

Working with data in industrial machine learning applications



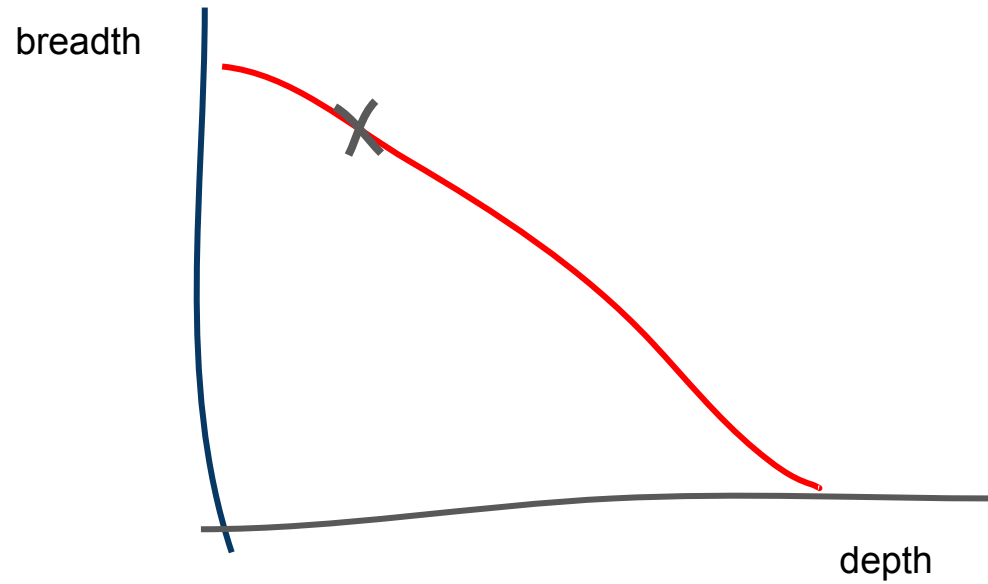
Andreas Damianou
Spotify

Course for Univ. Sapienza, 27 - 28 April 2022

Outline

- Data and model considerations, interplay between data and model
- Types of bias in data and related issues & solutions
- Uncertainty in data/models
- Data tools for industrial scale
- From raw data to ML features
- Time evolution considerations for data in ML (distribution shift, versioning...)
- Privacy and regulation
- Extra: Graph data representations for the industry

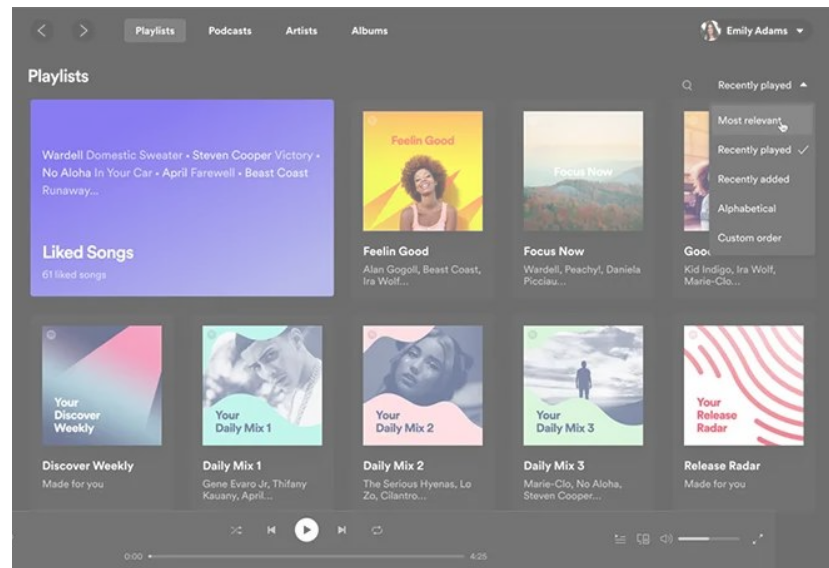
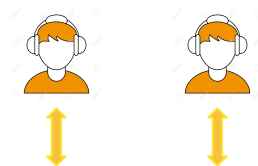
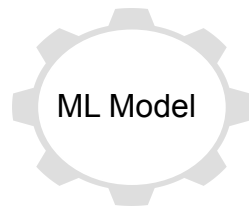
This talk



Data and model considerations

... and the interplay between data and model

ML-powered experience



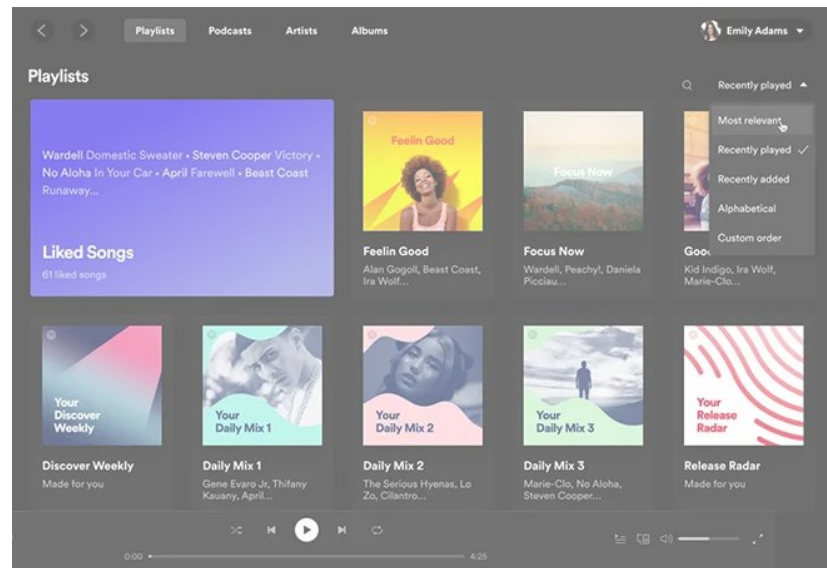
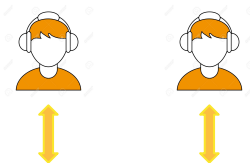
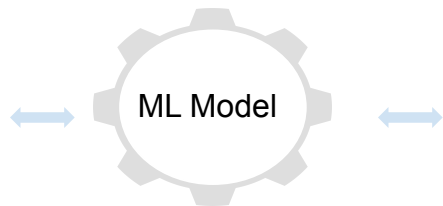
ML-powered experience. Data-powered ML

Show	Topic	Publisher
spotify:show:bl	Science	Spotify
spotify:show:ab	Education	Spotify
spotify:show:sA	Politics	Spotify

Episode	Entity
spotify:episode:qd	wiki.org/David_A
spotify:episode:9s	wiki.org/Elton_Jo
spotify:episode:gA	wiki.org/David_A

User	Episode
fjsa-xkgw-afFs-	spotify:episode:gA
gkad-kd98-ajgs	spotify:episode:9s
gkad-kd98-ajgs	spotify:episode:qd

User	SearchQuery
fjsa-xkgw-afFs-	gjsGjdAdghsgUU
gkad-kd98-ajgs	gjAsgSkshx0sa
gkad-kd98-ajgs	gDaobifjq9bmsa



ML in a nutshell

Data



+

Model



compute



Inference

9
42
12

Various roles working on an industrial ML project

DATA ENGINEER

ML ENGINEER

ML RESEARCHER/
SCIENTIST

DATA
SCIENTIST

PEOPLE'S MANAGER

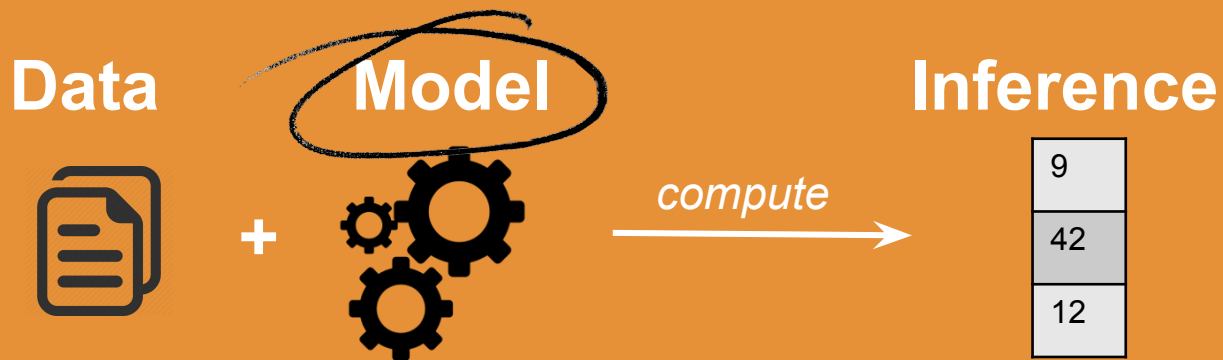
BACKEND ENGINEER

ENGINEER MANAGER

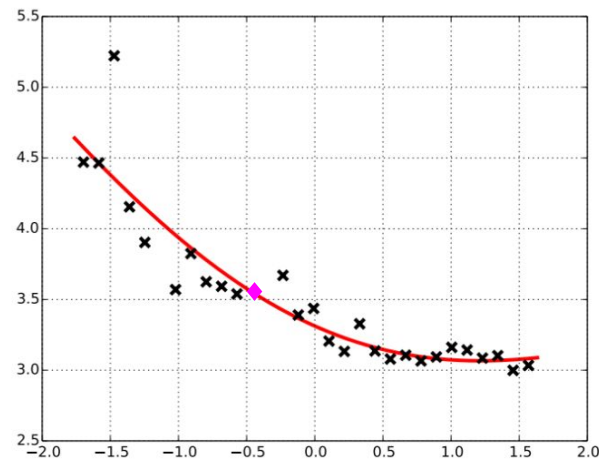
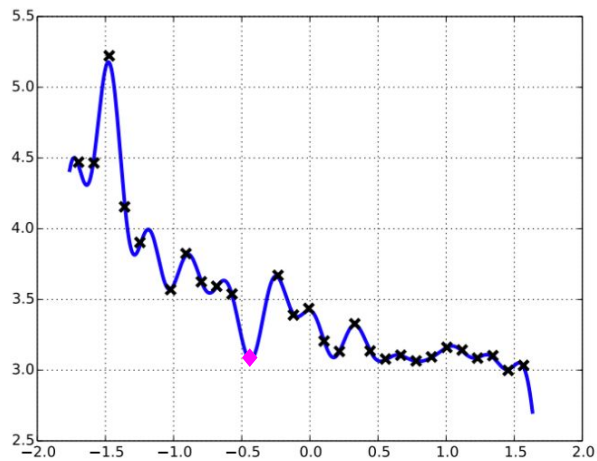
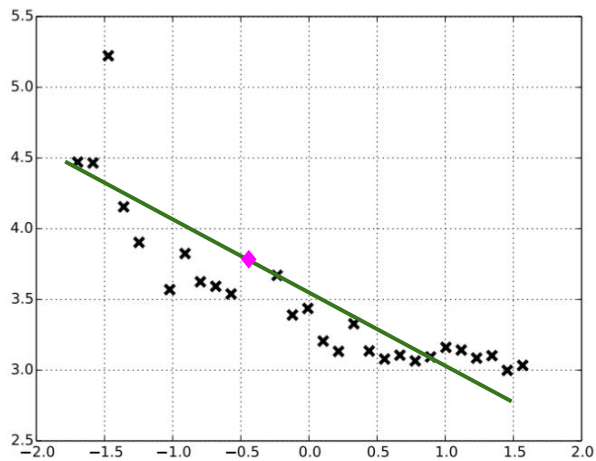
PRODUCT MANAGER



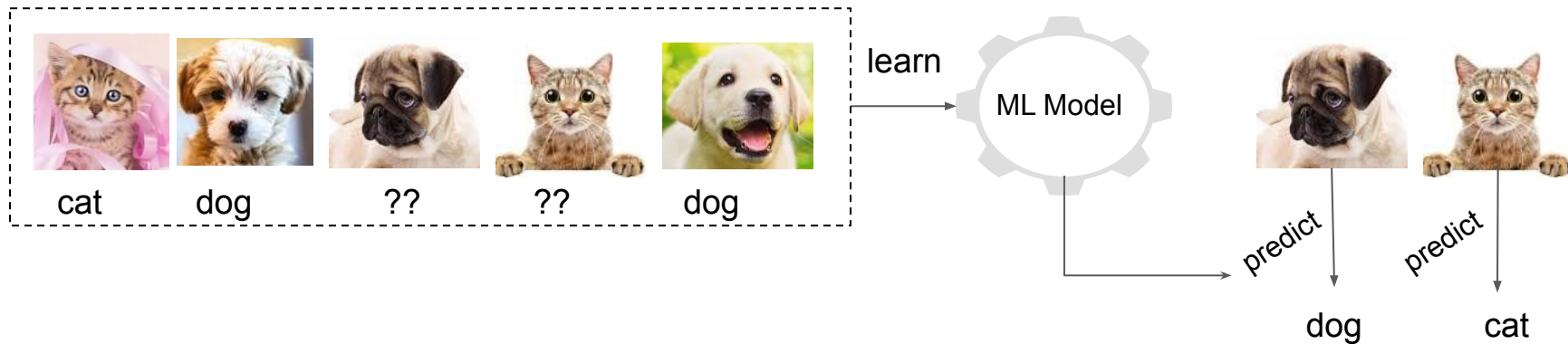
ML in a nutshell



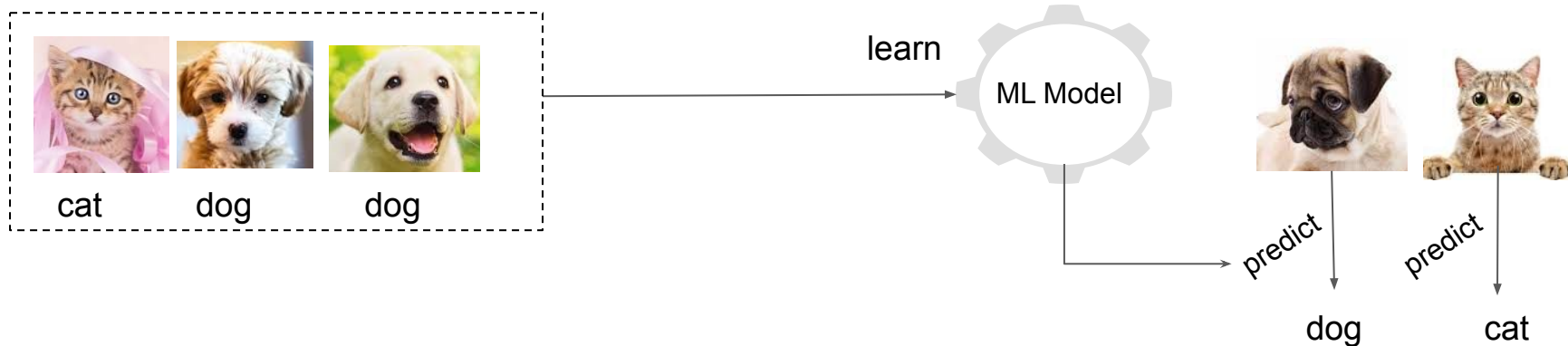
Model assumptions



Transductive learning



Inductive learning



Open-world learning

Training data



Test data



Open-world learning

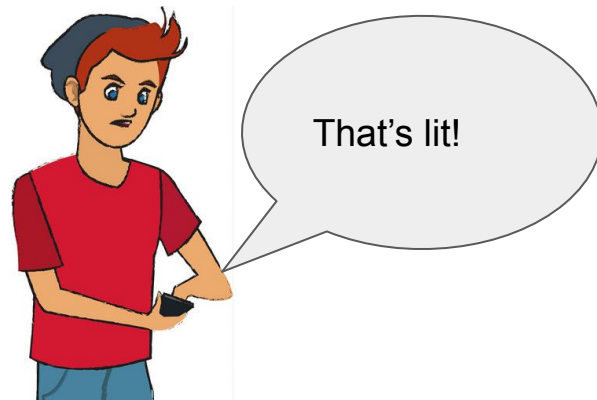
Training data

- *"Hello"*
- *"Can I get some help?"*
- *"I have sent my request"*

train chatbot



Test data



Sequential learning

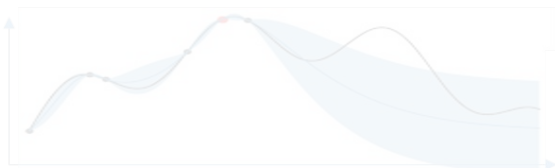
► Active Learning:

- Select images to label such that expected accuracy is maximized



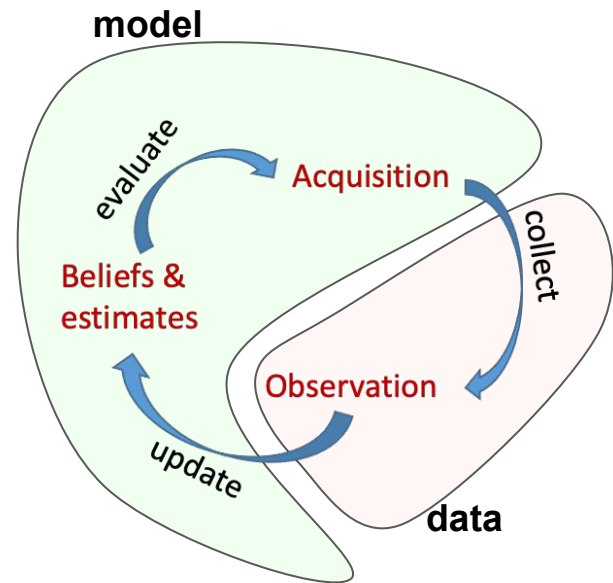
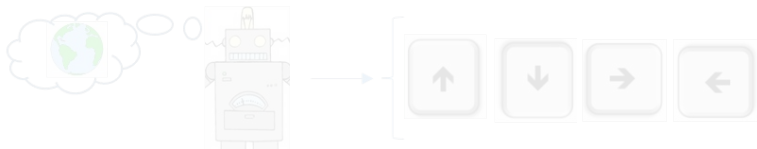
► Bayesian Optimization:

- Find the minimum of a function f



► Reinforcement Learning:

- Take K actions to collect maximum combined reward



Sequential learning

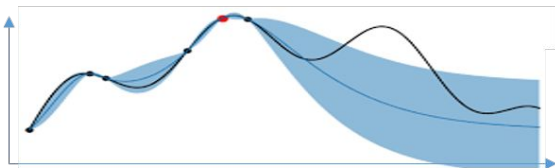
► Active Learning:

- Select images to label such that expected accuracy is maximized



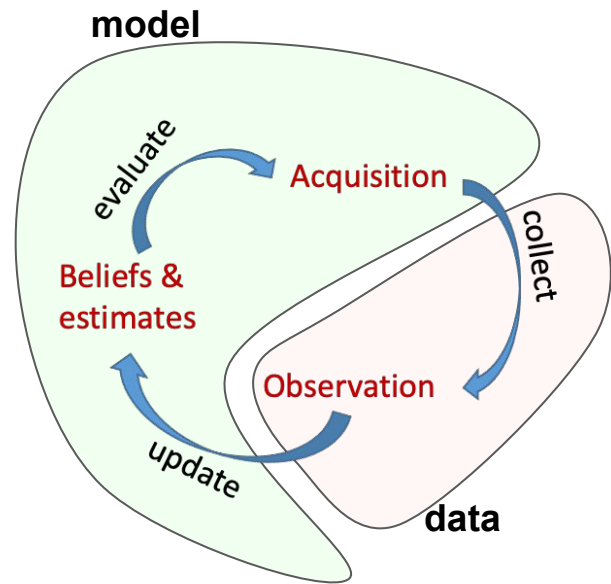
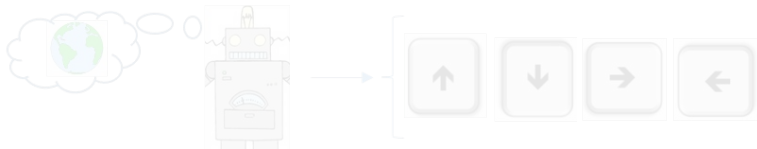
► Bayesian Optimization:

