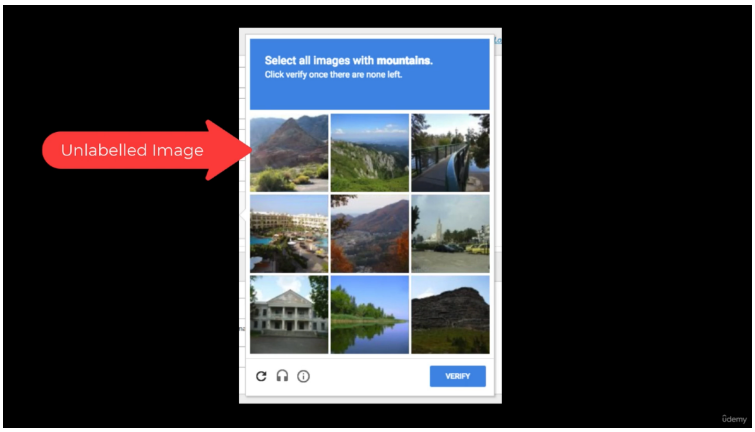
Lecture #29: Google reCAPTCHA



Notes:



Notes:



Notes:

Lecture #31: Start with a Simplified Problem

Start with a simplified problem

Binary classification ✓

Uni-dimensional regression ✓

Notes:

Problem Statement

Our problem is best framed as 3-class, single-label classification, which predicts whether a video will be in one of three classes—{very popular, somewhat popular, not popular}—28 days after being uploaded.

Notes:

Simplified Problem Statement ver 1

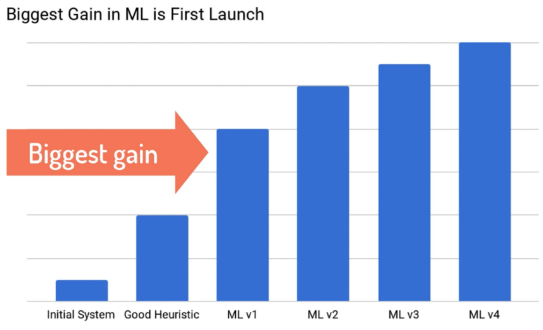
We will predict whether an uploaded video is likely to become popular or not (binary classification)

Notes:

Simplified Problem Statement ver 2

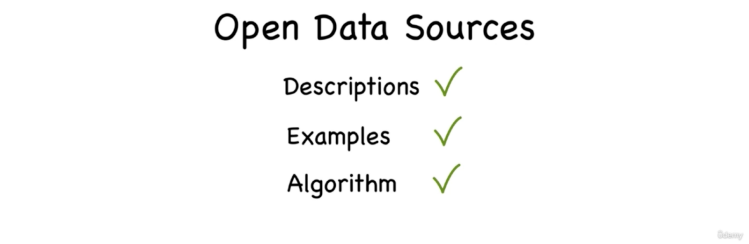
We will predict an uploaded video's popularity in terms of the number of views it will receive within a 28 day window (regression).

Notes:

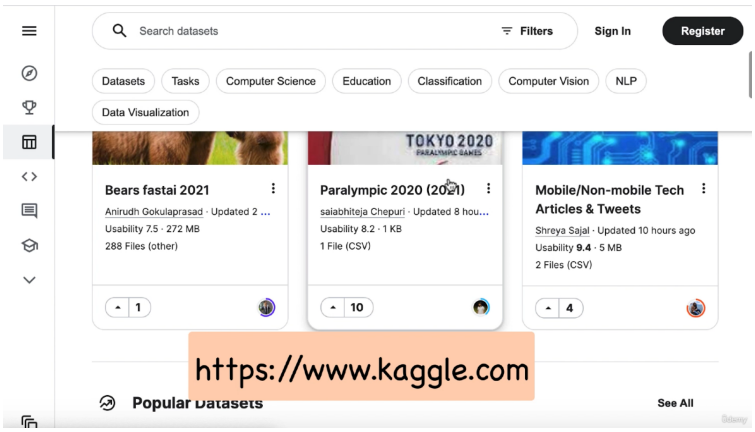


Notes:

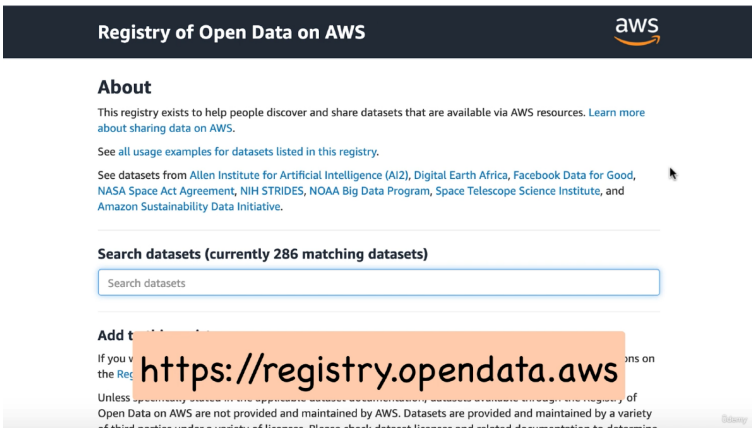
Lecture #33: Top Open Data Sources



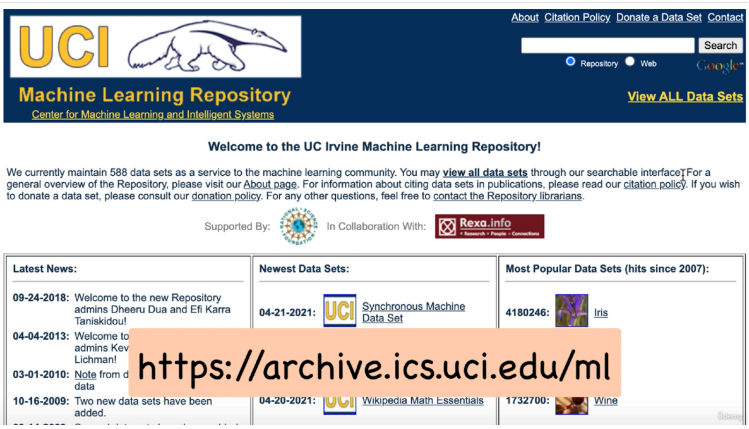
Notes:



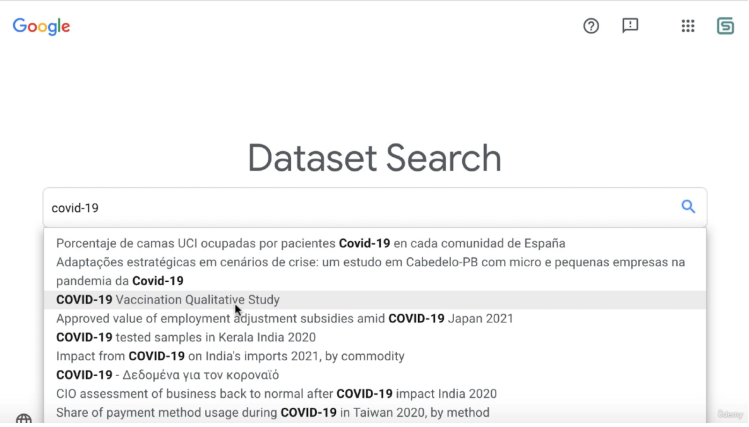
Notes:



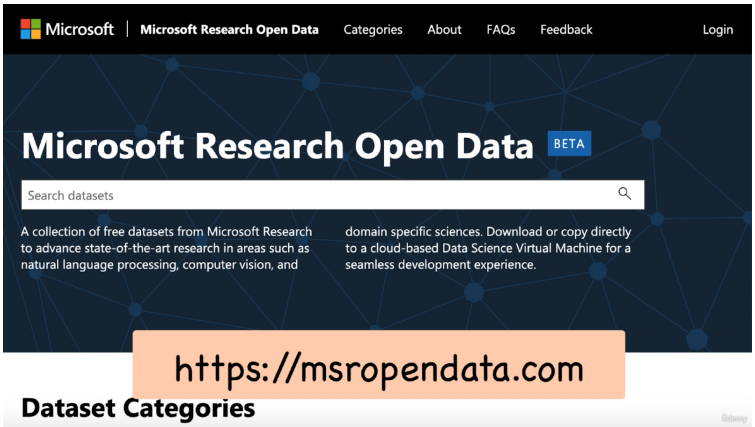
Notes:



Notes:

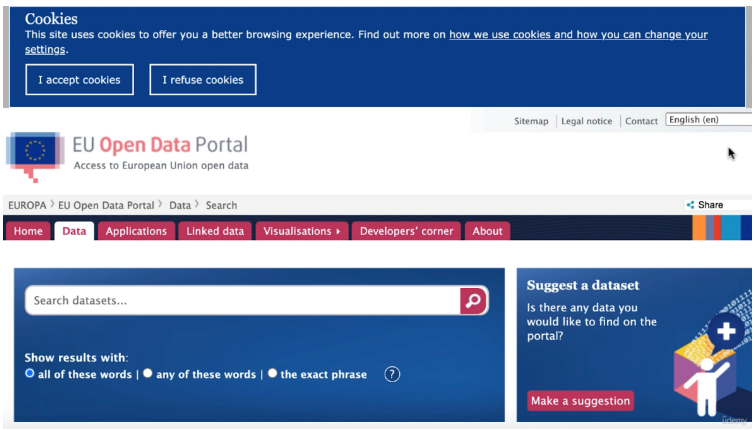


Notes:



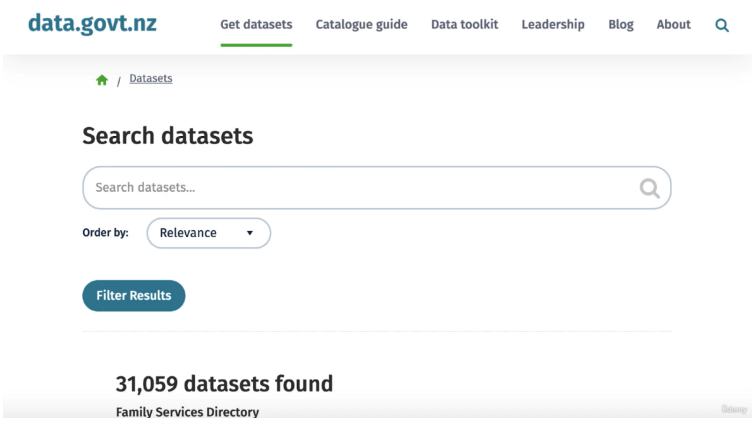
Notes:

Notes:

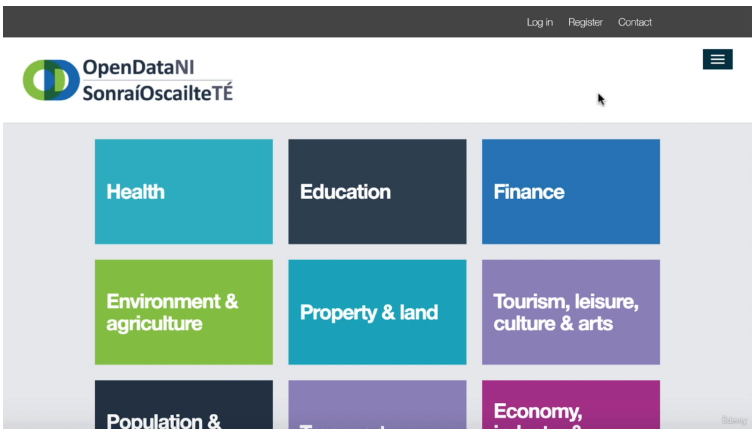


Notes:

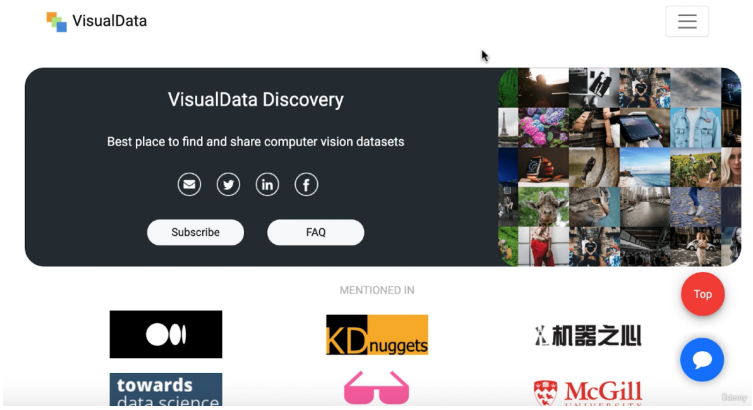
Notes:



Notes:

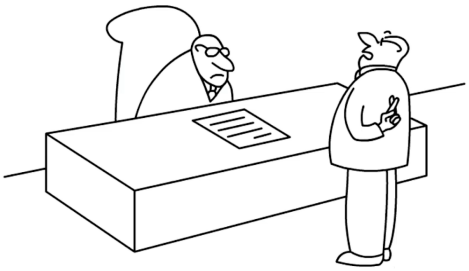


Notes:



Notes:

Lecture #34: How Much Data Do I Need?



"Yes sir, you can absolutely trust those numbers"

Notes:

How much data do I need? → It depends..

Rule of the thumb

data ≥ 10x (# of features)

Notes:

Feature Combinations

Degree	5 yrs Exp	Kids	Salary
Y	Y	N	$X
Y	Y	Y	$Y
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

Notes:

Simple model + big datasets

is better than

Fancy model + small dataset

Notes:

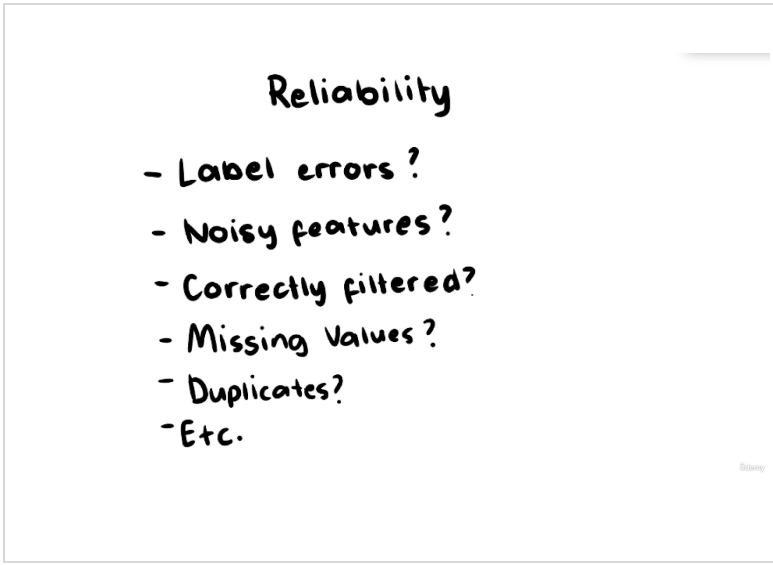
Data set	Size (number of examples)
Iris flower data set	150 (total set)
MovieLens (the 20M data set)	20,000,263 (total set)
Google Gmail SmartReply	238,000,000 (training set)
Google Books Ngram	468,000,000,000 (total set)
Google Translate	trillions

Notes:

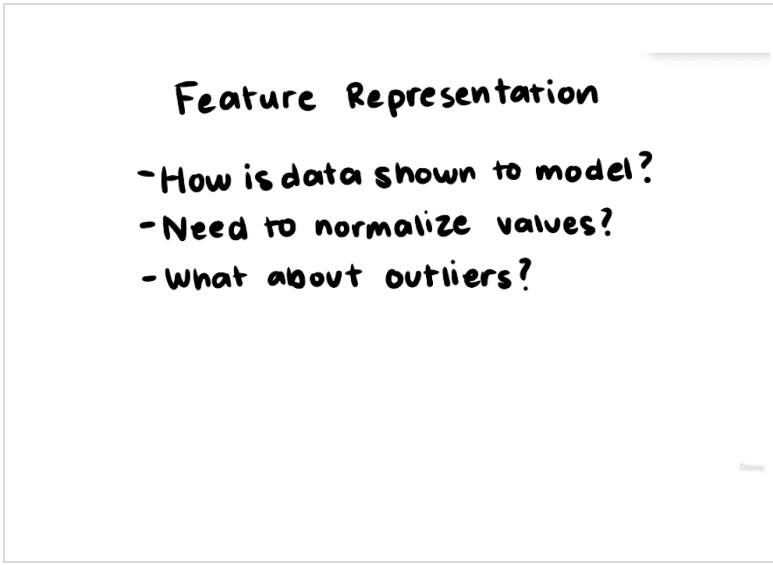
What is Quality?

Data is good if it accomplishes your goal.