- Find the minimum of a function f



► Reinforcement Learning:

- Take K actions to collect maximum combined reward



Sequential learning

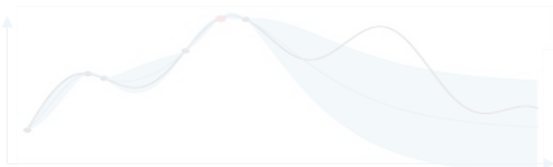
► Active Learning:

- Select images to label such that expected accuracy is maximized



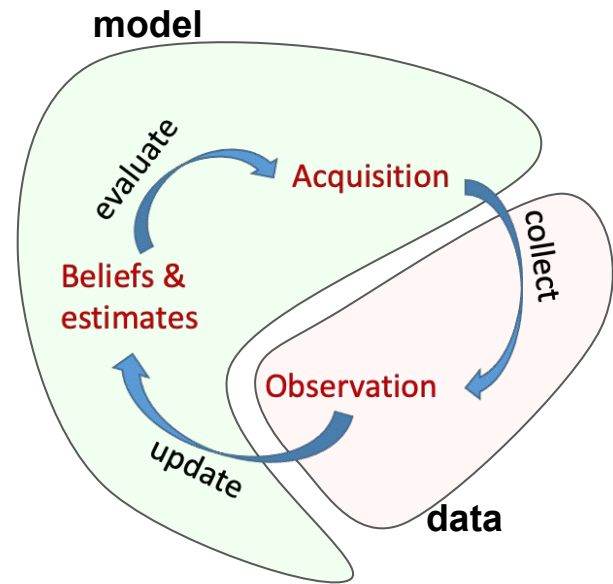
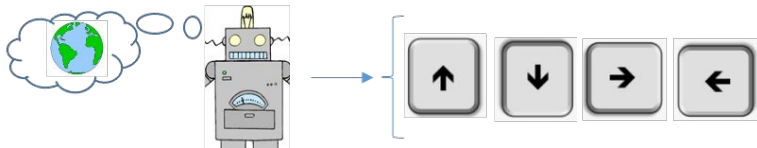
► Bayesian Optimization:

- Find the minimum of a function f



► Reinforcement Learning:

- Take K actions to collect maximum combined reward

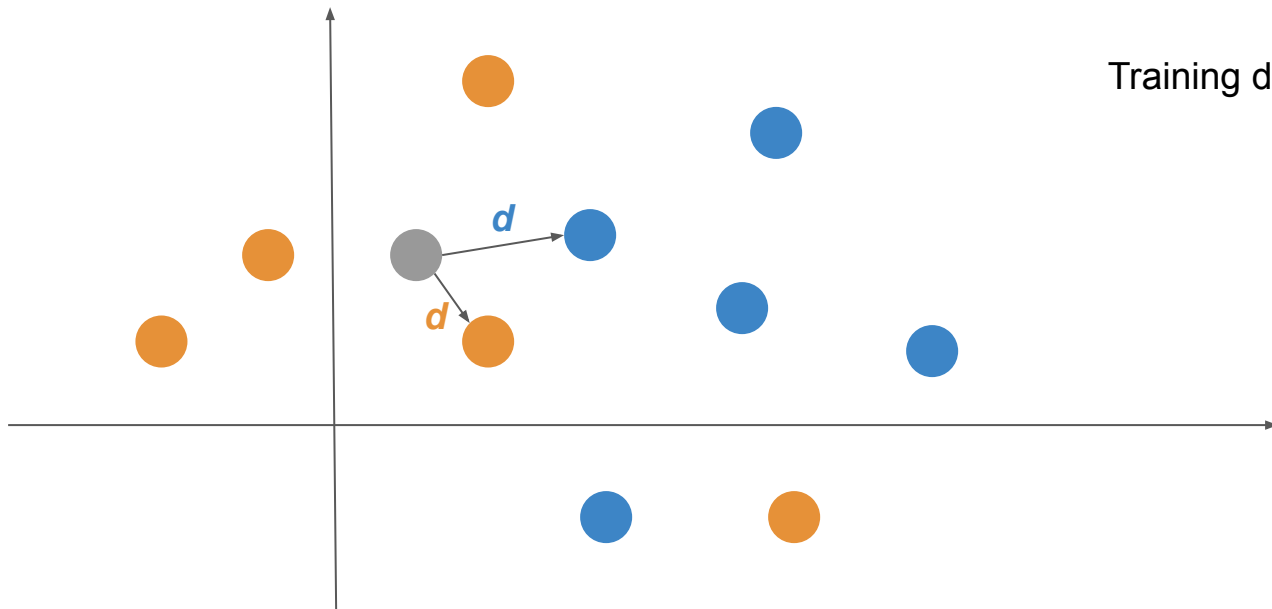


(Non)parametric learning

1-NN

$$\text{class}(\text{grey circle}) = \text{class of } \underset{\text{argmin}}{\text{d [grey circle , blue circle] , d [grey circle , orange circle]}}$$

Training data is part of the predictor!



Bias in model: algorithmic bias

Mozzarella → Prosciutto

Gorgonzola → Speck

Cheddar → Speck

Parmesan → Prosciutto

Brovola → Pineapple



Take Home Message

The way we go about modeling affects how we leverage information stored inside data and, in turn, it affects how inevitable data imperfections carry on to predictions.

Types of bias in data...

... and related issues and solutions

ML in a nutshell

Data



+

Model



compute



Inference

9
42
12

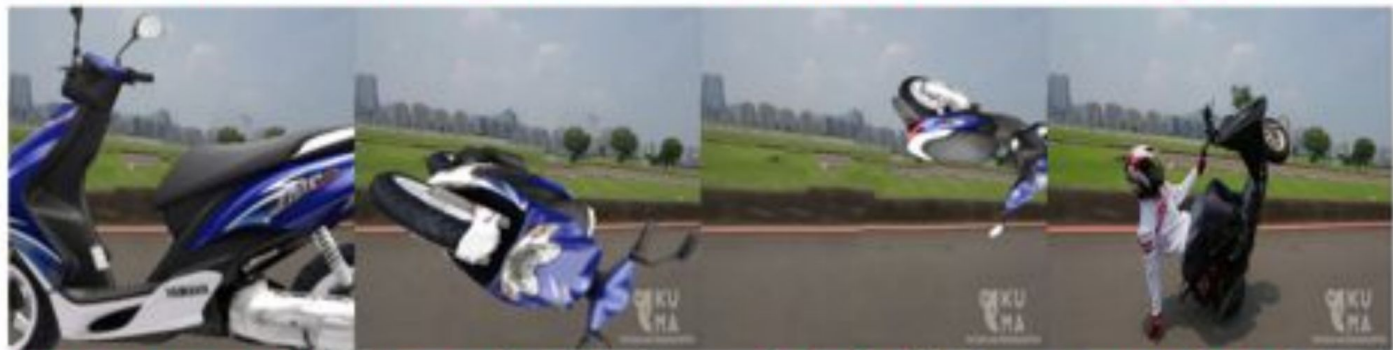
Garbage-in, garbage-out



Bias in data



school bus 1.0 **garbage truck** 0.99 **punching bag** 1.0 **snowplow** 0.92



motor scooter 0.99 **parachute** 1.0 **bobsled** 1.0 **parachute** 0.54

Bias in data: sample bias

Training conditions



Deployment conditions






We have sample bias when we perform data collection without proper randomization.


Bias in data: measurement bias



Bias in data: recall bias

Delivery  EXCELLENT 

Goods 

Customer service 

Your total rating 5.00/5.00
EXCELLENT

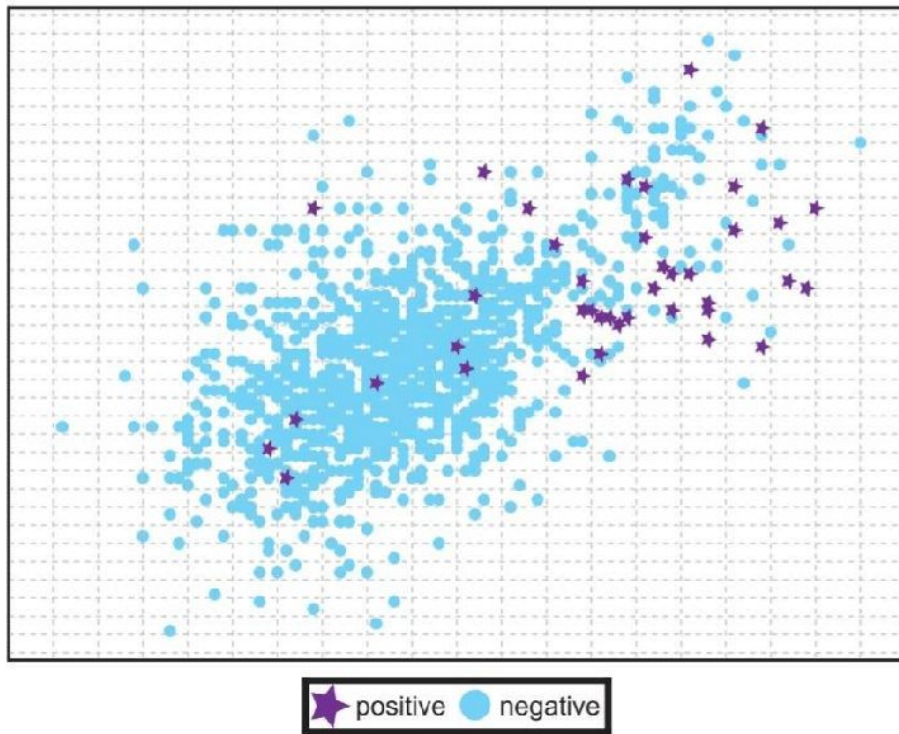
Your comment on this rating

Describe your experiences and give your review a personal touch.

still 400 characters possible

[Submit review](#)

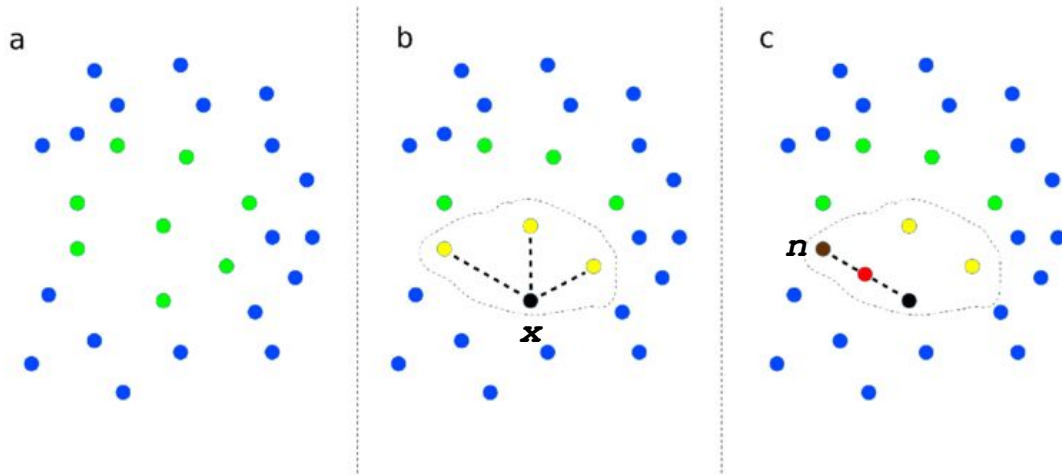
Imbalance data bias



Imbalance data bias - SMOTE

SMOTE as alternative to oversampling

1. Pick minority class point randomly, \mathbf{x}
2. Find its k neighbours
3. Pick one of those neighbours, \mathbf{n}
4. Generate a point between the line that connects \mathbf{x} and \mathbf{n}



Source: Imbalance-Aware Machine Learning for Predicting Rare and Common Disease-Associated Non-Coding Variants

Let's be clear about our assumptions

... and their consequences!

Problem: Classify risk for disease (1 = high risk, minority class)

False negative: Patient misses out on treatment, possibly dies

False positive: Patient takes an extra test.

SMOTE oversampling boosts the population of the minority class with artificial samples \Rightarrow

It leads to more instances predicted as minority class of which:

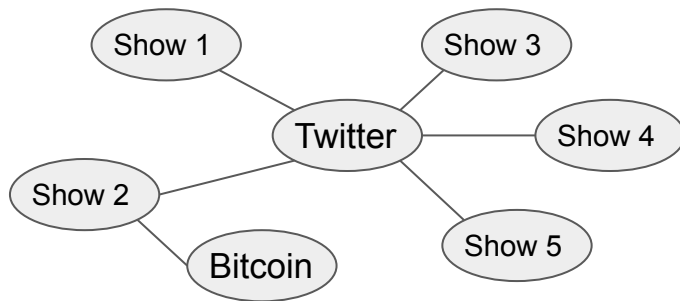
- Correctly classified as minority \rightarrow fewer false negatives (incr. recall)
- Incorrectly classified as minority \rightarrow more false positives (decr. precision)

Imbalance data bias: advice

If undersampling seems reasonable given the data availability, start with that strategy.

Other data biases

- Anomaly detection: Remove outliers. These tend to be:
 - Few
 - Different
- ... but so are “novelties”!
- “Stop” words besides NLP





Model capacity is one thing...

... model learnability (on a dataset) is another thing..!

Our models are not perfect learners. Removing “confusing” datapoints is a valid real-world strategy for improving learning.

Best practices for data collection

- Produce statistics of the training data for different slices (e.g. populations) to rule out biases
- Investigate bias-related hypotheses
 - Better with diverse team
 - Use domain knowledge / experts
- Use multiple data sources, where possible, and check for consistency
- Have formal characterization of labelling protocols and data readiness [1].

[1] *Data Readiness Levels*. N. Lawrence, 2017

[2] [Techopedia](#)

Data Readiness Levels

- Band C: Data accessibility
 - Lowest: “We should have clicks logged in some table”
 - Highest: “The track consumption data is in table *track.listens.xyz*”
- Band B: Faithfulness of data
 - Lowest: Provenance of training labels is unknown
 - Highest: Missing data are flagged and filled in using method xyz
- Band A: Data in context (e.g. of application)
 - Lowest: Unsure what kind of information we can leverage in this data.
 - Highest: This data is ready to be deployed for optimizing long-term rewards

Best practices for model analysis

- Enhance qualitative results (e.g. engagement metrics) with qualitative analysis (e.g. clustering, dimensionality reduction, visualization).
- Use tools of model inspection, e.g. “what-if tool”
 - *“Using WIT, you can test performance in hypothetical situations, analyze the importance of different data features, and visualize model behavior across multiple models and subsets of input data, and for different ML fairness metrics.”*

Performance

500 c

^

☒ Datapoints ☐ Partial dependence plots

Nearest counterfactual ⓘ ☒ L1 ☐ L2 Feature value ▼

Thres Nearest counterfactual (neighbor of different classification)

Compares the selected datapoint with its nearest neighbor from a different classification using L1 or L2 distance. If a custom distance function is set, it uses that function instead.

[Explore a tutorial on using counterfactuals.](#)

Descending

-0.3 0.0 0.3

Feature	Value(s)	Counterfactual value(s)
---------	----------	-------------------------

relationship Husband Husband

capital-gain	3103	0
--------------	------	---

native-country	United-States	United-States
----------------	---------------	---------------

marital-status	Married-civ-spouse	Married-civ-spouse
----------------	--------------------	--------------------

^

Predict

(none)

(none)

Inference v_ε ▼

(default) ▼

Inference va

(c)

+

Legend

Colors

by In

- 64.7

● 53.5

- 42.2

Take Home Message

Bias and other types of noise/dirtiness in data can induce (often silent) problems in model deployment, often affecting sensitive aspects such as fairness.

It is essential to not treat data as a black box input, but strive to fully understand where it's coming from and how it should be sanitized. Exploration tools, policy tools (e.g. data rediness) and mitigation strategies (e.g. fixing imbalance) to the rescue.

Uncertainty in data/model

ML in a nutshell (in theory)

Data



+

Model



compute



Inference

9
42
12

ML in a nutshell (in reality)

Data



+

Model



compute

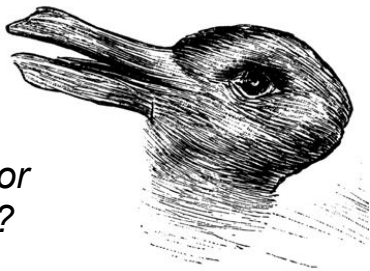


Inference

9
42
12

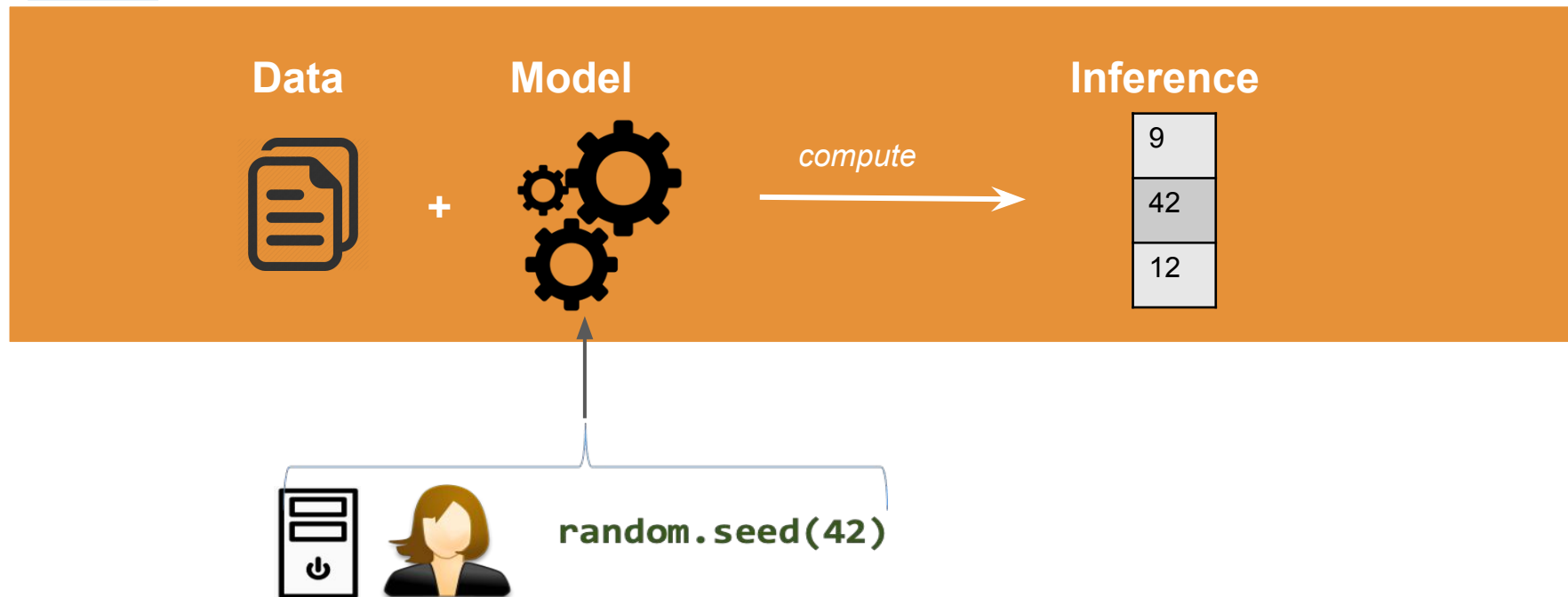
[INACCESSIBLE]

Duck or
rabbit?



Our model only sees **partial** and **noisy data**.

ML in a nutshell (in reality)



Our model is **imperfect**, possibly **biased** and often inherently **random**.

ML in a nutshell (in reality)

Data



+

Model



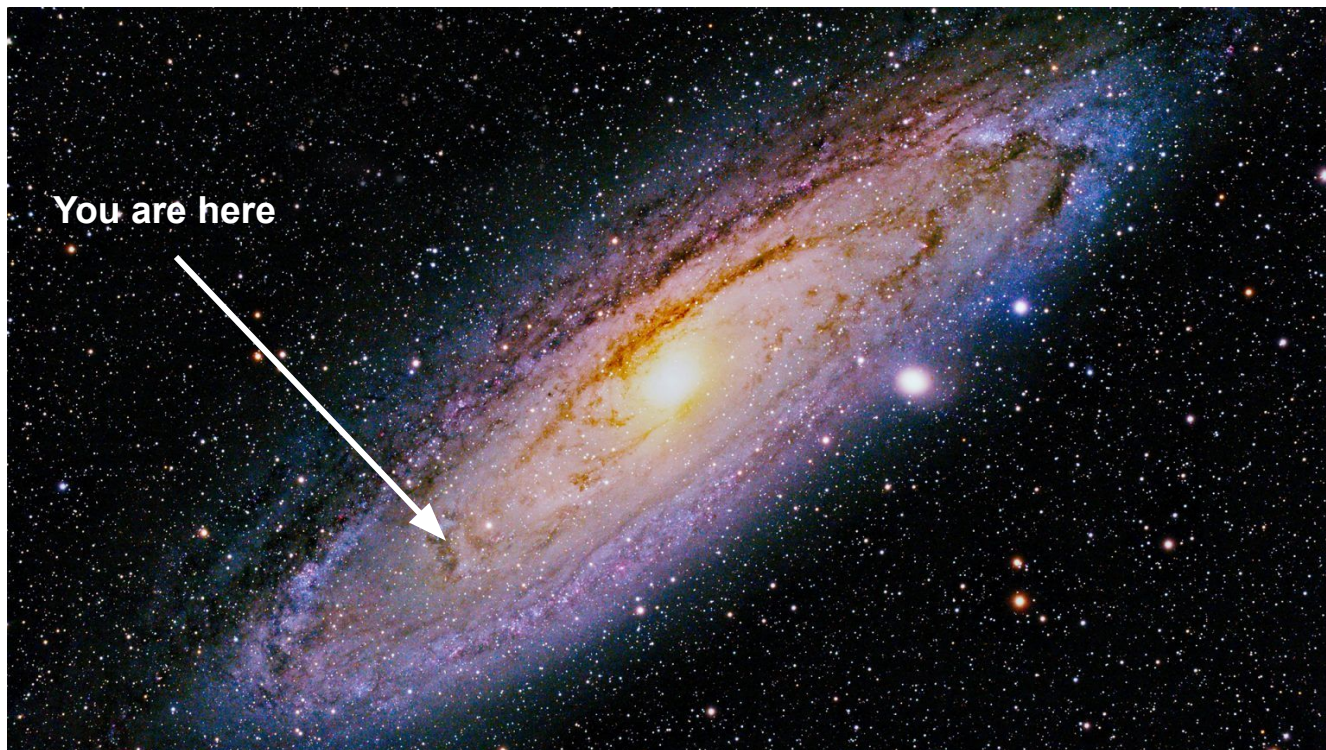
compute



Inference

Hot
Cold
Hot

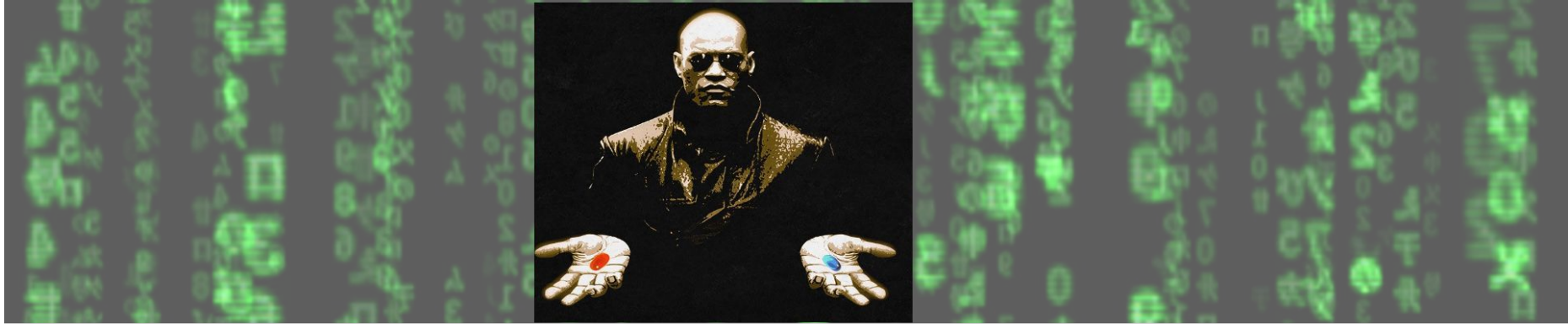
Interpretation of concepts can be **fuzzy**.



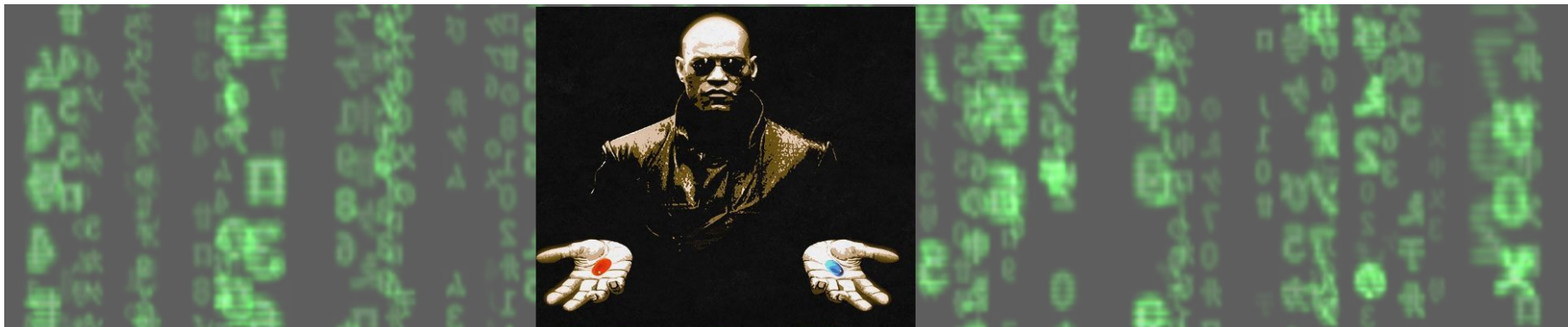
No matter what model we use, no matter how carefully we've collected data... we will *a/ways* have incomplete view of the domain.

This can be baked in as **uncertainty** in our model.

Uncertainty in modeling



Uncertainty in modeling



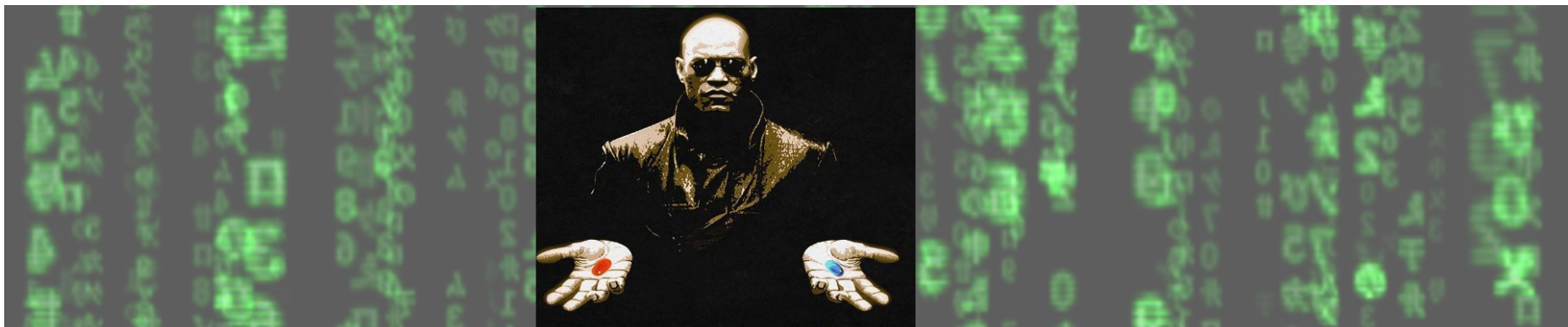
Epistemic uncertainty

Neo doesn't know that he lives in a simulation
(Ignorance about the correct model that
generated the data e.g. Matrix glitches)

Aleatoric uncertainty

Neo knows that he lives in a simulation but the
simulation's complexity introduces *inherent* uncertainty
(not enough capacity to perfectly observe the world)

Uncertainty in modeling



Epistemic uncertainty

Neo doesn't know that he lives in a simulation
(Ignorance about the correct model that
generated the data e.g. Matrix glitches)

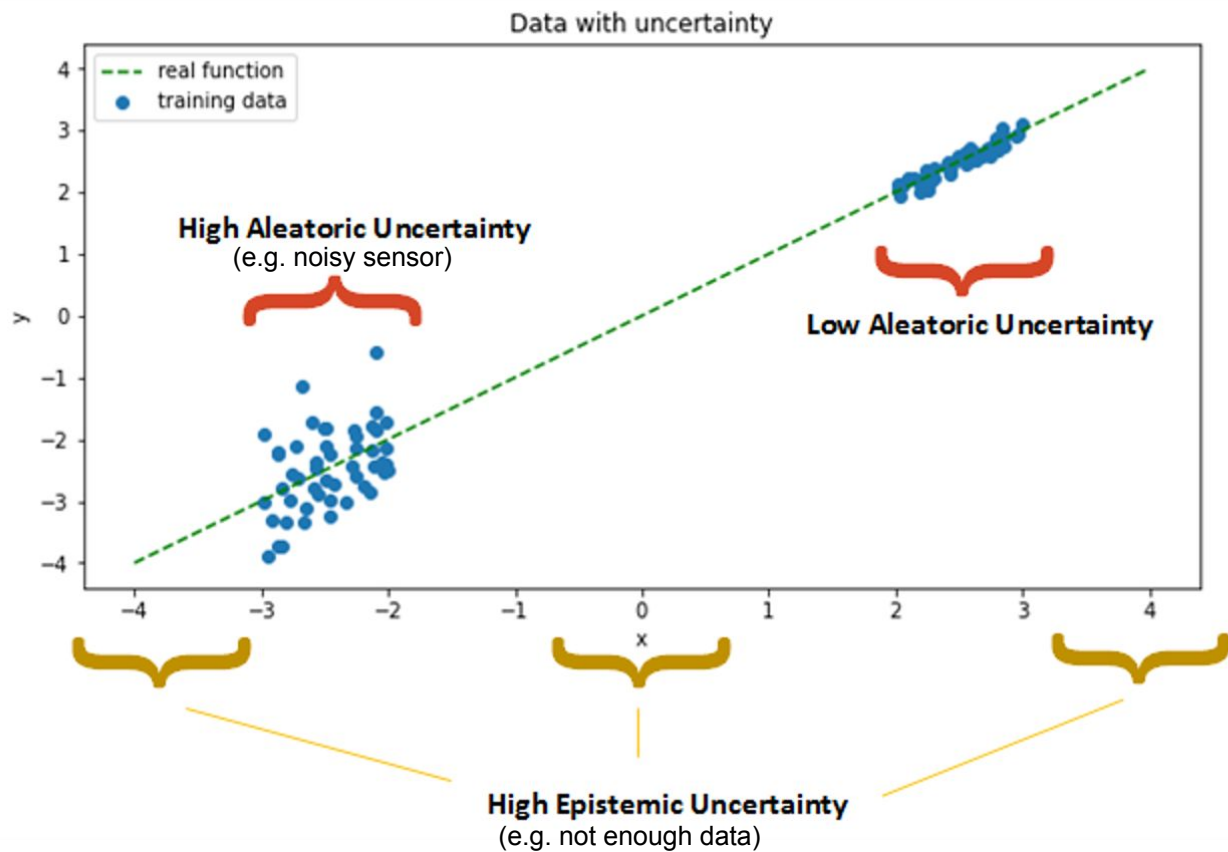
Aleatoric uncertainty

Neo knows that he lives in a simulation but the
simulation's complexity introduces *inherent* uncertainty
(not enough capacity to perfectly observe the world)

Epistemic: Our model does not know how to pick one explanation of the data over another. With more data we can reduce this.

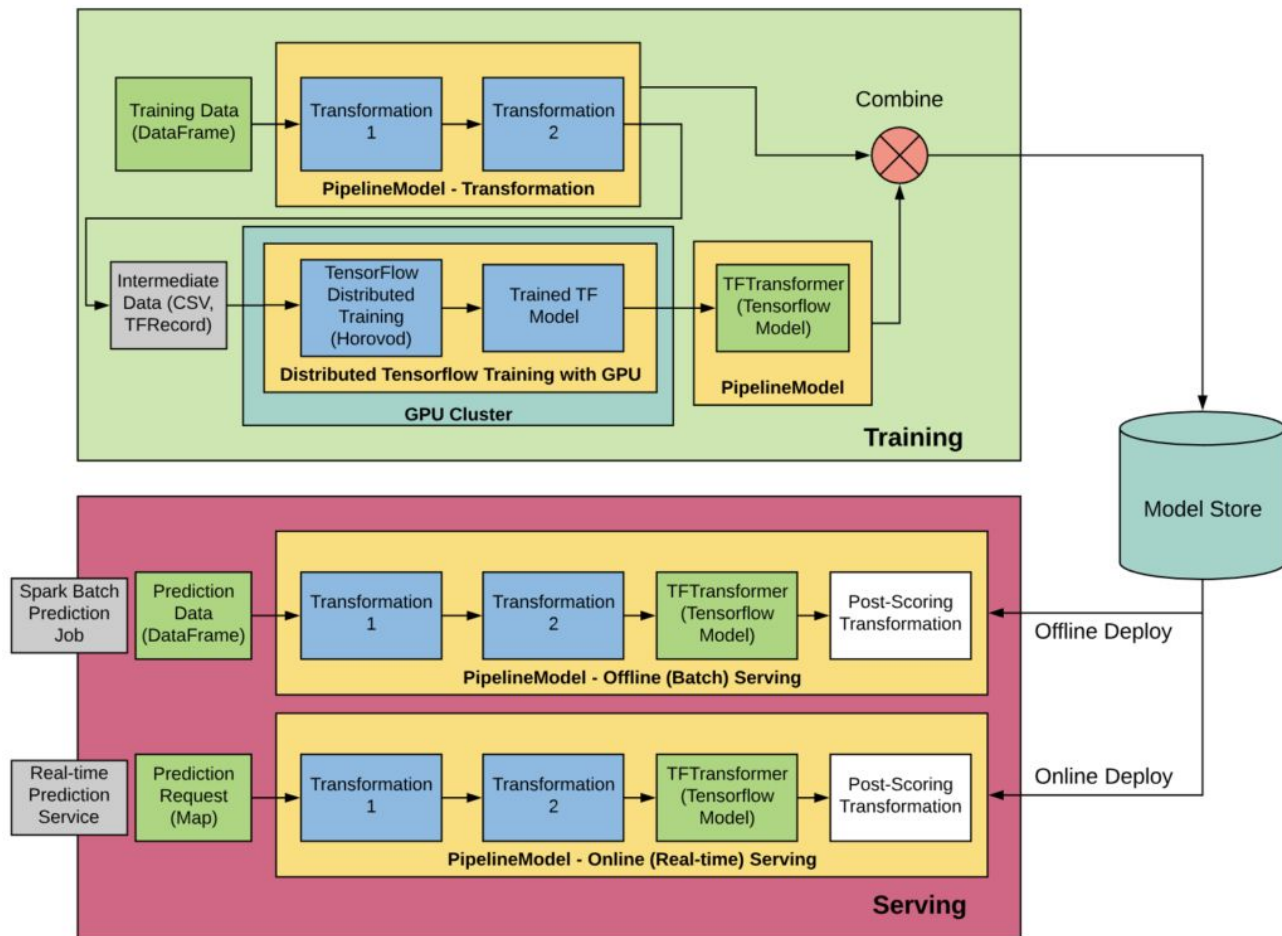
Aleatoric: Uncertainty inherent in observation noise. We cannot in principle reduce it if we get more and more observations of the world, since it all the sources in this world that cause apparent stochasticity.