Notes:

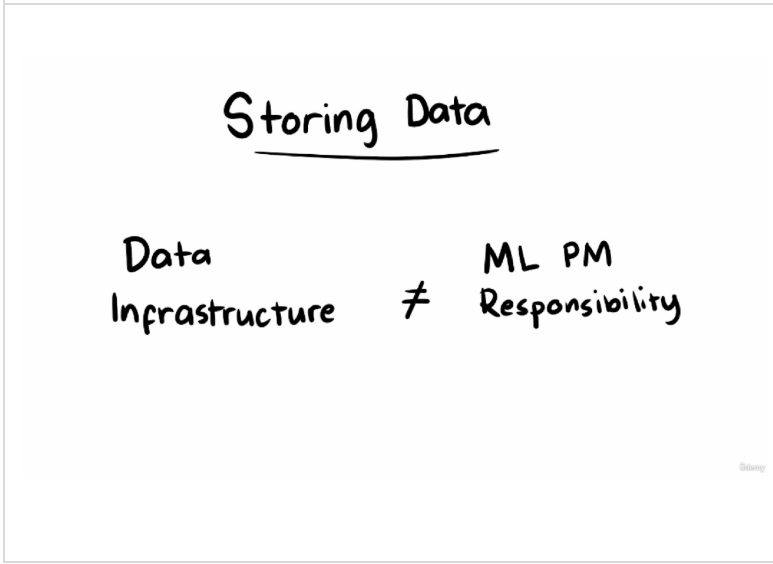


Notes:

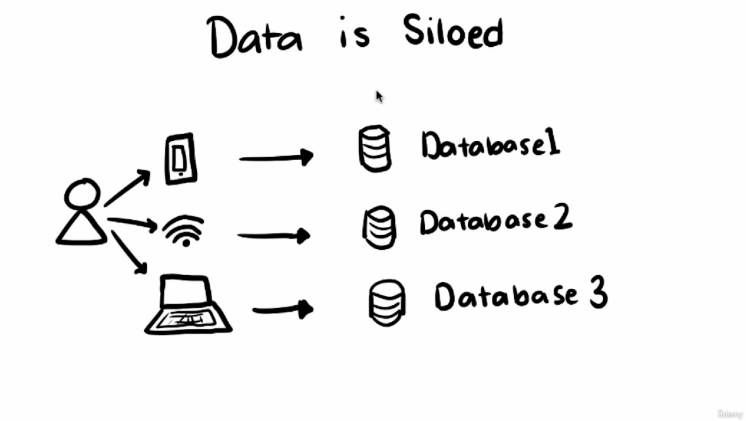


Notes:

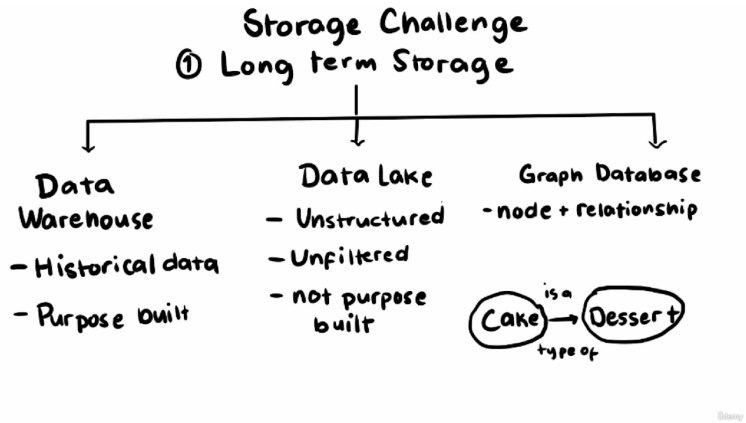
Lecture #35: Storing Data: Warehouses, Lakes & Graph Databases



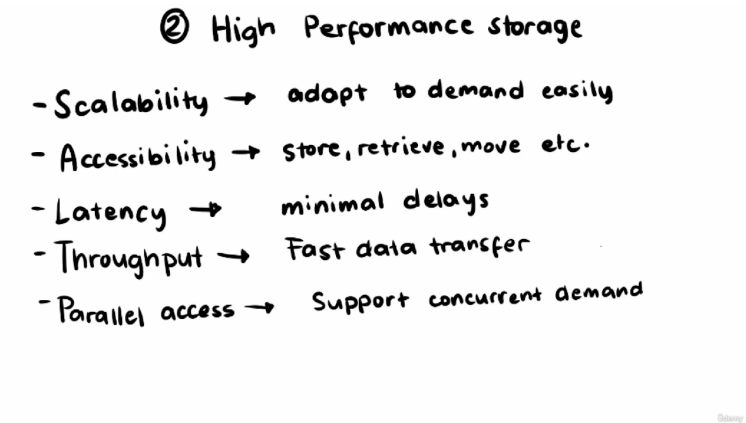
Notes:



Notes:



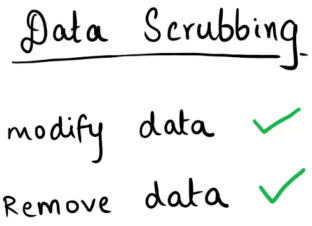
Notes:



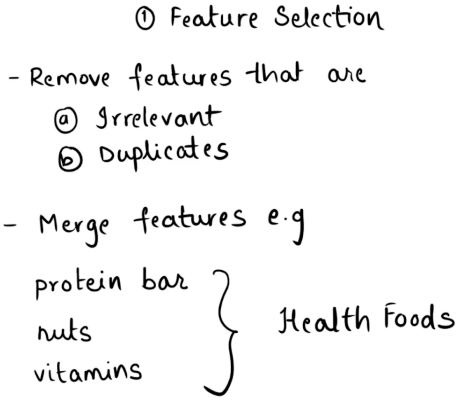
Notes:

Part 5: Prepare Your Data

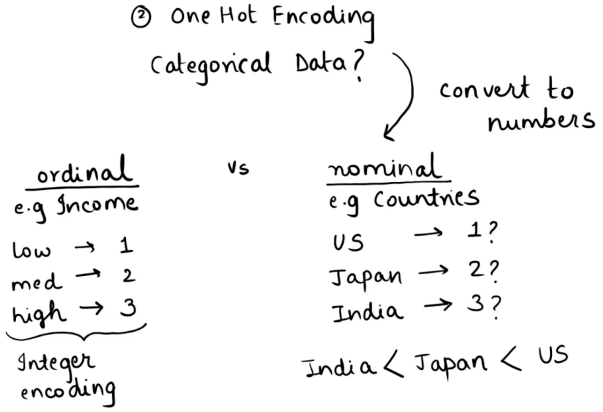
Lecture #36: Data Scrubbing



Notes:



Notes:



Notes:

Integer Encoding

Income	Category
Low	1
Med	2
High	3

Each category value is mapped to a number

One Hot Encoding

India	US	Japan
0	1	0
0	0	1
1	0	0

New features created for each category value

Notes:

③ Missing Data - Approaches

ⓐ Replace with mode value i.e most comman value

For : categorical ✓
binary variables ✓

ⓑ Replace with median value i.e middle value

For : Integers ✓
Continuous variables ✓

ⓒ Remove

For : Last resort ✓

Notes:

Lecture #37: Sampling and Splitting Data

Sampling
Google search?

- Random queries?
- Random sessions?
- Random users?

It depends...

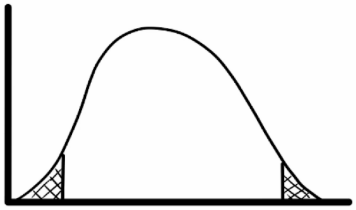
What problem are you solving?
e.g. previous query? → Session level
user behavior last week? → user level

Notes:

Other Filters

① Personally Identifiable Info
e.g. E-mail, phone #

② Infrequent features



Notes:

Imbalanced Data...