Noise & Uncertainty

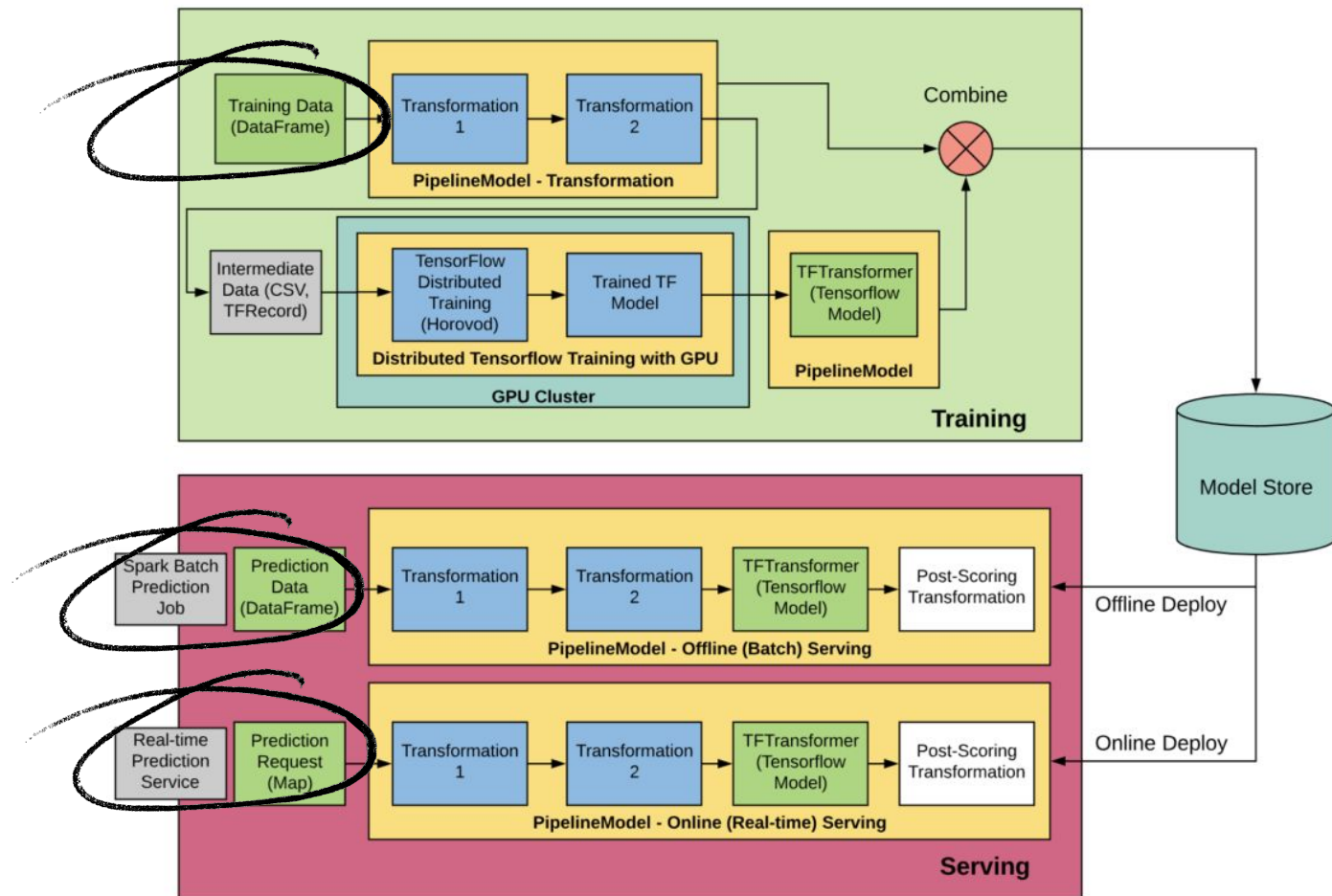


Data tools for industrial scale

Big picture: A ML pipeline

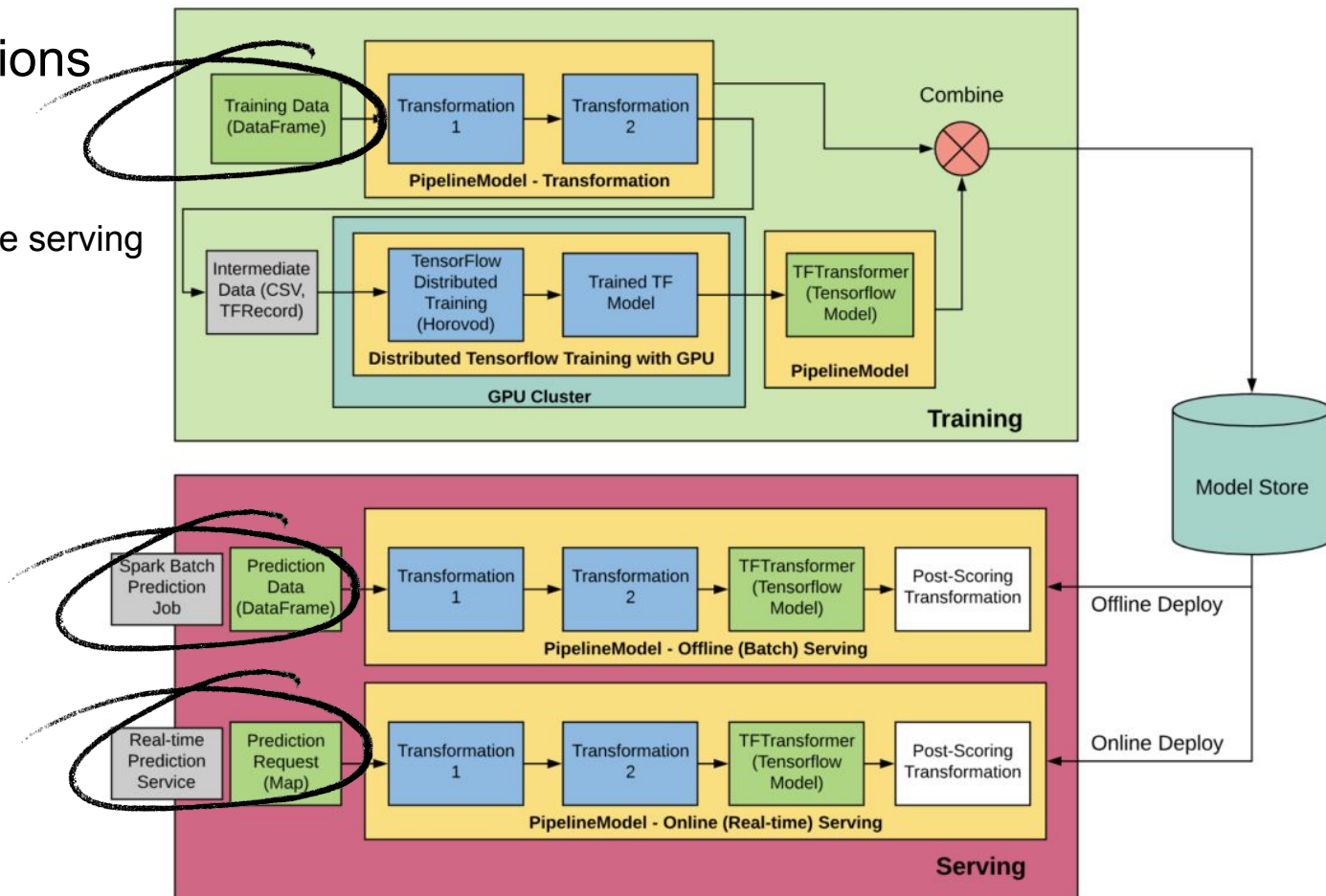


Big picture: A ML pipeline



Data Considerations

- Low latency for online serving
- Memory efficiency
- Scaling



Scalability



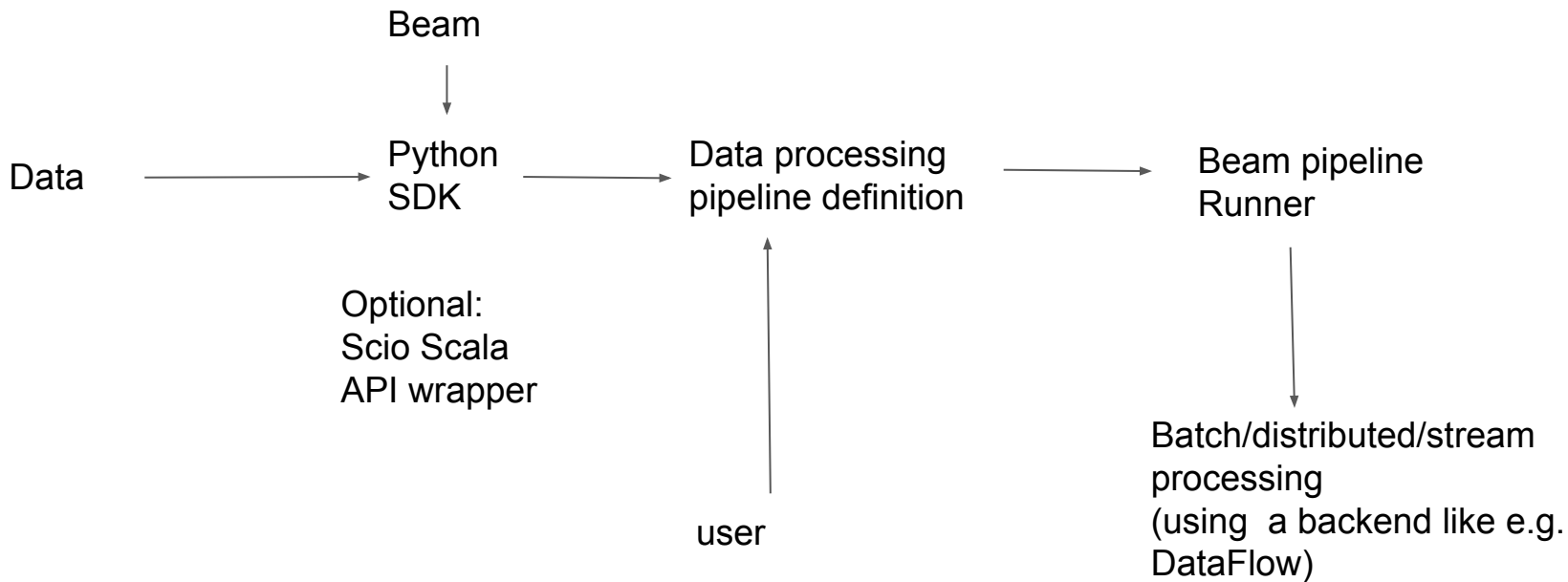
Vertical Scaling
(Scaling up)

Horizontal Scaling
(Scaling out)

Source: [cloudzero](https://cloudzero.com)

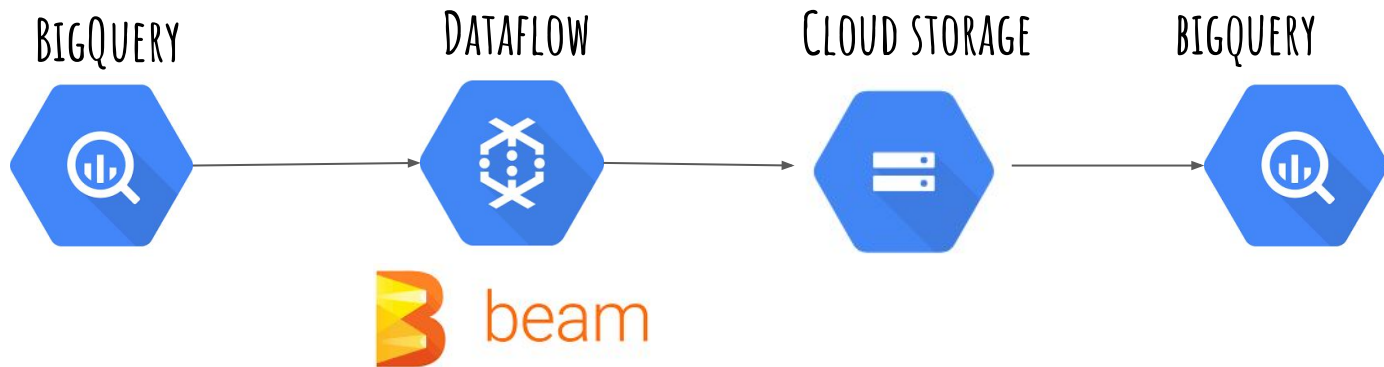
Data Workflow managers for scalability

E.g. Apache Beam (or Airflow, Spark, ...)



A unified framework

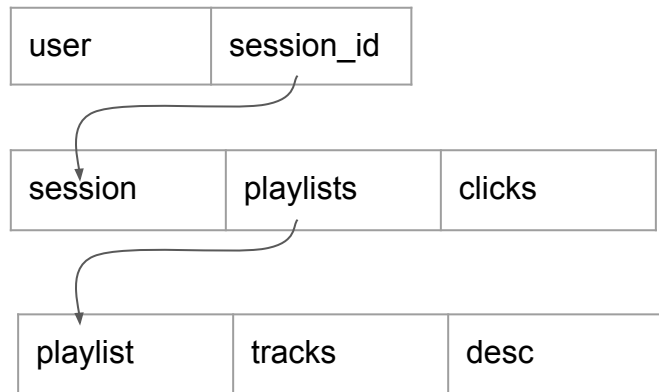
- Typically companies subscribe to a cloud service (AWS, Google Cloud, Microsoft Azure) that provides a consistent ecosystem of tools and services



Storage - How: (No)SQL

SQL has defined columns, strictly defined, with keys and foreign keys.
NoSQL is a bunch of files.

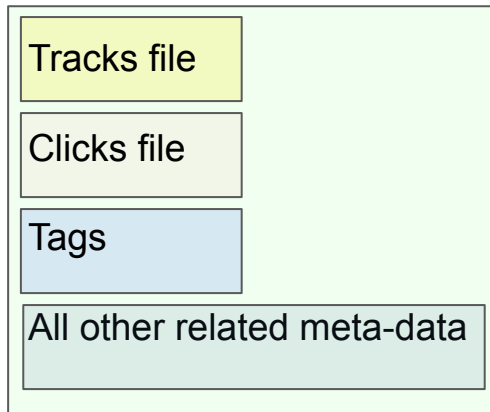
SQL (relational database)



```
SELECT user, tracks
FROM `sessions`
JOIN playlists on session_id
WHERE...
```

NoSQL

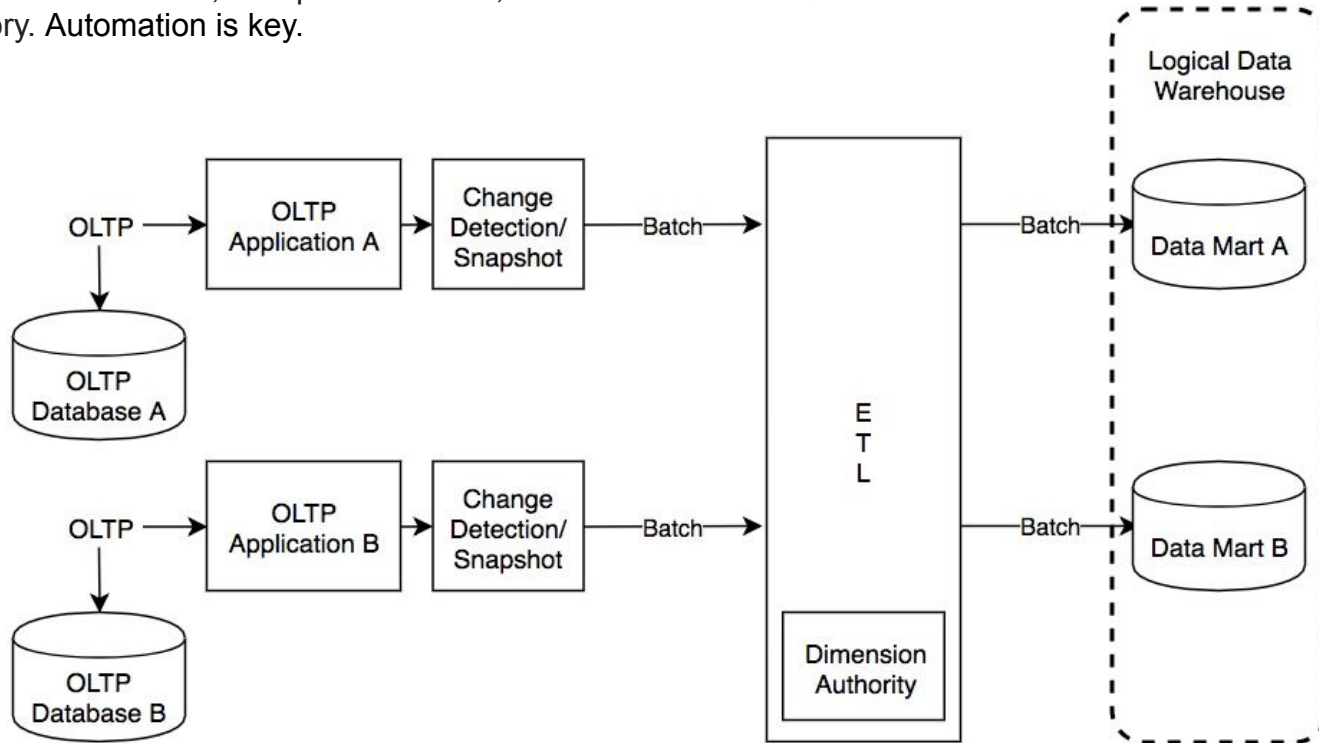
User Session bucket:



```
db.sessions.find({"tags": "classic rock"})
```

Data pipelines deep dive: Extract Transform Load (ETL)

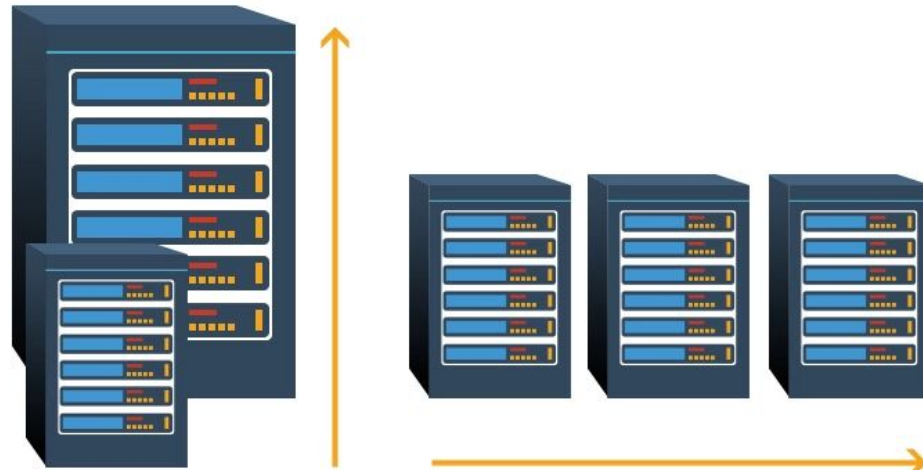
Move data from one database, multiple databases, or other sources to a unified repository. Automation is key.



Data pipelines deep dive: Extract Transform Load (ETL)

- **Extract:** E.g. request data from relational database or multiple sources.
- **Transform:** E.g. clean data, or standardize (“No” -> 0, “Yes” -> 1), aggregate etc
- **Load:** E.g. store in target data warehouse

Scalability



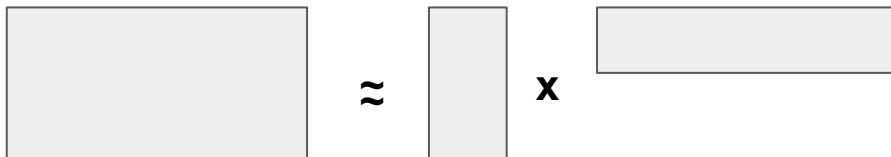
Vertical Scaling
(Scaling up)

Horizontal Scaling
(Scaling out)

Source: [cloudzero](https://cloudzero.com)

Model compression

- **What?** Reduce the size of a trained ML model (e.g. fewer parameters)
- **Why?** Reduce memory/speed/energy consumption/cost requirements
- **How?**
 - Quantization (0.23 \rightarrow 0.2) i.e. reduce the parameters size (in bits)
 - Sparsity (e.g. for conv. filters):



- Pruning: Remove connections between neurons
- Distillation

Knowledge distillation

A popular method for transfer learning.

However the principle stems from compression. The student model is typically smaller than the teacher model.

Further, a student model can distil information from multiple teachers.

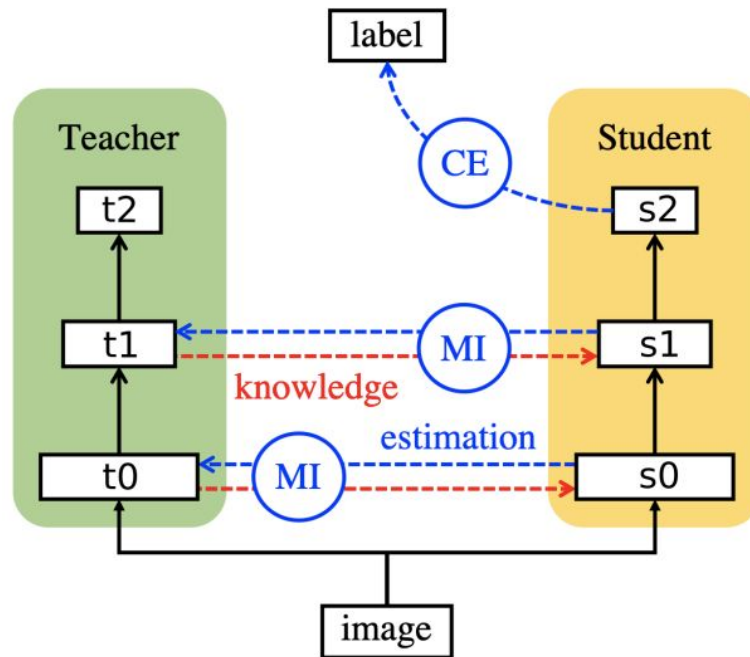
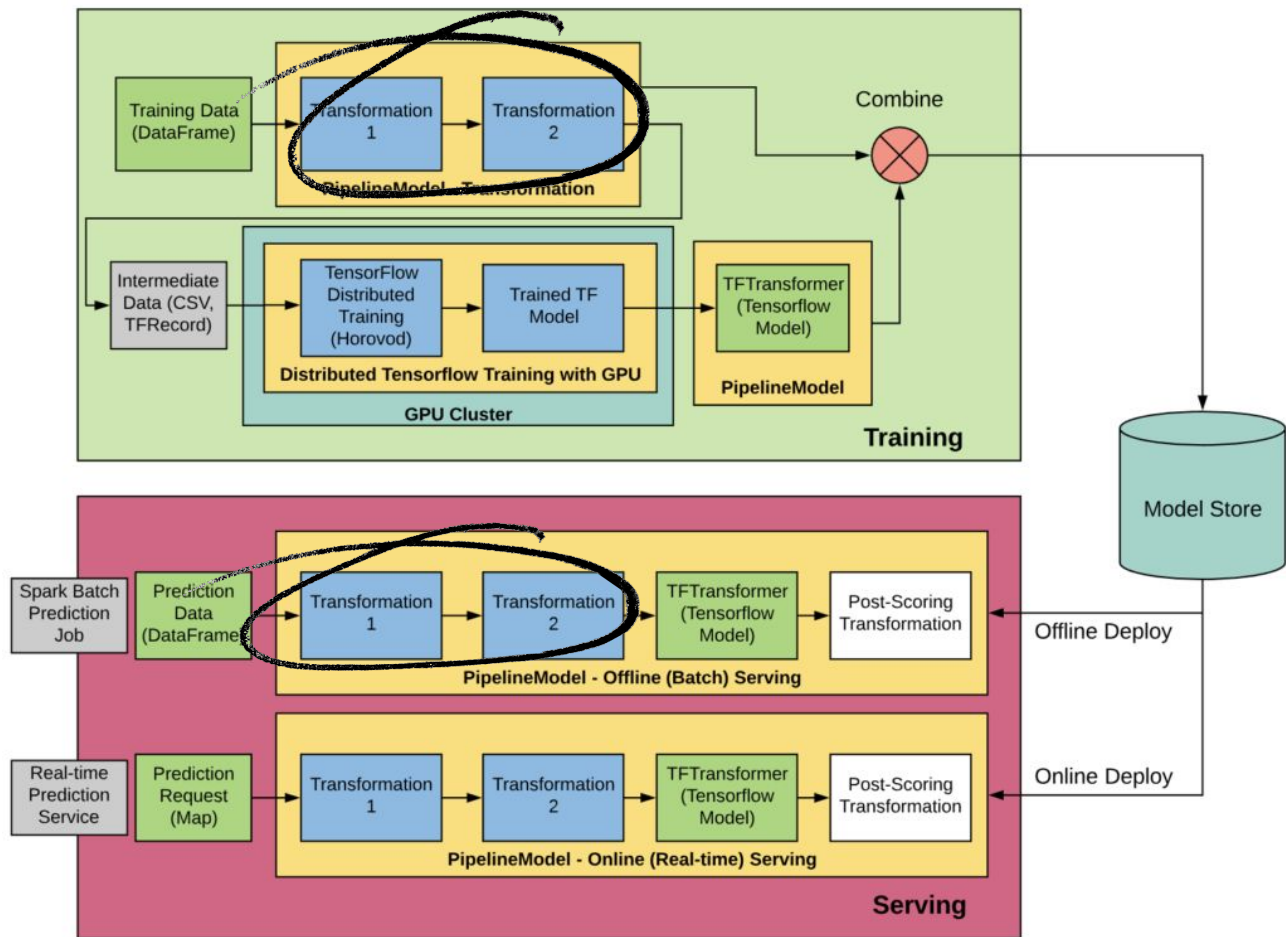


Figure 1: Conceptual diagram of the proposed knowledge transfer method. The student network efficiently learns the target task by minimizing the cross-entropy (CE) loss while retaining high mutual information (MI) with the teacher network. The mutual information is maximized by learning to estimate the distribution of the activations in the teacher network, provoking the transfer of knowledge.

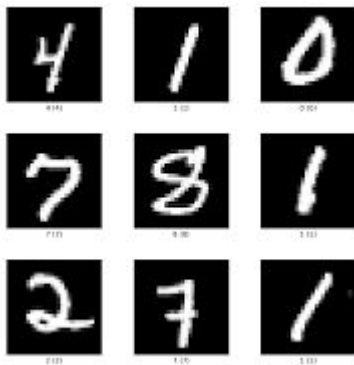
From raw data to ML features

Big picture: A ML pipeline



Data cleaning

Academic data



Real data

	Class	AGE	SEX	STEROID	ANTIVIRALS	FATIGUE	MALAISE	
0	0	30	2	1.0	2	2	2	
1	0	50	1	1.0	2	1	2	
2	0	78	1	2.0	2	1	2	
3	0	31	1	NaN	1	2	2	
4	0	34	1	2.0	2	2	2	

Data cleaning

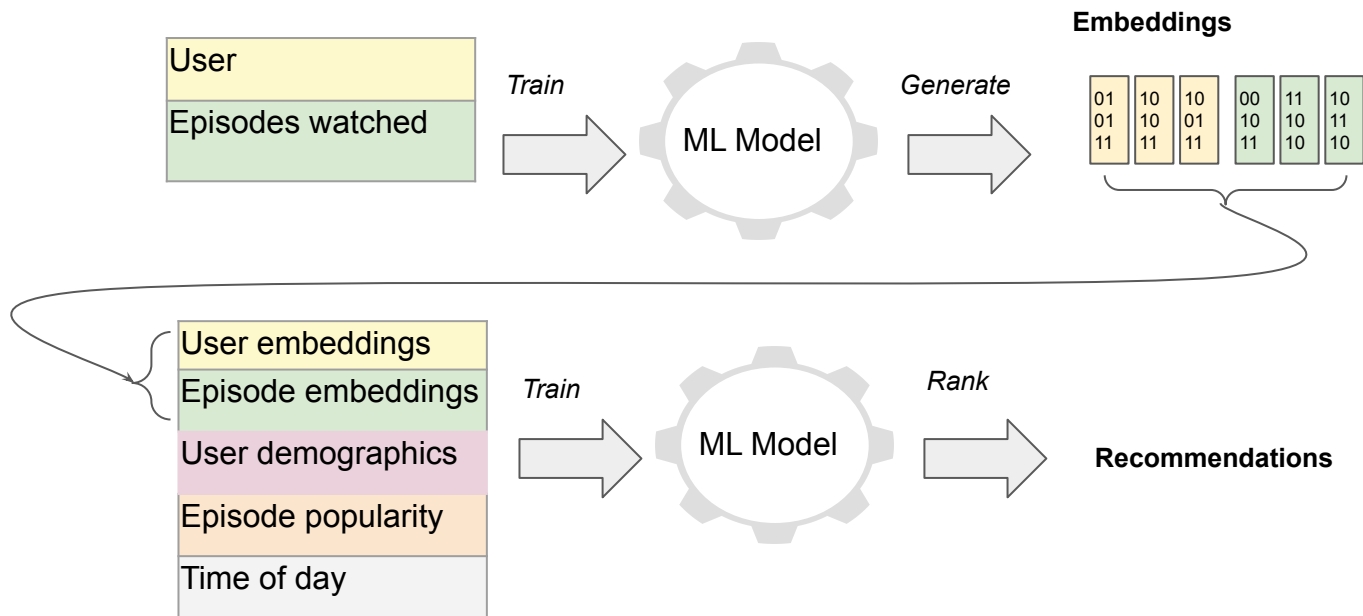
- Check **validity** (age > 0), **data types**...
- Check **consistency** (0 vs "0", N/A vs NaN) and uniformity ("2 days" vs "48 hrs")
- Remove **duplicates**, **outliers** etc.
- Deal with **missing data**
 - Delete rows/columns
 - Impute with arbitrary value / mean / most frequent
- Reassign unseen categorical values:
 - Training: [0, 3, **2**, 7, **2**, 9, 8]
 - Test: 1 → **2**

Featurization

- One-hot encoding, bag of words etc.
- Keeping in mind the nature of the data and production constraints. E.g. text features might be better featurized with language models (if latency is acceptable) or tf/idf.
- Automatic featurization solutions exist, e.g. Azure AutoML.

Embeddings

Often the output of a ML workflow is stored and used as feature for another ML workflow. For example, learned embeddings. Such dependencies need to be carefully monitored - ***when something changes it has downstream effects.***



For the second model, the embeddings are similar to "raw data"

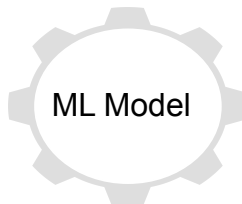
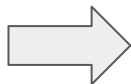
Time evolution considerations

New data

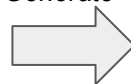
1/1/2021 - 1/1/2022

User
Episodes watched

Train



Generate



Embeddings

01	10	10	00	11	10
01	10	01	10	10	11
11	11	11	11	10	10

1/1/2022 - 25/5/2022

User
Episodes watched

Solution 1: Retrain

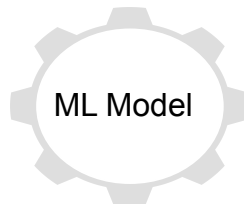
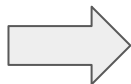
1/1/2021 - 1/1/2022

User
Episodes watched

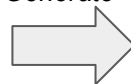
1/1/2022 - 25/5/2022

User
Episodes watched

Train



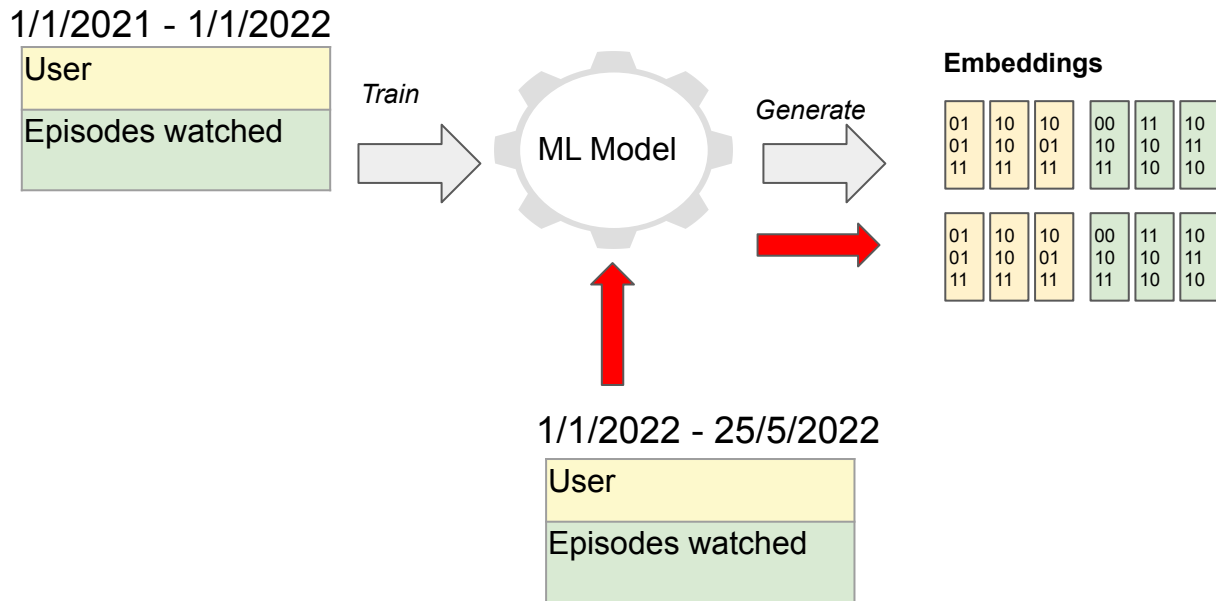
Generate



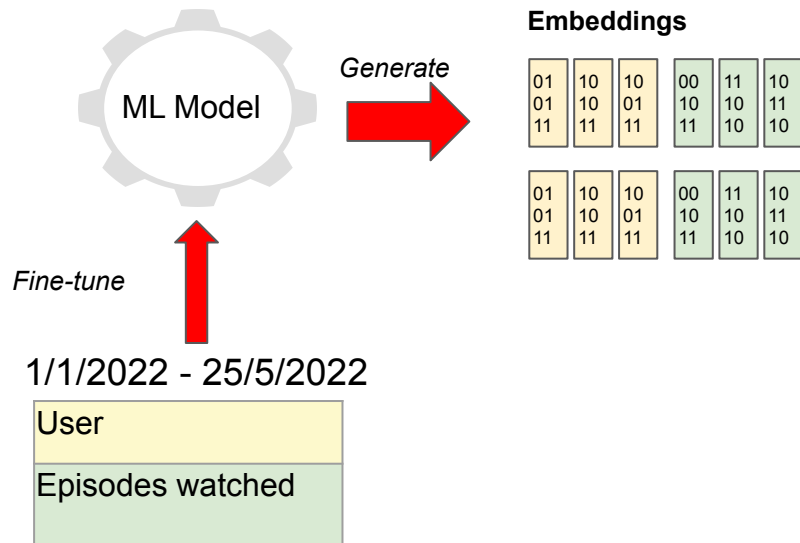
Embeddings

01	10	10	00	11	10
01	10	01	10	10	11
11	11	11	11	10	10
01	10	10	00	11	10
01	10	01	10	10	11
11	11	11	11	10	10

Solution 2: Inductive learning



Solution 3: Continual learning



Coresets

Distribution shift - What?

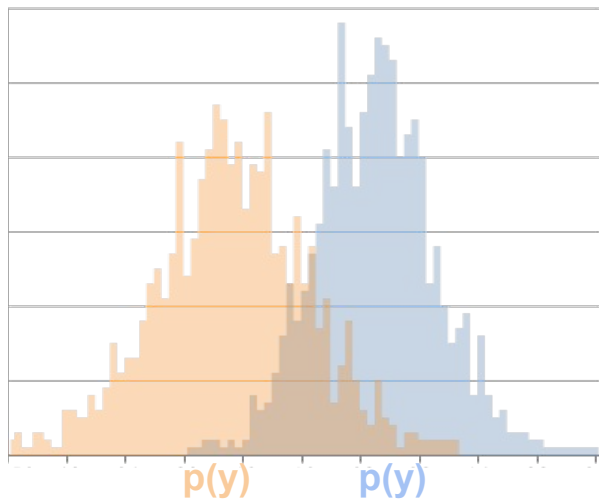
- Let's assume we have input/output pairs of data: x, y .
- These form a joint distribution: $p(x, y)$
- When this distribution changes, we can no longer use the trained model reliably.
- We can think of cases where our business expands to a new market (e.g. India) and thus our trained models are not suited for the new population.
- But usually distribution shift refers to more nuanced, temporal causes.

Distribution shift - Running example

- Context: A chain of physical shops
- $x \rightarrow$ item/user features
- $y \rightarrow$ expected number of purchases
- Model: predict y from x

Distribution shift - Prior probability shift

Same $p(x|y)$
Different $p(y)$



E.g. The user/item features we track haven't changed (same $p(x)$) but sales have gone down because lockdown forced people to stay home (different $p(y)$).

Distribution shift - Covariate shift

Same $p(x|y)$
Different $p(x)$

E.g. The shops chain has expanded from US to UK (different $p(x)$ since we have different user data), but it turns out that the UK population shopping habits are governed by the same functional relationship between x and y .

This is a source of bias in the data that can affect predictions (we have test cases which do not occur in the training set).

Distribution shift - Concept drift

Different $p(x|y)$ or $p(y|x)$

i.e. the relationship between the variables has changed.

For example, this can occur when the environment is non-stationary, e.g. shopping trends and the overall market simply make consumers evolve their preferences.

In an industrial setting, it is important to always consider the business context that generated the data we are asked to model. E.g sales data from around Christmas should not be representative.

Data / model versioning



```
$ git diff data.dat
diff --git a/test/data.dat b/test/data.dat
index cb6f284..6c52c7f 100644
Binary files a/test/data.dat b/test/data.dat differ
```



```
$ dvc add data/data.dat
$ git add data/data.dat.dvc data/.gitignore
$ git commit -m "Added data"
```

Data / model versioning

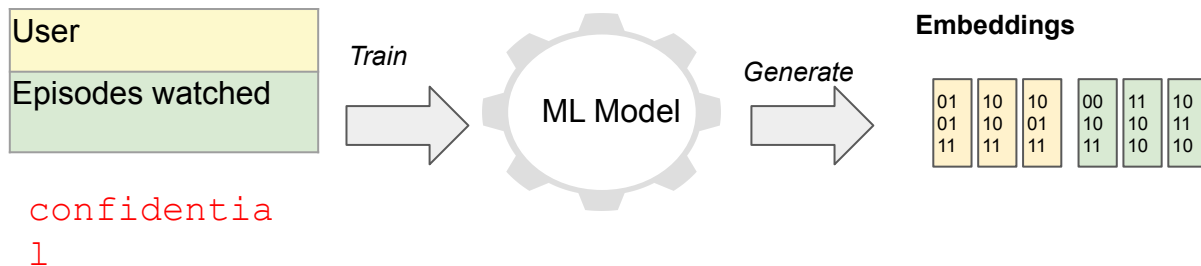
- Data Version Control (DVC) allows to keep track of data and models
- Promotes **reproducibility, collaboration** and **transparency**
- Facilitates **rollback**
- Can be thought of as git but for data/models
- Can be used with a local or remote server

Privacy and regulation

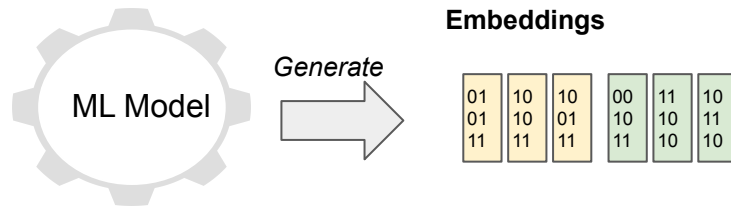
Data protection

- Naturally, a company's data are confidential to protect the data's source (e.g. user data) and the company (e.g. from competitors) as well as for complying with the law.
- Main complementary approaches for protection: **policies** and **access/encryption tools**
- Data confidentiality is further classified into various buckets with strict policies about:
 - How they're stored
 - How long they're stored for
 - Who may access them
 - Through what channels
- Data categorization often is informed by government and organization policies e.g. the US national security classification scheme (confidential, secret, top secret).

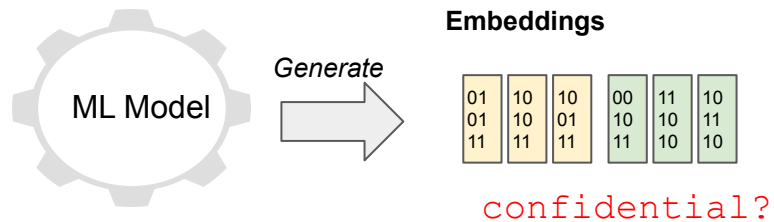
What about ML?



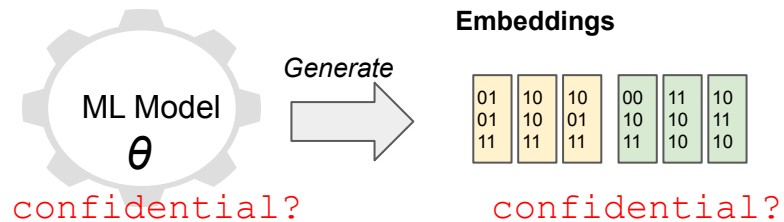
What about ML?



What about ML?



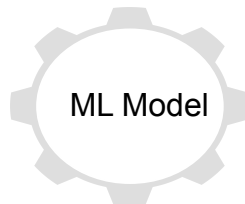
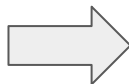
What about ML?



What about ML?