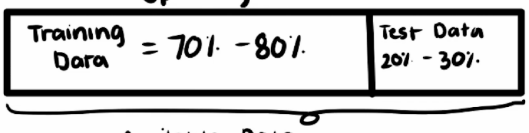
Is a classification problem

Model only learns from this

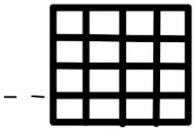
Fix using → Downsampling ✓
Upweighting ✓

Notes:

Splitting Data



Available Data

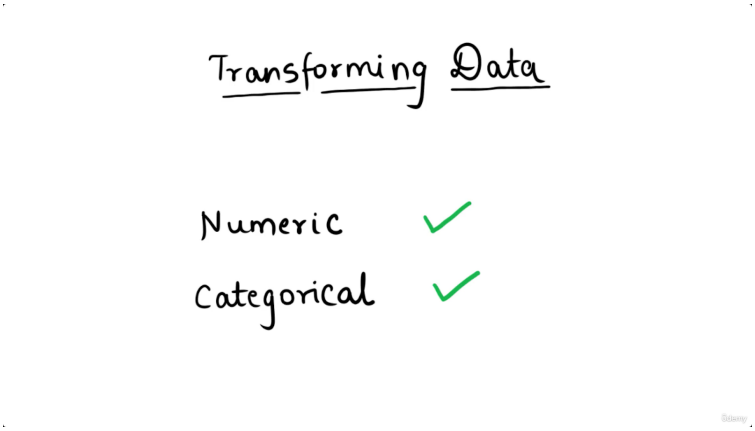


→ Split rows after randomizing data

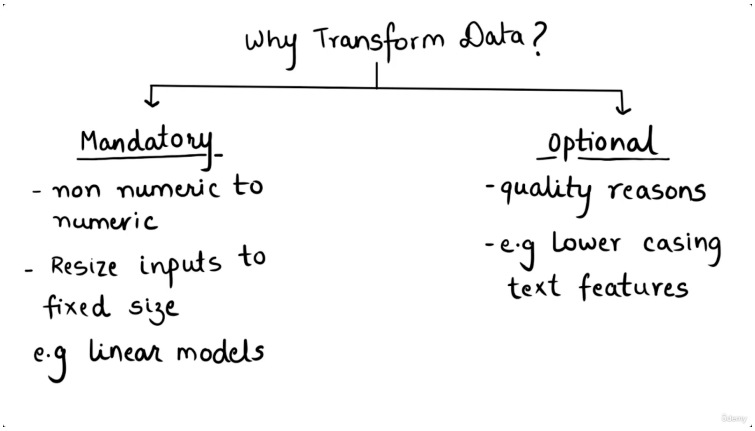
Deterministic Randomization → means can reproduce datasets

Notes:

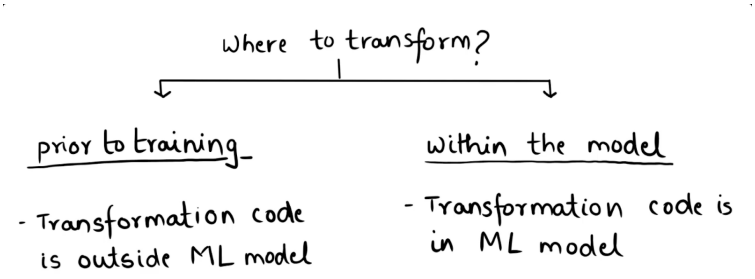
Lecture #38: Transforming Data



Notes:



Notes:

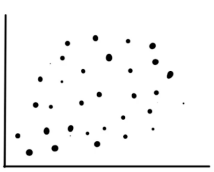


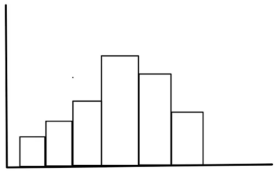
Notes:

Remember to...

Explore and clean data before transforming it

- Basic Statistics
- Missing data
- Visualize data often





Notes:

Numeric Data

Normalization

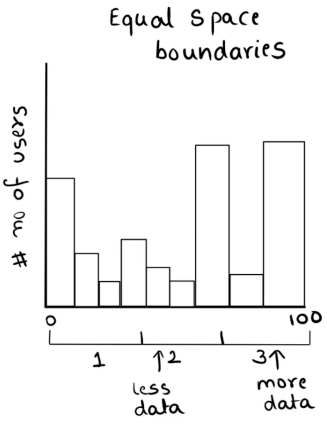
- make same scale
- Different ranges

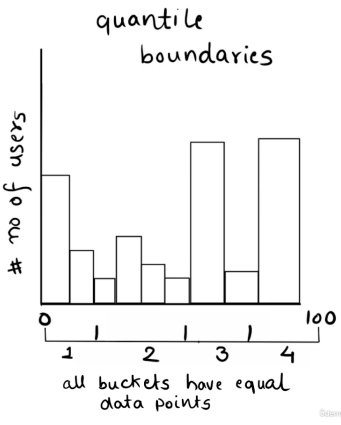
e.g age & income

Bucketing

- numeric to categorical
- data allocated to buckets

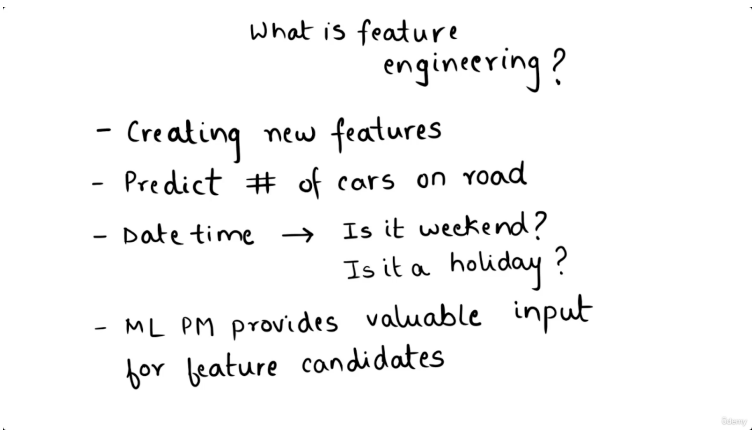
Notes:



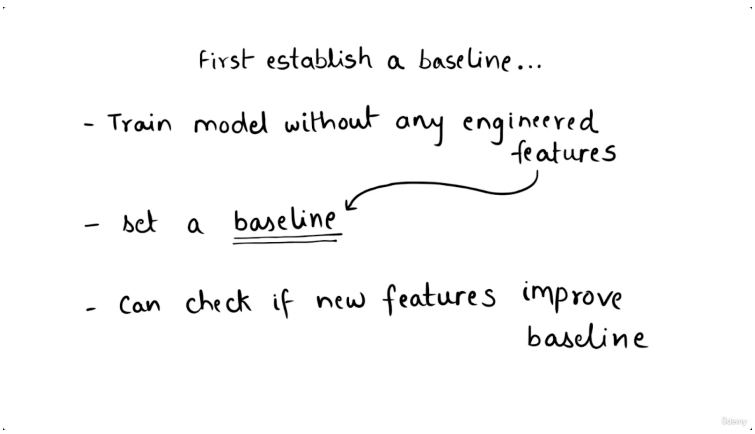


Notes:

Lecture #39: Feature Engineering



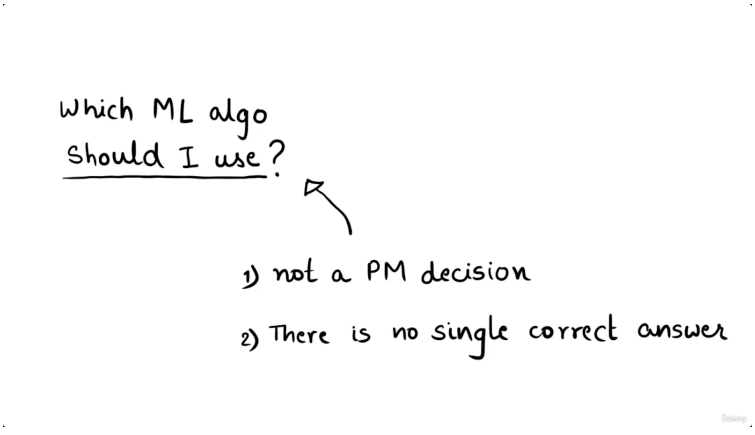
Notes:



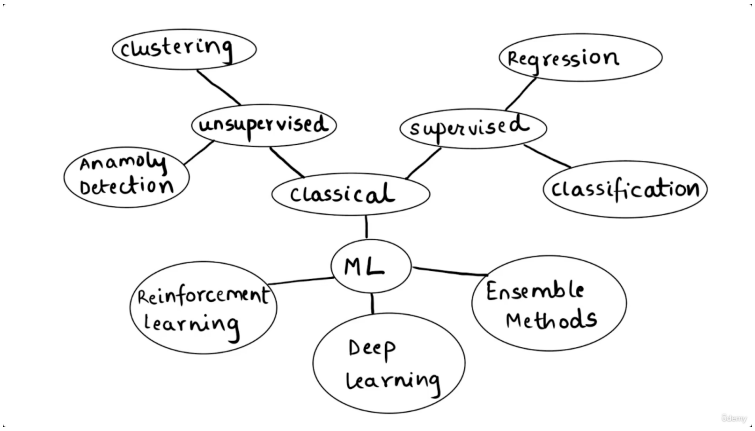
Notes:

Part 6: Build Your Model

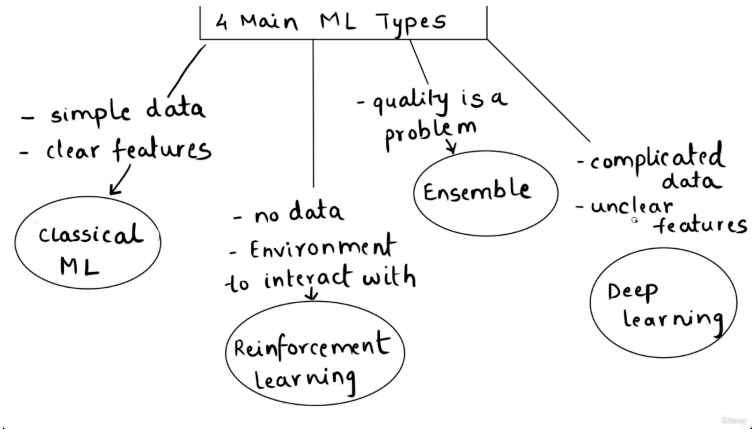
Lecture #41: Which ML algorithm should I use?



Notes:

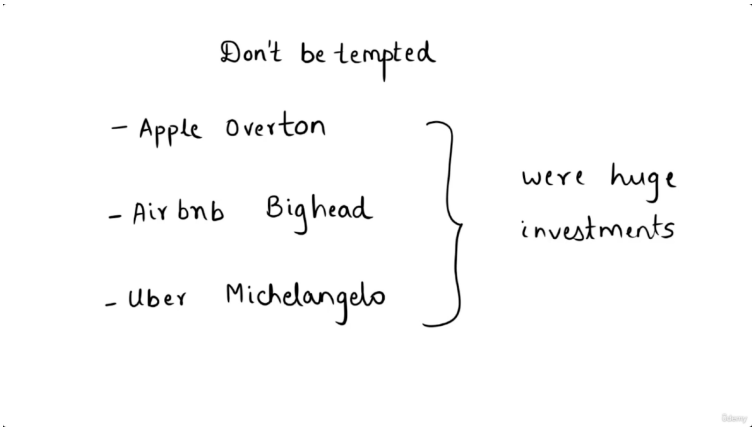


Notes:

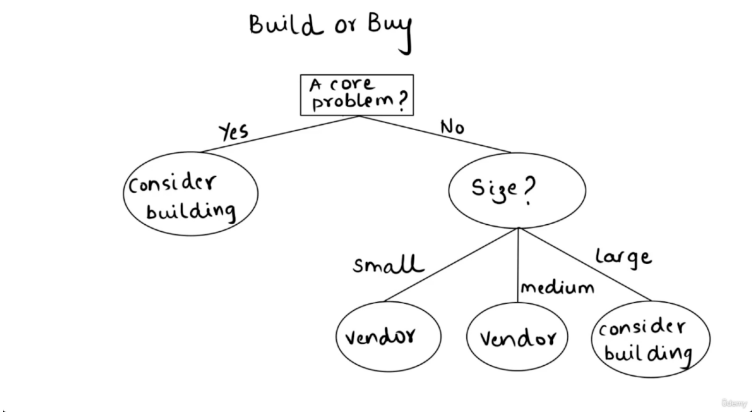


Notes:

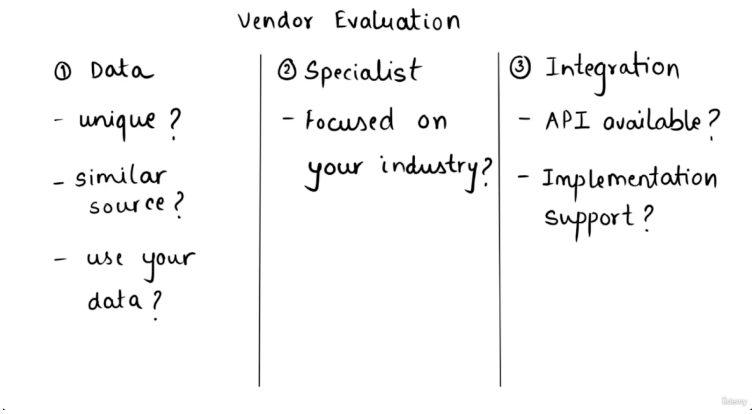
Lecture #42: Build, Outsource or Buy Your ML Solution?



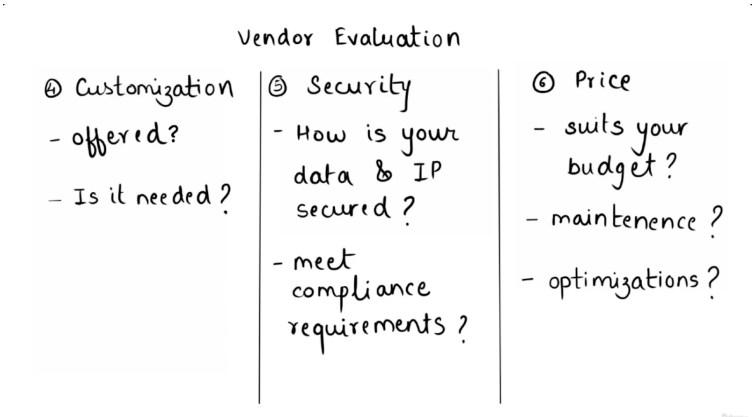
Notes:



Notes:



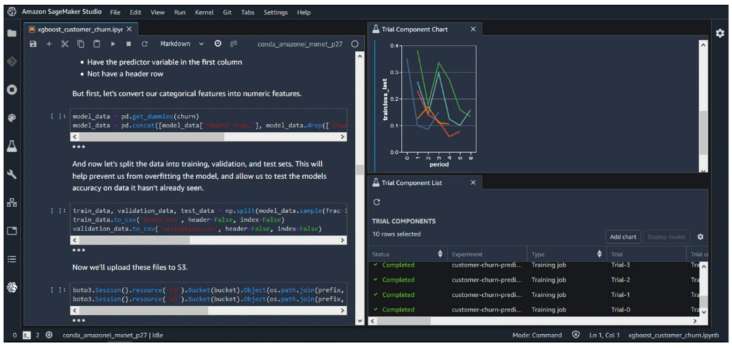
Notes:



Notes:

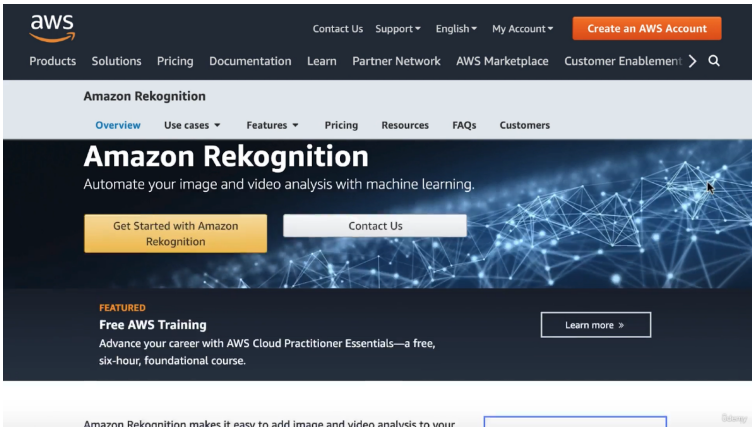
Lecture #43: Machine Learning as a Service (MLaaS)

Notes:



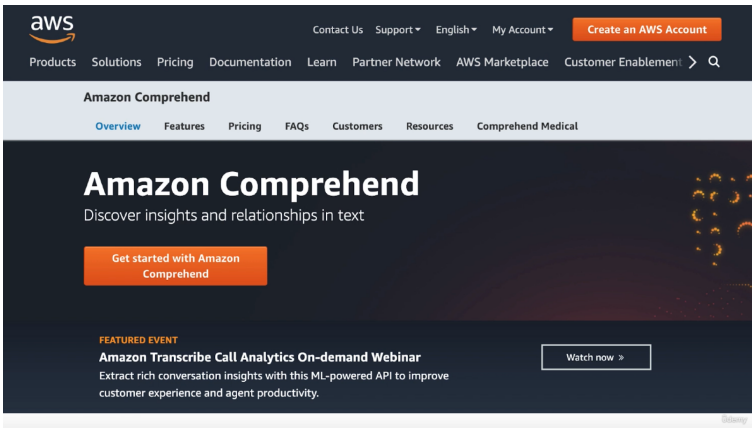
Notes:

Notes:



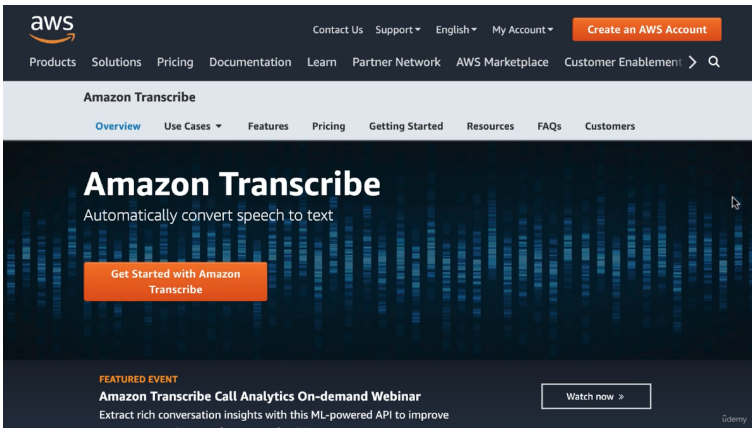
Notes:

Notes:



Notes:

Notes:



Notes:

Notes:

Vertex AI

Notes:

Vertex AI

Auto ML

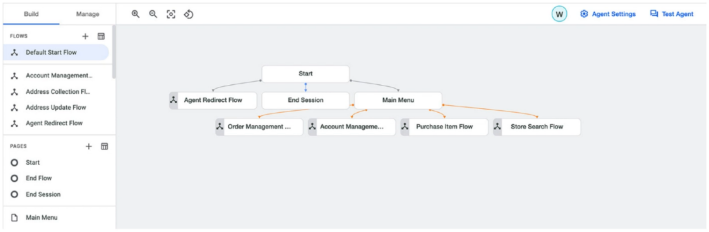
Notes:

Vertex AI

AI Infrastructure

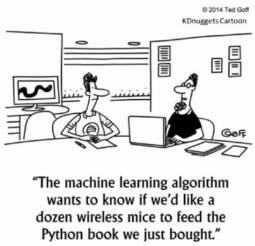
Auto ML

Notes:



Notes:

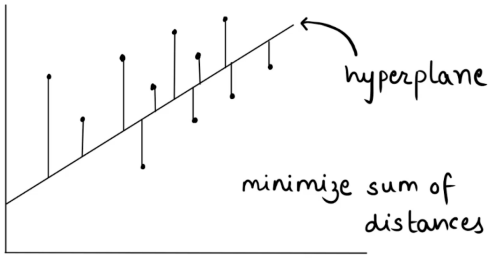
Next: ML Algorithms



Notes:

Lecture #44: Regression (Linear, Polynomial, Logistic)

Linear Regression



Notes:

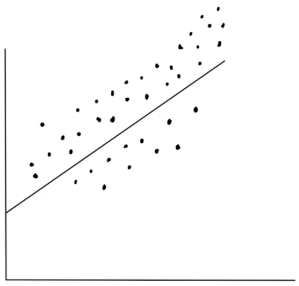
1 Feature = Simple Linear Regression

$y = b_0 + b_1 * x_1$

+1 Feature = Multiple Linear Regression

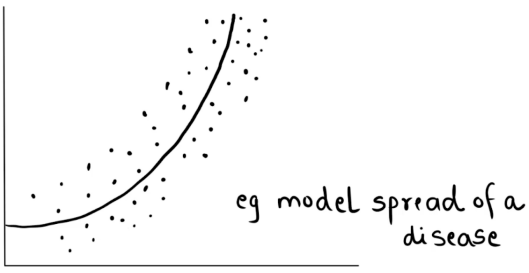
$y = b_0 + b_1 * x_1 + b_2 * x_2 \ldots\ldots$

Notes:



Notes:

Polynomial Regression



$y = b_0 + b_1 x_1 + b_2 x_1^2 \ldots\ldots + b_n x_n^2$

Notes:

Logistic Regression

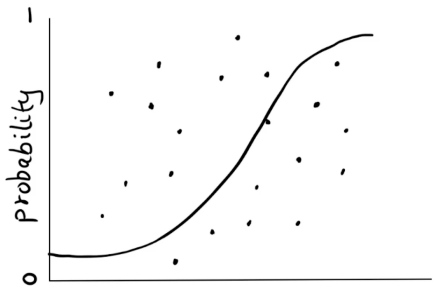
↑

used for classification

1) Binary e.g yes or no

2) Multi class e.g 0 - 9

Notes:



Notes:

spam Vs not spam

- set your threshold

e.g if probability is > 70% → spam

- Threshold decided by your risk tolerence

e.g cancer diagnosis → low tolerance

fraud loan application → higher tolerance

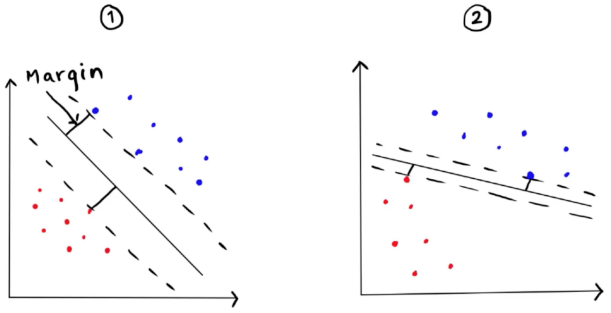
Notes:

Lecture #45: Classification (SVM, K-NN, Decision Trees)

Support Vector Machines

- Geometrically motivated
- Examples : cat or dog?
  positive or negative review?
  red or blue dots?

Notes:



Lagrangian Optimization

Notes:

Parametric models

- linear regression
- logistic regression
- SVM

} predefined parameters

Non Parametric models

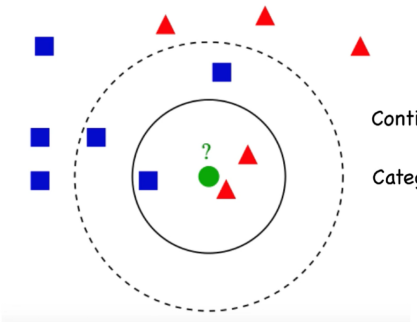
- K-nearest neighbor
- Decision Tree

} No Predefined Parameter

Notes:

K-Nearest Neighbor

Notes:



Continuous —> Take average

Categorical —> Take mode

Notes:

K-NN Examples

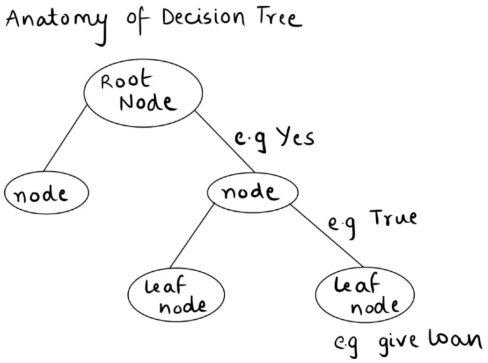
- Fraud detection (classification)
- Housing prices (regression)

useful when physical proximity matters

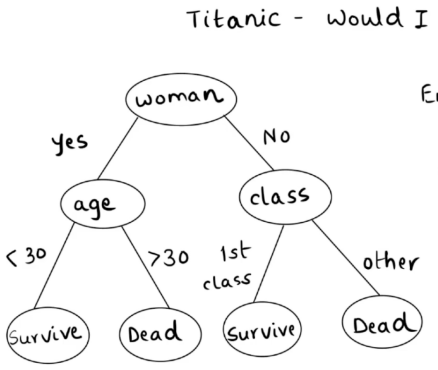
Notes:

Decision Trees

Notes:



Notes:



Notes:

Decision Trees

Pros

- Easy to understand
- Need less data
- Can be used with other algos

Cons

- unstable
- Relatively inaccurate

Notes:

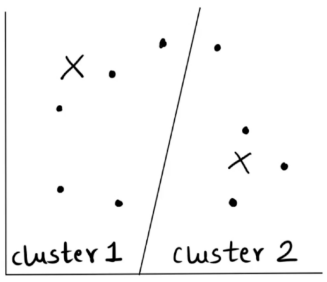
Lecture #46: Clustering (K-Means, Means Shift)