User	Episodes watched
X	A, B
Y	B, C
Z	D

Train



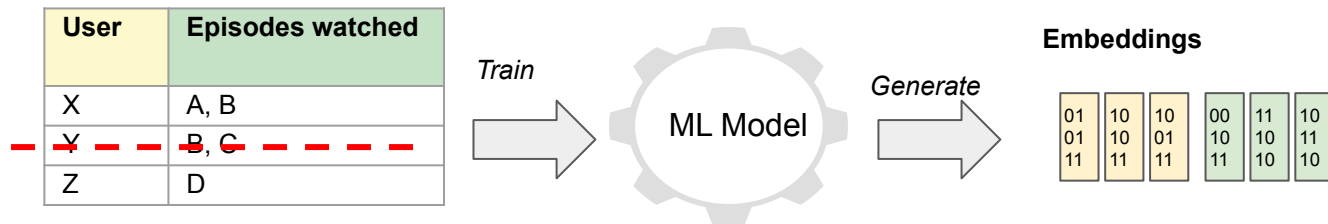
Generate



Embeddings

01	10	10	00	11	10
01	10	01	10	10	11
11	11	11	11	10	10

What about ML?



User Y requests deletion of their records.

Can we make the model “un-learn” all the information of user Y? Even indirect influences?

→ (Adaptive) Machine Unlearning [3,4]

Many open problems exist and provability is hard.

[3] *Machine Unlearning*, Bourtole et al. 2019

[4] *Adaptive Machine Unlearning*, Gupta et al. 2021

What about parametric models?

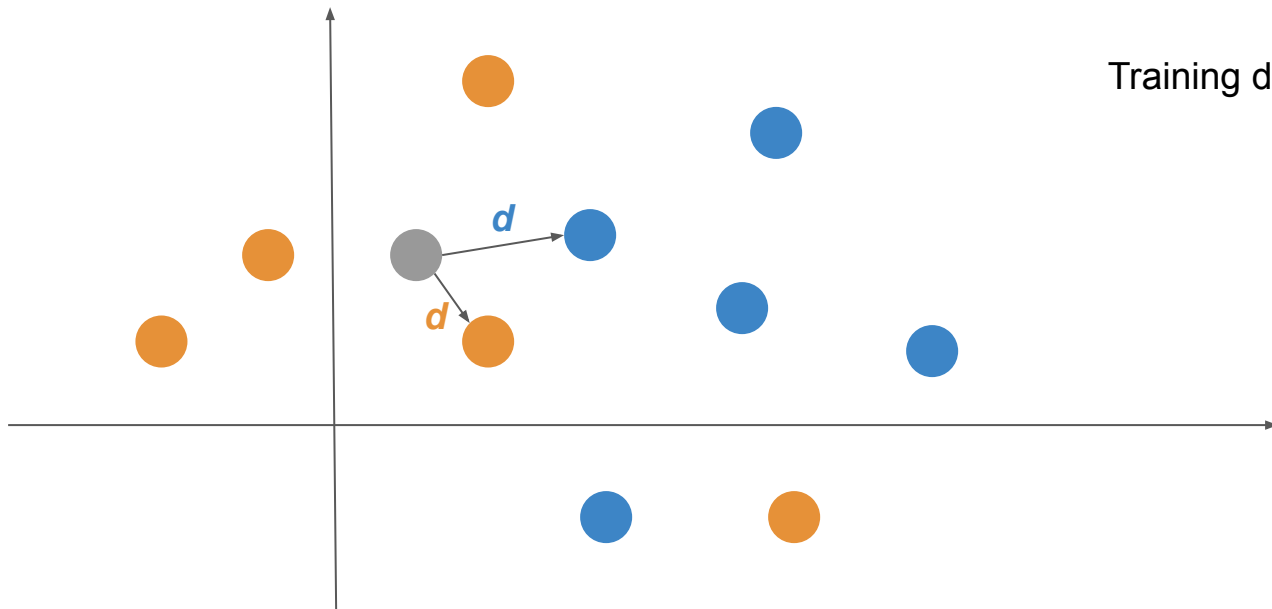
At which portion of the parameters we have user's Y influence?

What about non-parametric models?

1-NN

$$\text{class}(\text{grey circle}) = \text{class of } \underset{\text{argmin}}{\text{d [grey circle , blue circle], d [grey circle , orange circle]}}$$

Training data is part of the predictor!



k-anonymity

...aka hiding in the crowd.

Name	post-code	age
Paul B	AB129KL	56
Louise P	AB150NP	59
Paul M	AB152VV	23
Helen V	FN822JJ	76
Nick K	FN800SD	43
Sean K	FN877NN	90

k-anonymity

suppression

generalisation

Name	post-code	age	Age range
Paul B	AB1xxxx	56	55 - 65
Louise	AB1xxxx	59	55 - 65
Paul M	AB1xxxx	47	45-55
Helen	FNxxxx	76	75-85
Nick K	FNxxxx	68	65-75
Sean K	FNxxxx	90	85-95

2-anonymous

Not anonymous

[REDACTED]

[REDACTED]

k-anonymity

suppression

generalisation

Name	post-code	age	Age range
Paul B	AB1xxxx	56	45 - 60
Louise	AB1xxxx	59	45 - 60
Paul M	AB1xxxx	47	45 - 60
Helen	FNxxxx	76	65+
Nick K	FNxxxx	68	65+
Sean K	FNxxxx	90	65+

3-anonymous

The diagram illustrates k-anonymity using a table of personal data. The table has four columns: Name, post-code, age, and Age range. The first three rows (Paul B, Louise, Paul M) are grouped together by a bracket on the right, labeled '3-anonymous'. These rows have a pink background for the post-code and age range columns. The last three rows (Helen, Nick K, Sean K) have a green background for the post-code and age range columns. Red dashed boxes labeled '[REDACTED]' are placed over the Name and age columns for the first three rows. Arrows point from the text 'suppression' to the redacted Name column and from 'generalisation' to the redacted age column. A large bracket on the right groups the first three rows, indicating they form a 3-anonymous set.

Slide from: *borjaballe.github.io*

Anonymization Fiascos

Disturbing Headlines and Paper Titles

- ▶ “A Face Is Exposed for AOL Searcher No. 4417749” [Barbaro & Zeller '06]
- ▶ “Robust De-anonymization of Large Datasets (How to Break Anonymity of the Netflix Prize Dataset)” [Narayanan & Shmatikov '08]
- ▶ “Matching Known Patients to Health Records in Washington State Data” [Sweeney '13]
- ▶ “Harvard Professor Re-Identifies Anonymous Volunteers In DNA Study” [Sweeney et al. '13]
- ▶ ... and many others

In general, removing identifiers and applying anonymization heuristics is not always enough!

Why we need privacy guarantees

Ethnicity
Visit date
Diagnosis
Procedure
Medication
Charge
ZIP
Birth date
Sex

*Health data anonymized for
research*

Name
Address
Date Registered
Party affiliation
Date last voted
ZIP
Birth date
Sex

Voter list

Why we need privacy guarantees

Ethnicity
Visit date
Diagnosis
Procedure
Medication
Charge
ZIP
Birth date
Sex

*Health data anonymized for
research*

Name
Address
Date Registered
Party affiliation
Date last voted
ZIP
Birth date
Sex

Voter list

Data is anonymized... that should be OK... right?

Why we need privacy guarantees

Ethnicity
Visit date
Diagnosis
Procedure
Medication
Charge
ZIP
Birth date
Sex

*Health data anonymized for
research*

Name
Address
Date Registered
Party affiliation
Date last voted
ZIP
Birth date
Sex

Voter list

Possible to identify medical
records for the Governor of MA:

- 6 people with the same **birth date**
- Of which 3 were **men**
- Of which he was the only one registered in his 5-digit **ZIP** code

Data is anonymized... that should be OK... right?

Wrong! Data can be linked!

Differential privacy

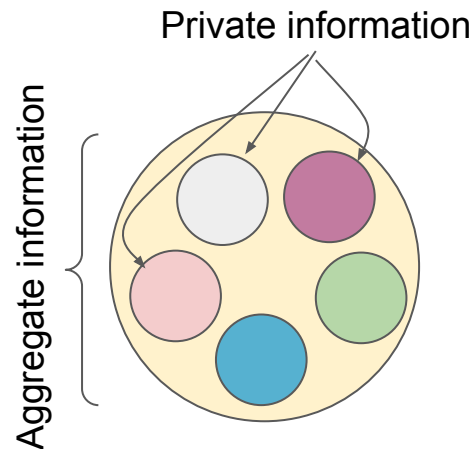
...aka plausible deniability

- A **definition** of privacy, rather than a specific technique.
- It sets this **requirement**:
“Any information-related risk to a person should not change significantly as a result of that person’s information being included, or not, in the analysis.”
- In other words:
*“Anything it’s revealed by an output from a database containing some individual's information is almost as likely to have come from a database without that individual's information... and this is true for *any* individual and *any* dataset.*

Differential privacy

...aka plausible deniability

Purpose: Make use of aggregate user information and statistics without compromising the privacy of individuals.



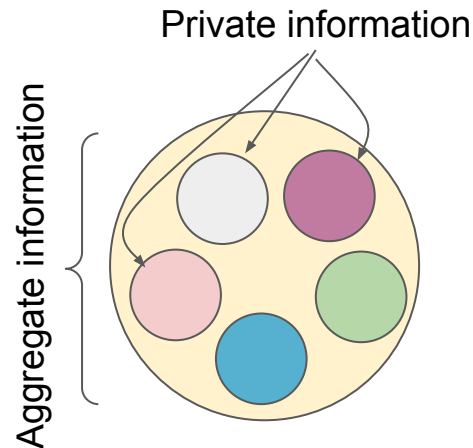
Differential privacy

...aka plausible deniability

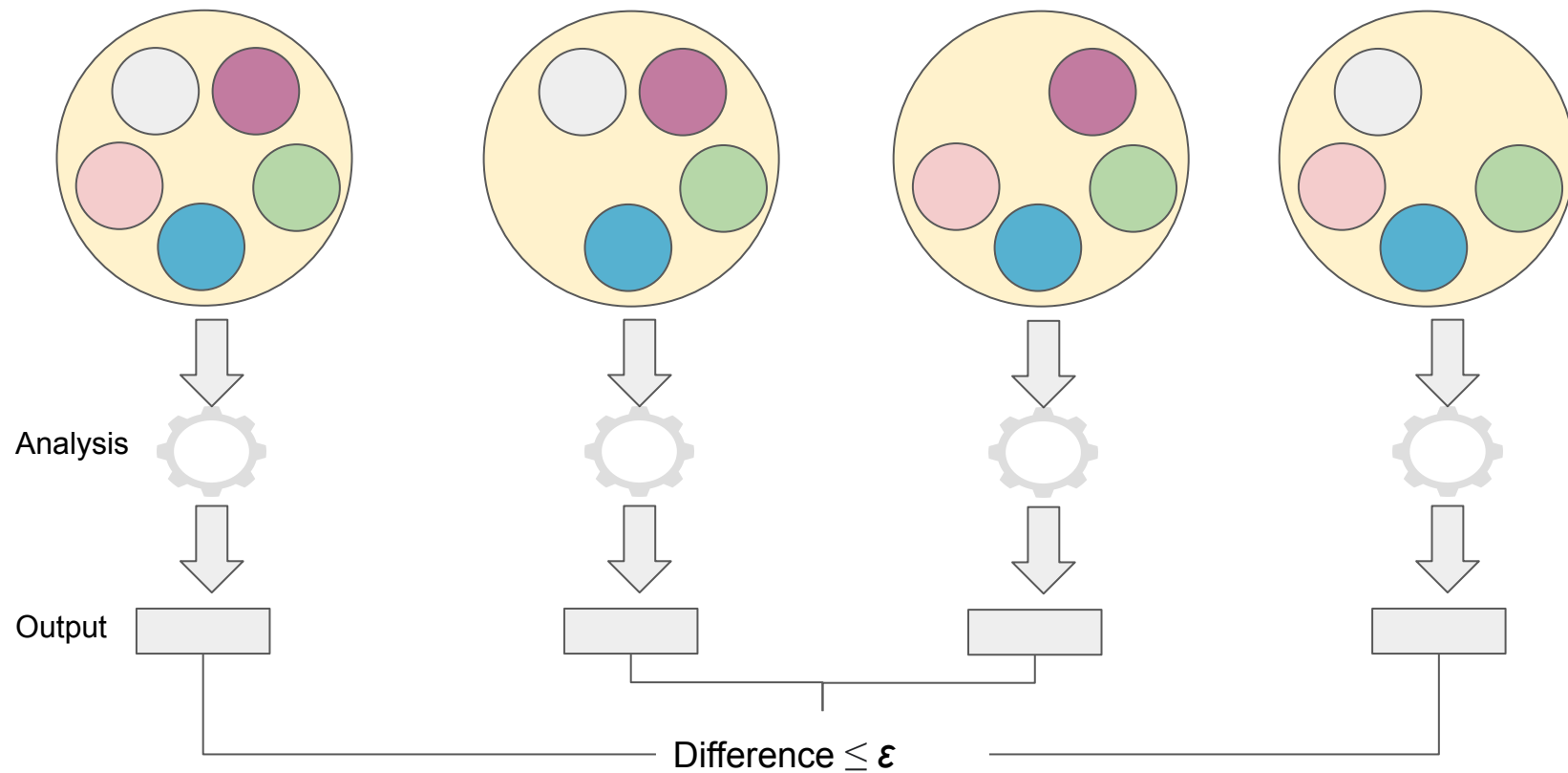
Purpose: Make use of aggregate user information and statistics without compromising the privacy of individuals.

Example:

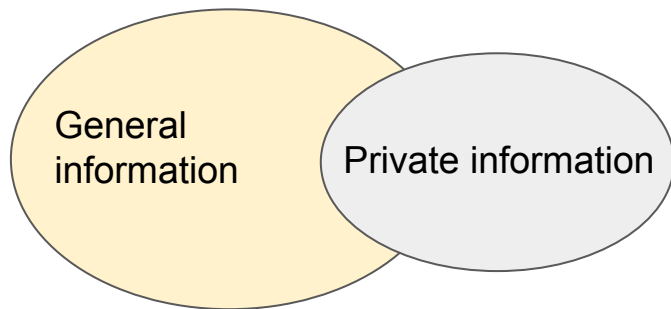
- The health authority conducts a survey about the health status of individuals in Rome.
- It gets its data from local hospitals.
- Due to the sensitive status of the data, the hospitals do not allow direct access to individuals' health records.
- With Differential Privacy, the health authority can still make inferences about the populations' health (on aggregate) without directly accessing the individuals' records.



Aggregate vs private information



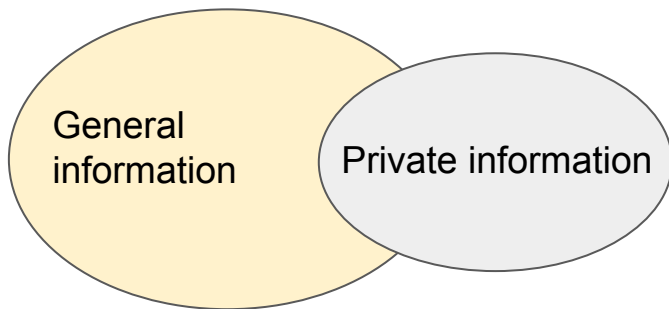
General vs private information



General information: The aggregate information for the population the user belongs to.

Private information concerns only a specific user/individual.

General vs private information



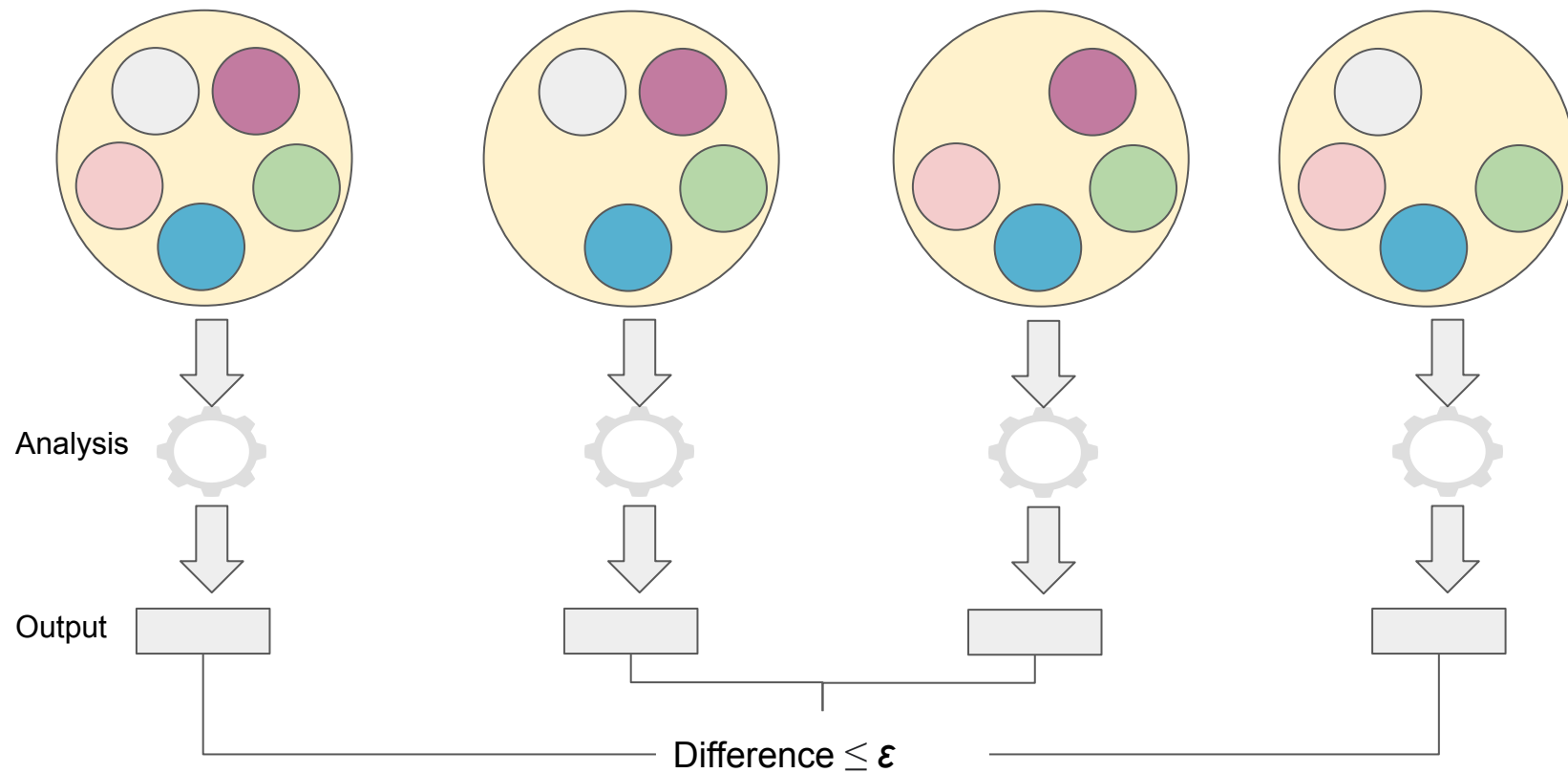
General information: The aggregate information for the population the user belongs to.

Private information concerns only a specific user/individual.

Scenario*

- John, a smoker, participates in a study.
- The study, after analysis, concludes that smoker causes cancer.
- We now have one extra datapoint about John's health: he has higher risk for cancer.
- Was John's information leaked?
 - No: it is the results of the study that revealed the extra (general) information, not the fact that he was in the study.

Aggregate vs private information



Privacy guarantee

- Consider two datasets D and D' which differ by at most a row.
- A randomized mechanism $M: X \rightarrow Y$ operates on the datasets to produce a result.
- Differential privacy tells us that for all D, D' it holds that $M[D] \approx M[D']$

Formally:

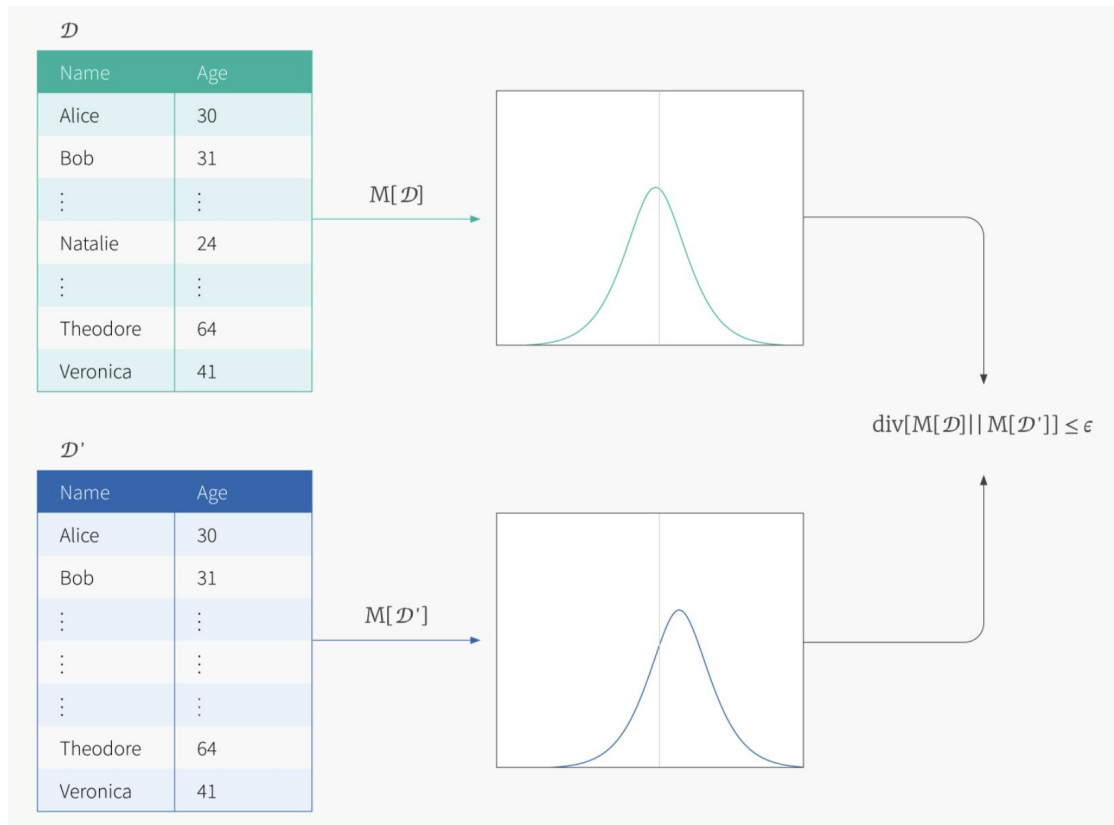
$M: X \rightarrow Y$ is ϵ -differentially private if for all subsets $E \subset Y$ *and datasets* D, D' we have:

$$\Pr(M[D] \in E) \leq \exp[\epsilon] \Pr(M[D'] \in E)$$

Relation to Renyi divergence

$$\text{div}[M[D] \parallel M[D']] \leq \epsilon$$

ϵ quantifies the divergence of the distributions of outcomes from applying the mechanism to similar datasets.



Privacy-utility trade-off

$M: X \rightarrow Y$ is ϵ -differentially private if for all subsets $E \subset Y$ and datasets D, D' we have:

$$\Pr(M[D] \in E) \leq \exp[\epsilon] \Pr(M[D'] \in E)$$

Relation to Renyi divergence: $\text{div}[M[D] \parallel M[D']] \leq \epsilon$

- ϵ tells us how much privacy is leaked when the mechanism is applied on the data.
- $\epsilon = 0$ gives $\Pr(M[D] \in E) = \Pr(M[D'] \in E)$
(Same output independent of data i.e. pure noise)
- Obviously there is a **trade-off** between data utility and privacy preservation.

How to create diff. private data?

Scenario:

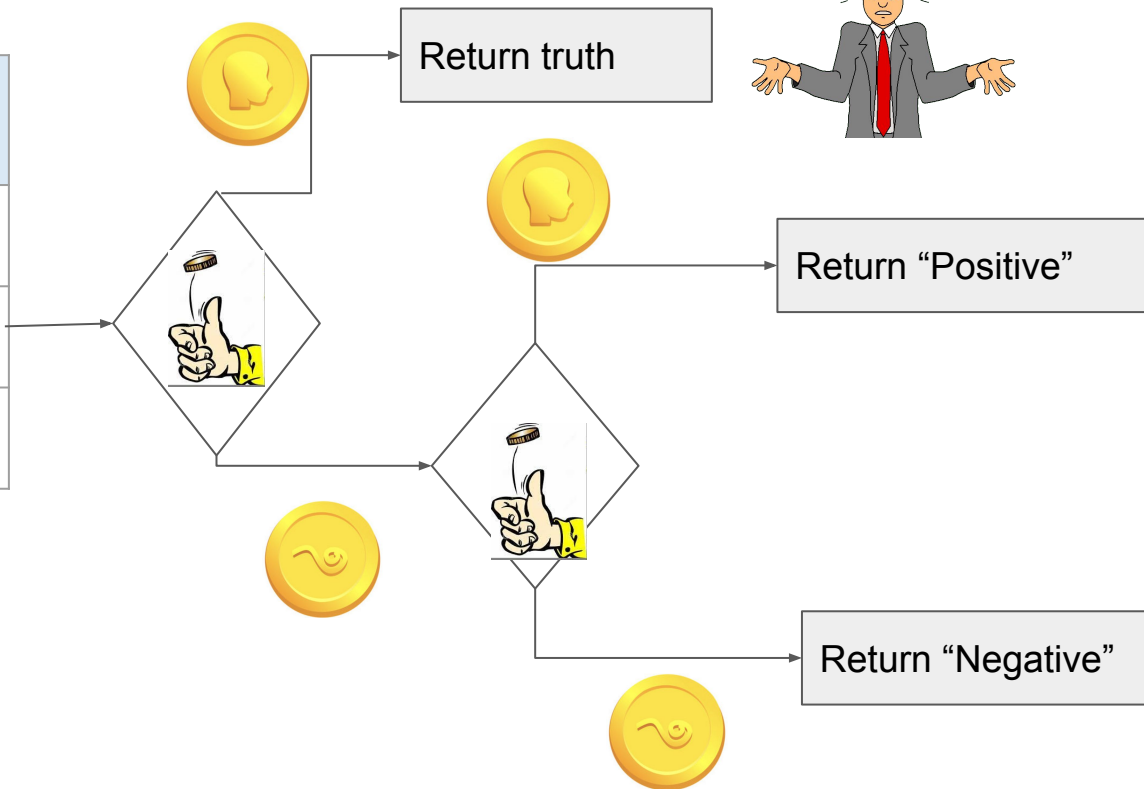
“A group of employees give feedback on their manager. The employees do not feel comfortable if their manager knows they gave bad feedback”.

Can the manager still get valuable aggregate feedback from the employees, while each employee has plausible deniability that there might not be the one to give the negative feedback?

What if all feedbacks come back as negative? Each employee can claim that this is due to injected noise to protect privacy! ... Even if extremely unlikely (for all of the employees), it cannot be ruled out for each individual employee.

Randomized response

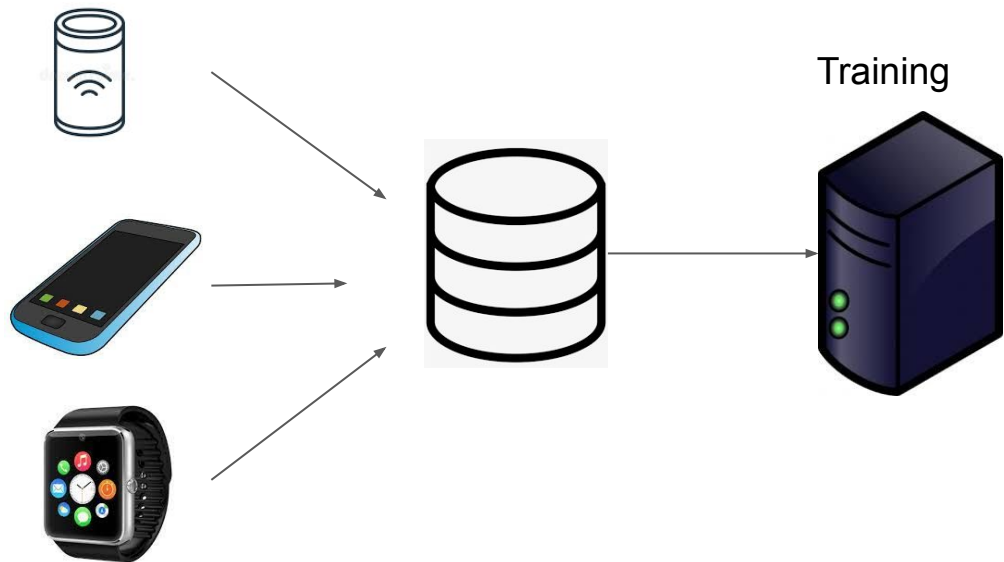
Name	Feedback
Bob	Negative



Must have been the stupid algorithm, boss!

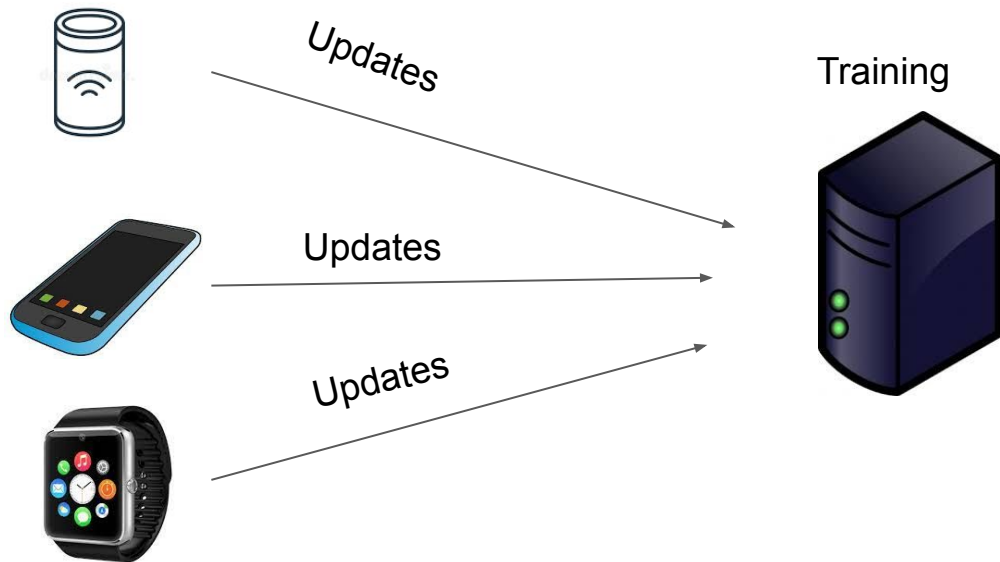
Federated learning

Traditional ML training

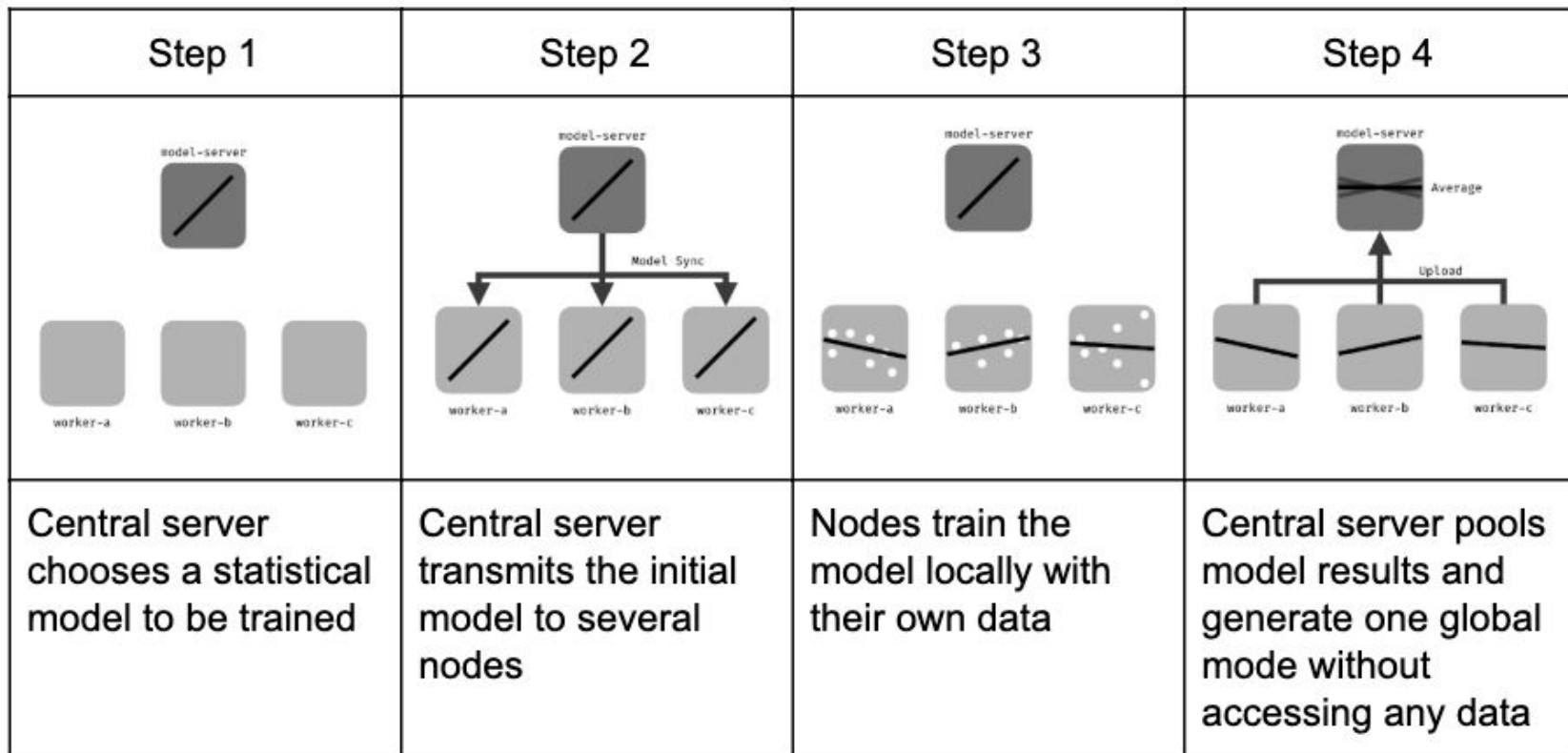


Federated learning - What?

Federated ML training: A decentralized form of ML. Models are trained locally on each device without actual data leaving the device. A central server aggregates local training results, *not data* (and updates a central, privacy preserving model).



Federated learning - What?

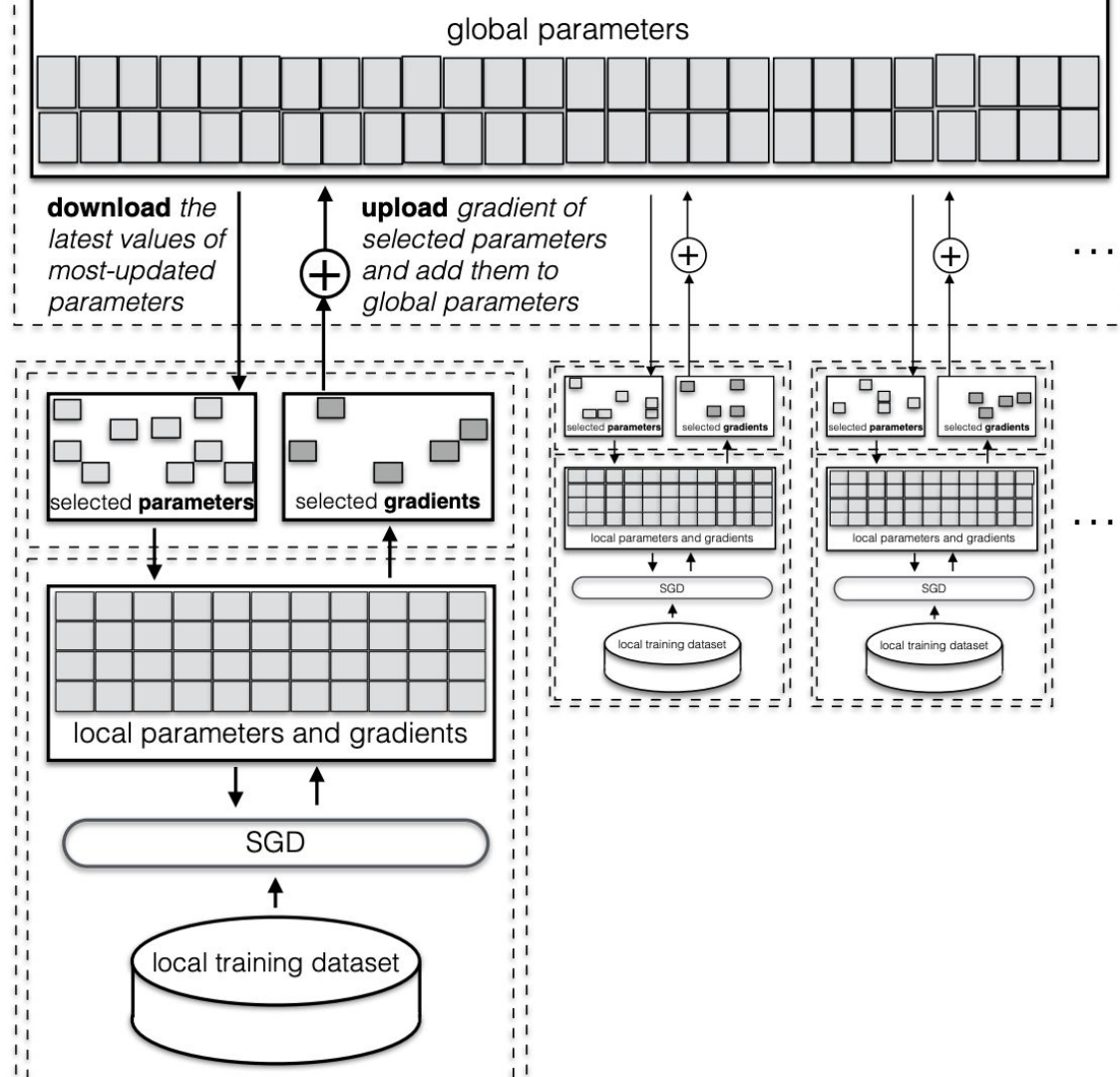


Federated learning - Why?

- Privacy reasons: We don't want to send sensitive user data to a centralized server.

Applications:

- Text Autocomplete
 - Smart assistant wake-up word
 - ...
-
- Results are personalized: since training is (mostly) done locally.
 - Latency is small.
 - ML Service stays light (since processing is done on the devices).



Federated learning - How? FedSGD

α	Learning rate of stochastic gradient descent
θ_d, θ_u	Fraction of parameters selected for <u>d</u> ownload and <u>u</u> pload
γ	Bound on gradient values shared with other participants
τ	Threshold for gradient selection

Choose initial parameters $\mathbf{w}^{(i)}$ and learning rate α .

Repeat until an approximate minimum is obtained:

1. Download $\theta_d \times |\mathbf{w}^{(i)}|$ parameters from server and replace the corresponding local parameters.
2. Run SGD on the local dataset and update the local parameters $\mathbf{w}^{(i)}$ according to (1).
3. Compute gradient vector $\Delta \mathbf{w}^{(i)}$ which is the vector of changes in all local parameters due to SGD.
4. Upload $\Delta \mathbf{w}_S^{(i)}$ to the parameter server, where S is the set of indices of at most $\theta_u \times |\mathbf{w}^{(i)}|$ gradients that are selected according to one of the following criteria:
 - *largest values*: Sort gradients in $\Delta \mathbf{w}^{(i)}$ and upload θ_u fraction of them, starting from the biggest.
 - *random with threshold*: Randomly subsample the gradients whose value is above threshold τ .