K-means Clustering

- Cluster into "K" groups
- Larger "K" means smaller groups
- Data points assigned to groups
- centroid for each group

Notes:

1. Select a centroid
2. Check distance
3. Check mean value of data point
4. Move centroid to new mean value
5. Repeat till convergence



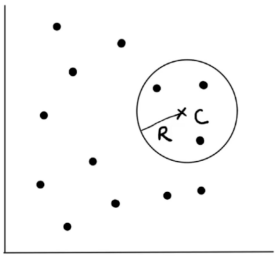
Notes:

Means shift clustering

- sliding window
- centroid based
- Goal is to find most dense areas

Notes:

① Select "C" & "R" value
② Check mean value inside circle
③ Move centroid to new mean value
④ Repeat this for all data points till convergence



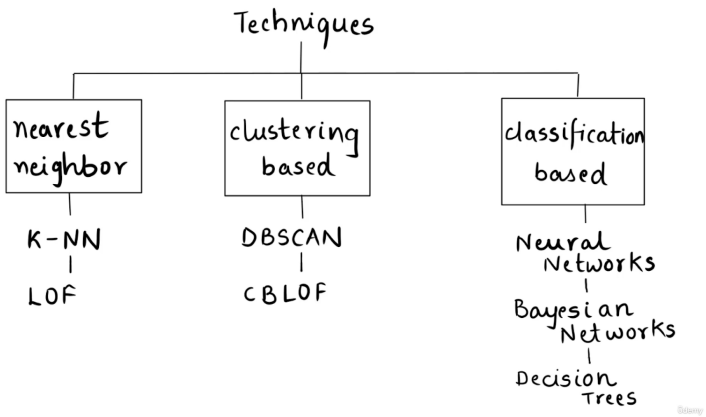
Notes:

Lecture #47: Anomaly Detection (Local Outlier Factor, DBSCAN)

Anomaly Detection

Intrusion ✓
Fraud ✓
Fault ✓
ETC

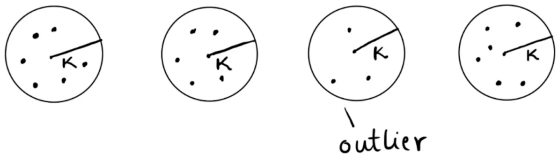
Notes:



Notes:

Local Outlier Factor

- looks at local density



- LOF score $< 1 \rightarrow$ not outlier
- LOF score $> 1 \rightarrow$ outlier

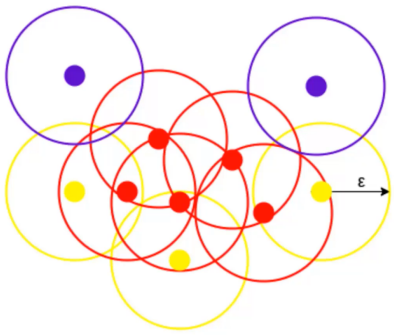
Notes:

DBSCAN

1) Epsilon - Radius of circle around each data point

2) minPoints - min # of points needed for a core point

Notes:



Notes:

DBSCAN

Pros

- Simple
- only 2 parameters
- scales well

Cons

- only uses distance
- sensitive to Epsilon & minpoints values
- can't handle large differences in densities

Notes:

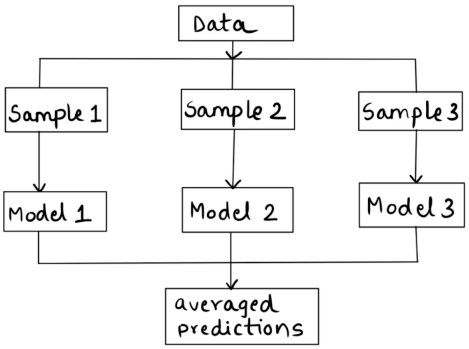
Lecture #48: Ensemble Methods (Bagging, Boosting, Stacking)

Ensemble Methods

- combine model predictions for better results
- sequential – Base learners used in sequence eg AdaBoost
- parallel – Base learners used in parallel

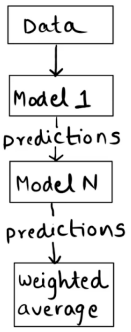
Notes:

① Bagging



Notes:

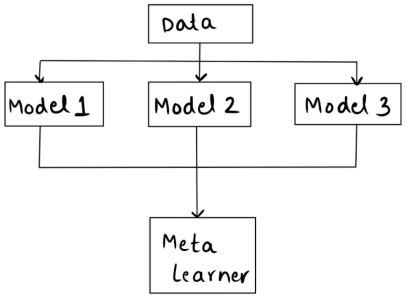
② Boosting



e.g
- GBM
- XGBM
- Light GBM
- Cat Boost

Notes:

③ Stacking



Notes:

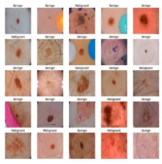
Part 7: Deploy Your Model

Lecture #49: The Confusion Matrix

Accuracy

Fraction of predictions model gets right

Accuracy = # of correct predictions / # of total predictions



Dataset includes 100 examples

Accuracy = 0.91 —> 91 correct predictions

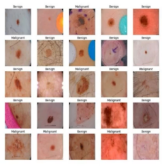
Is our model performing well?

Notes:

Accuracy

Fraction of predictions model gets right

Accuracy = # of correct predictions / # of total predictions



Dataset includes 100 examples

91 are benign —> 91 correctly identified

9 are malignant —> 0 correctly identified

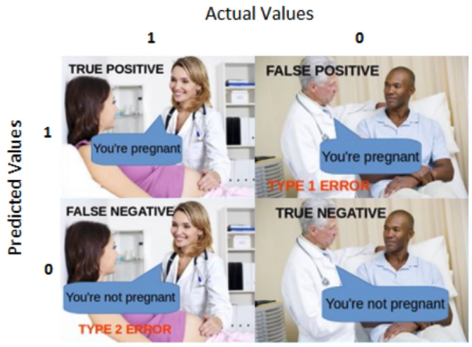
Accuracy = 91/100

Notes:

Predicted Class

Actual Class		Positive	Negative
	Positive	True Positive (TP)	False Negative (FN) **Type II Error**
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)

Notes:



Notes:

True Positive

Actual Predicted

Notes:

Lecture #50: Evaluation Metrics (Precision, Recall & F1 Score)

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Notes:

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

1) Accuracy = (TP+TN)/Total
(100+50)/165 = 0.91

2) Error Rate = (FP+FN)/Total
(10+5)/165 = 0.09

3) True Positive Rate = TP/Actual Yes
100/105 = 0.95 (Recall)

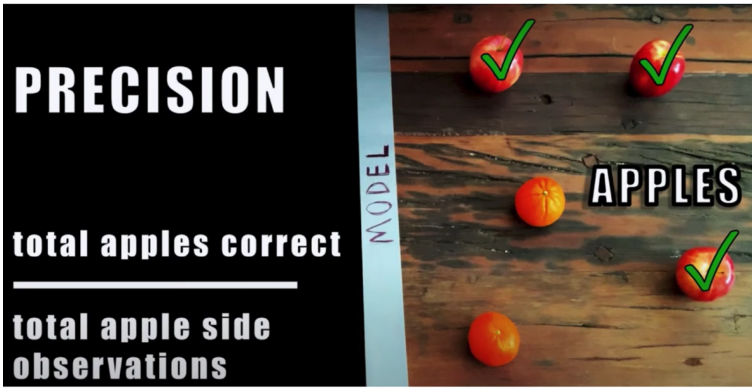
4) False Positive Rate = FP/Actual No
10/60 = 0.17

5) True Negative Rate = TN/Actual No
50/60 = 0.83

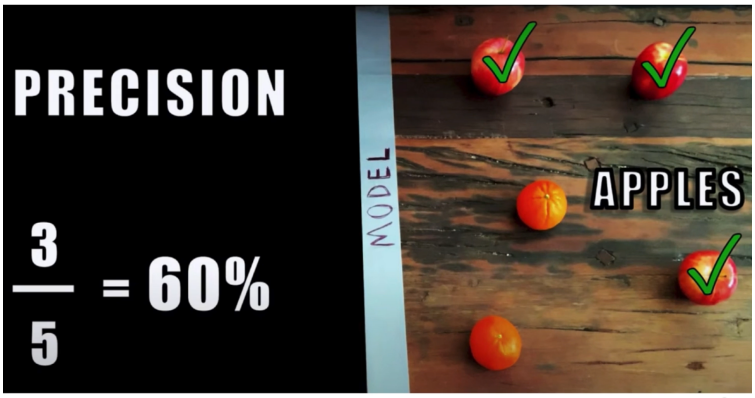
6) Precision = TP/Predicted Yes
100/110 = 0.91

7) Prevalence = Actual Yes/Total
100/165 = 0.64

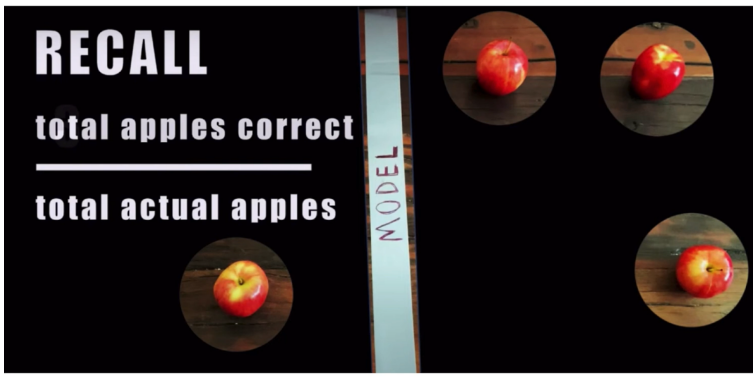
Notes:



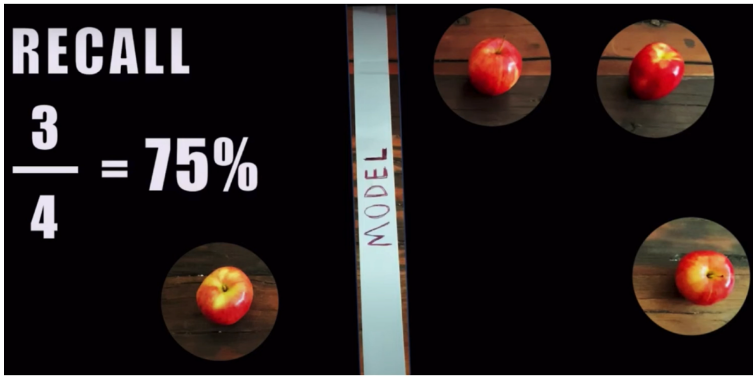
Notes:



Notes:

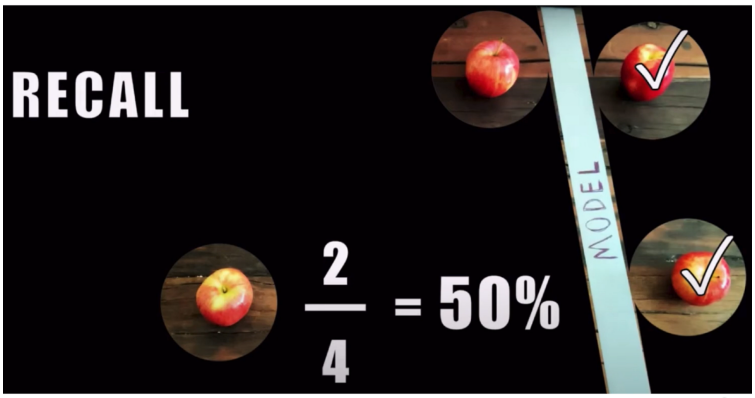


Notes:

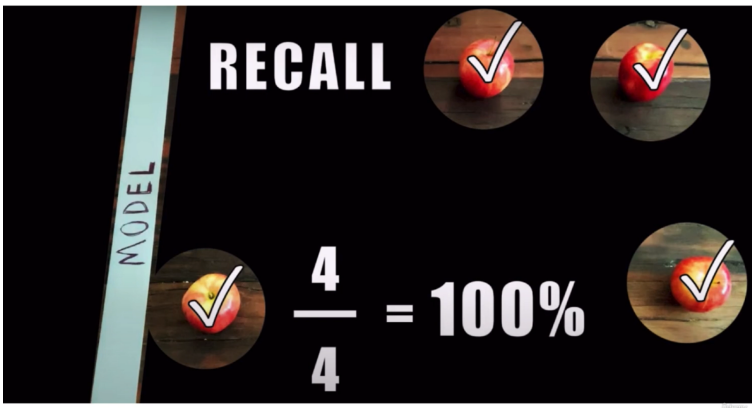


Notes:

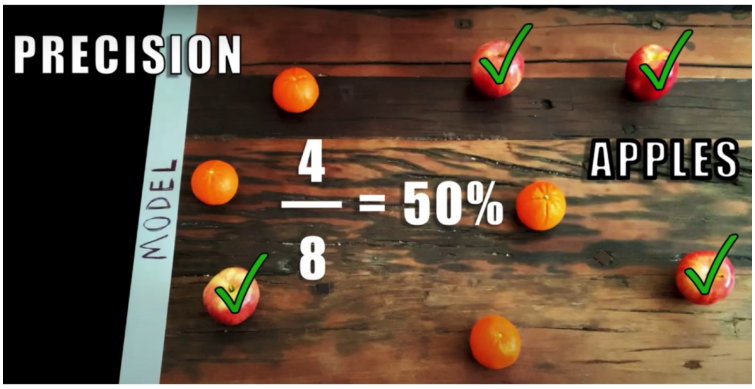
Notes:



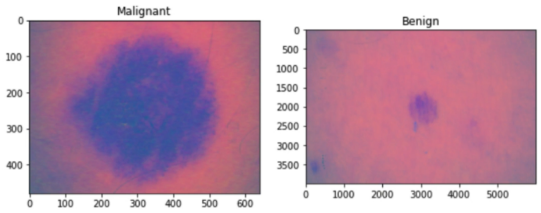
Notes:



Notes:

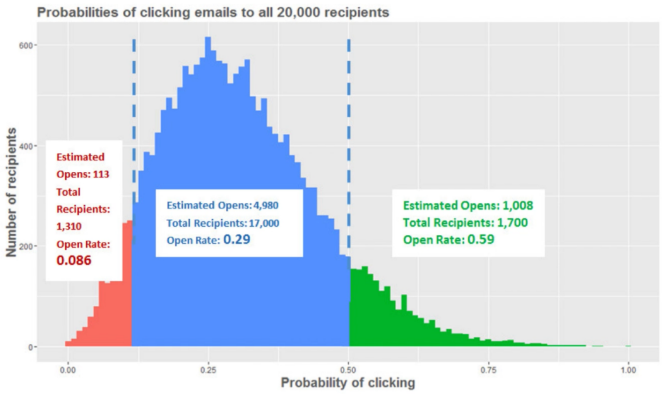


Notes:



Optimize for high recall rate

Notes:



Notes:

F-1 Score (0 —> 1)

Combines precision and recall

Harmonic mean of precision & recall

$$= 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Notes:

Lecture #52: User Experience Optimization

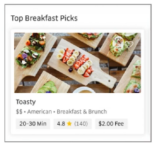
	Precision	Recall	F1 Score
Model 1	82%	90%	86%
Model 2	75%	80%	77%
Model 3	87%	70%	78%

Uber Eats

Notes:

	Predicted: No	Predicted: Yes
Actual: No	TN	FP
Actual: Yes	FN	TP

Uber Eats



Notes:

	Predicted: No	Predicted: Yes
Actual: No	TN	FP
Actual: Yes	FN	TP

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

Notes:

	Predicted: No	Predicted: Yes
Actual: No	TN	FP
Actual: Yes	FN	TP

Notes:

	Predicted: No	Predicted: Yes
Actual: No	TN	FP
Actual: Yes	FN	TP

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

Notes:

	Precision	Recall	F1 Score
Model 1	82%	90%	86%
Model 2	75%	80%	77%
Model 3	87%	70%	78%

Uber Eats

Notes:

Lecture #54: Deployment Methods

Notes:

Model Deployment

Build ---> Evaluate ---> Deploy

✓ ✓ ? ?

Notes:

Realtime

- Hosted on server as an endpoint
- via API's
- Fast predictions on-demand

Batch

- Triggered by an event
- ML model deployed to make prediction for that batch

Edge

- Data not passed to backend
- prediction is made on edge device eg IOT

Notes:

Latency

- prediction based on user input

↓

Real time may be preferred

Privacy

- Hosted by 3rd party?
- PII info involved?

↓

Deploy on device on edge

connectivity

- limited or no connectivity?

↓

Deploy on device on edge

Cost

- Real time predictions?

↓

24/7 availability & dedicated team