The selection criterion is fixed for the entire training.

Procedure for node i

W_t : the global params at step t broadcasted by server

Client	FedSGD	FedAvg
	<p>Compute gradient on local data</p> $g_k = \nabla F_k(w_t)$	<p>Compute gradient on local data</p> $g_k = \nabla F_k(w_t)$ <p>Perform local updates on the parameters (multiple times)</p> $w_{t+1}^k \leftarrow w_t - \eta g_k$
Server	<p>Aggregate gradients and update params</p> $w_{t+1} \leftarrow w_t - \eta \sum_k \frac{n_k}{n} g_k$	<p>Aggregate locally updated params</p> $w_{t+1} \leftarrow \sum_k \frac{n_k}{n} w_{t+1}^k$

Take home message:

- Treating user's and sensitive data privately is important
- Ad-hoc anonymizations might not do the trick (remember linkage attack)
- Diff. privacy is a framework for provable privacy guarantees
- Federated learning is a framework for decentralized ML such that local (e.g. clients') data never leave their devices

Extra: Graph based
data representation

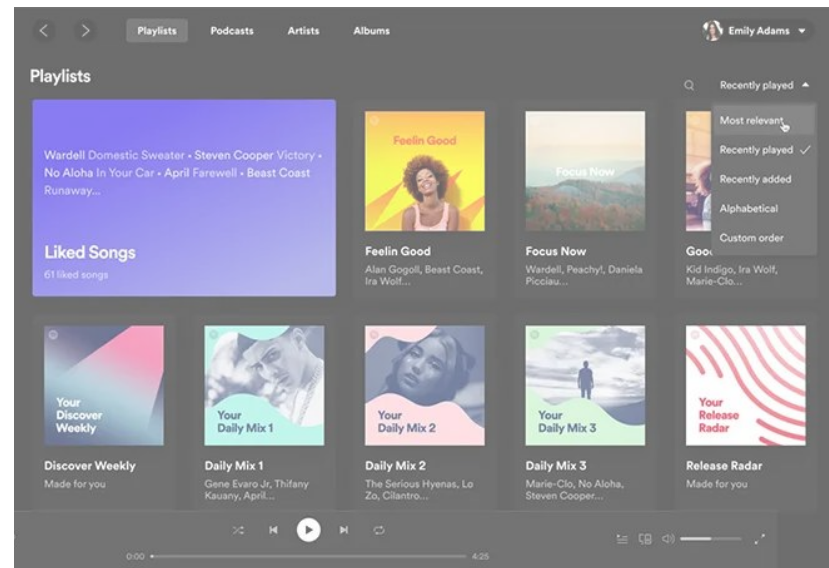
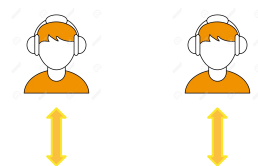
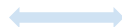
Tabular vs Graphical data representations

Show	Topic	Publisher
spotify:show:bl	Science	Spotify
spotify:show:ab	Education	Spotify
spotify:show:sA	Politics	Spotify

Episode	Entity
spotify:episode:qd	wiki.org/David_A
spotify:episode:9s	wiki.org/Elton_Jo
spotify:episode:gA	wiki.org/David_A

User	Episode
fjsa-xkgw-affs-	spotify:episode:gA
gkad-kd98-ajgs	spotify:episode:9s
gkad-kd98-ajgs	spotify:episode:qd

User	SearchQuery
fjsa-xkgw-affs-	gjsGjdAdghsgUU
gkad-kd98-ajgs	gjAsgSkshx0sa
gkad-kd98-ajgs	gDaobifjq9bmsa



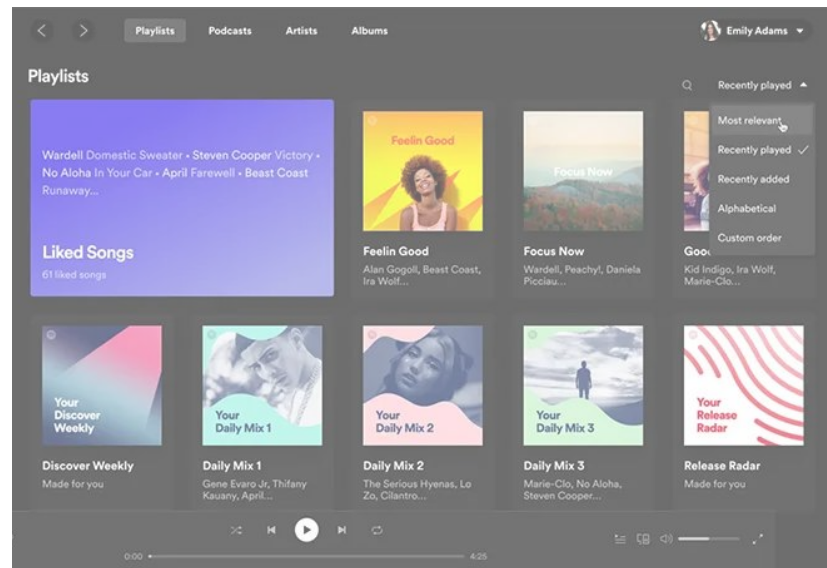
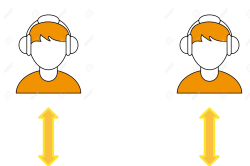
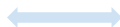
Tabular vs Graphical data representations

Show	Topic	Publisher
spotify:show:bl	Science	Spotify
spotify:show:ab	Education	Spotify
spotify:show:sA	Politics	Spotify

Episode	Entity
spotify:episode:qd	wiki.org/David_A
spotify:episode:9s	wiki.org/Elton_Jo
spotify:episode:gA	wiki.org/David_A

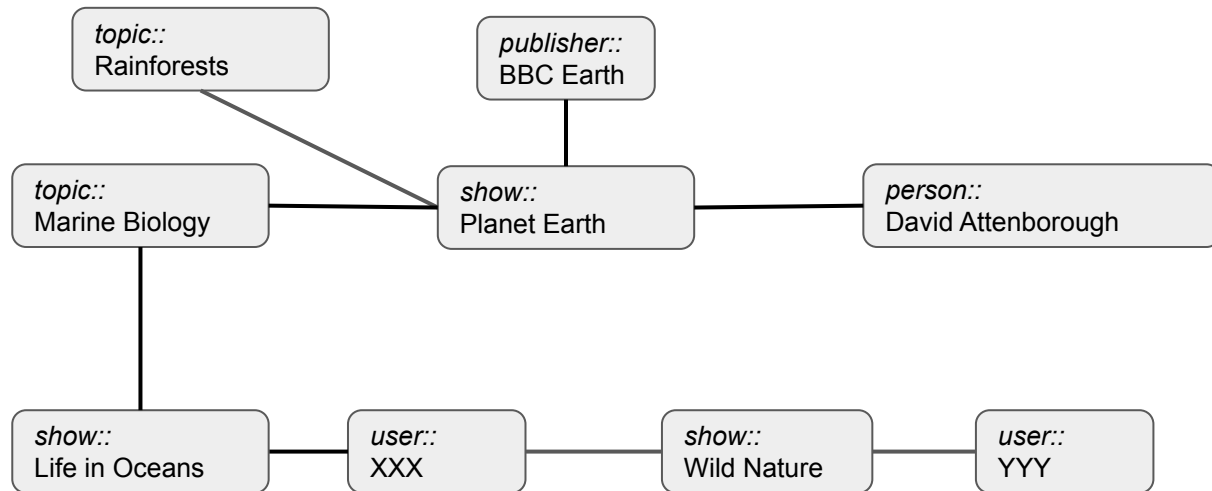
User	Episode
fjsa-xkgw-affs-	spotify:episode:gA
gkad-kd98-ajgs	spotify:episode:9s
gkad-kd98-ajgs	spotify:episode:qd

User	SearchQuery
fjsa-xkgw-affs-	gjsGjdAdghsgUU
gkad-kd98-ajgs	gJAsgSkshx0sa
gkad-kd98-ajgs	gDaobifjq9bmsa



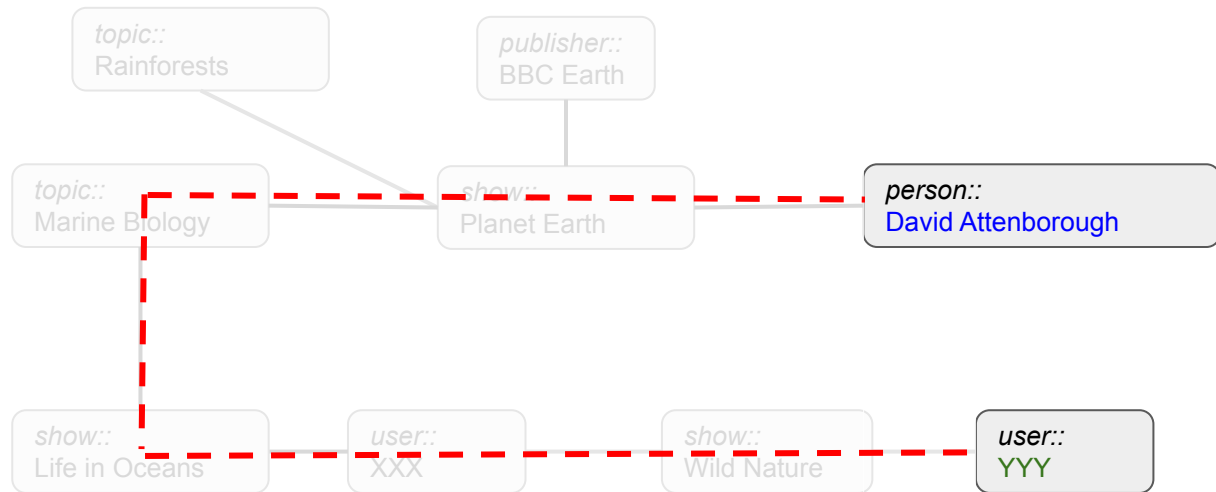
Relationships in Podcast data

Relationships are conventionally represented as graphs.

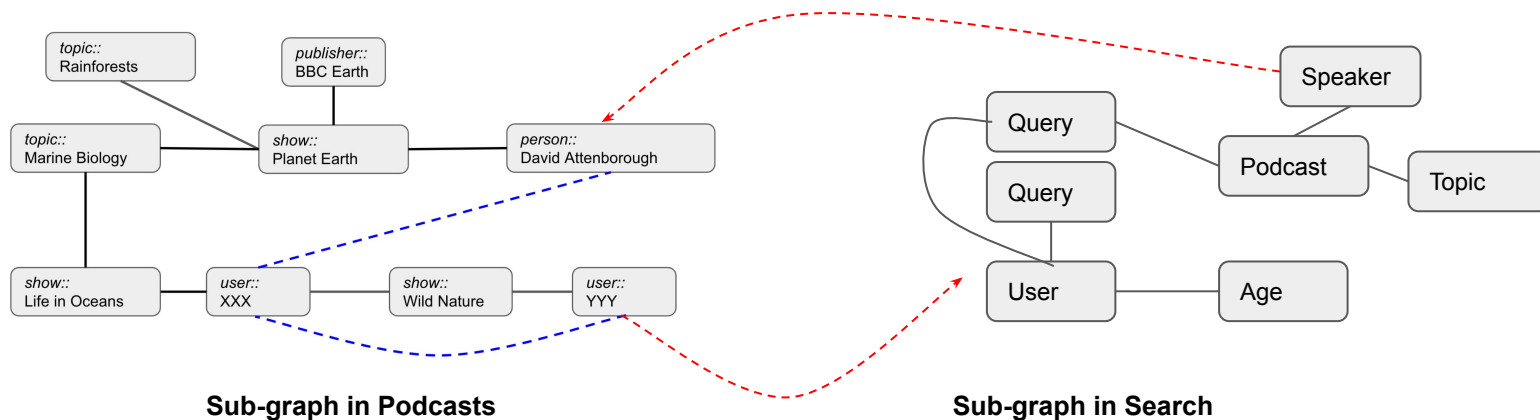


Relationships in Podcast data

For example, we can recommend other episodes featuring **David Attenborough** to **user YYY** even if this user has contributed minimal consumption data.



Graphs across different parts of the platform can be linked

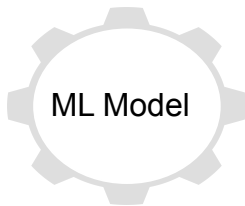
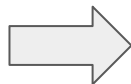


- Bringing together heterogeneous knowledge from different parts of the platform to form a more holistic understanding of users and content.
- Learn more nuanced embeddings capturing complex relationships

Pipeline for ML-driven tasks

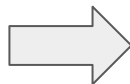
User	Episodes watched
X	A, B
Y	B, C
Z	D

Train



Train an ML model.

Generate



Embeddings

01	10	10
01	10	01
11	11	11
00	11	10
10	10	11
11	10	10

Generate embeddings
distilling our data's info

Embedding similarity



Tasks:
Recommendations,
Quality score, ...

Use embedding similarity in downstream tasks.

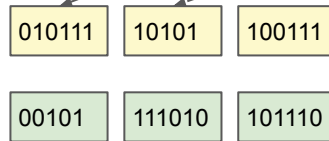
For example, the embedding for episode A is similar to that of episode B, hence we might recommend episode B to users who have watched episode A.

Graph Neural Network embeddings are richer

Traditional learning

User	Episodes watched
X	A, B
Y	B, C
Z	D

*extract
embeddings*



user X embedding

user Y embedding

episode embeddings

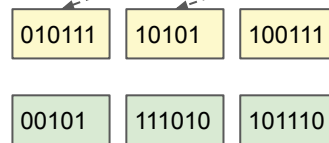
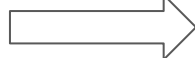
Every user or episode embedding encapsulates information about the corresponding user or episode plus the *simple* relations revealed by paired tabular data.

Graph Neural Network embeddings are richer

Traditional learning

User	Episodes watched
X	A, B
Y	B, C
Z	D

extract
embeddings



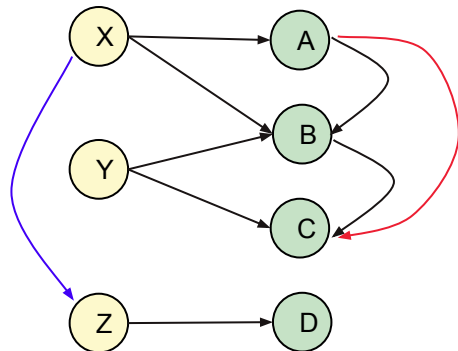
user X embedding

user Y embedding

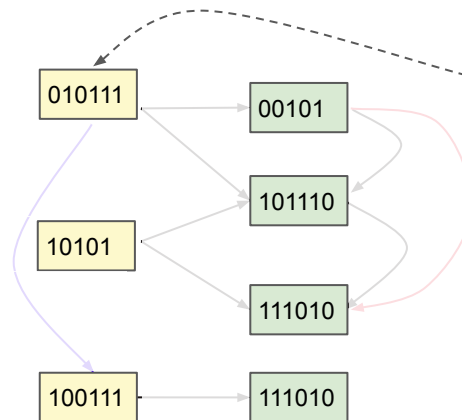
episode embeddings

Every user or episode embedding encapsulates information about the corresponding user or episode plus the *simple* relations revealed by paired tabular data.

Graph learning



extract
embeddings



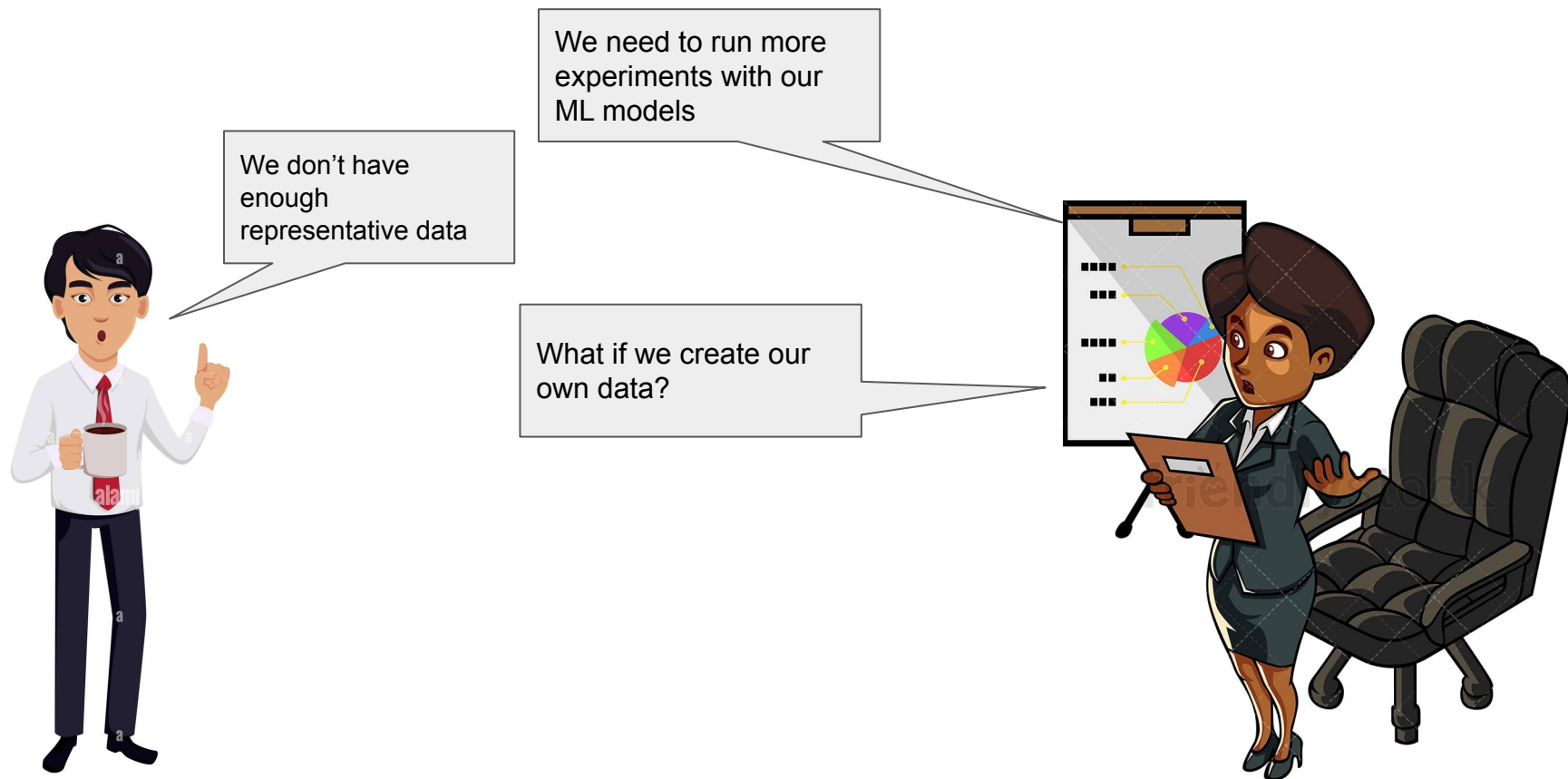
user X *graph node* embedding

Every node embedding encapsulates information about the corresponding user or episode plus its *complex* relations, the relations of its neighbours and so on. They are complex because they are inferred across large areas of the graph.

Since Graph Neural Networks (GNNs) take into account holistic information about our data relations, the corresponding embeddings are also reflecting this more nuanced understanding.

Extra: Data Generation

What if we don't have enough data?



Synthetic data generation

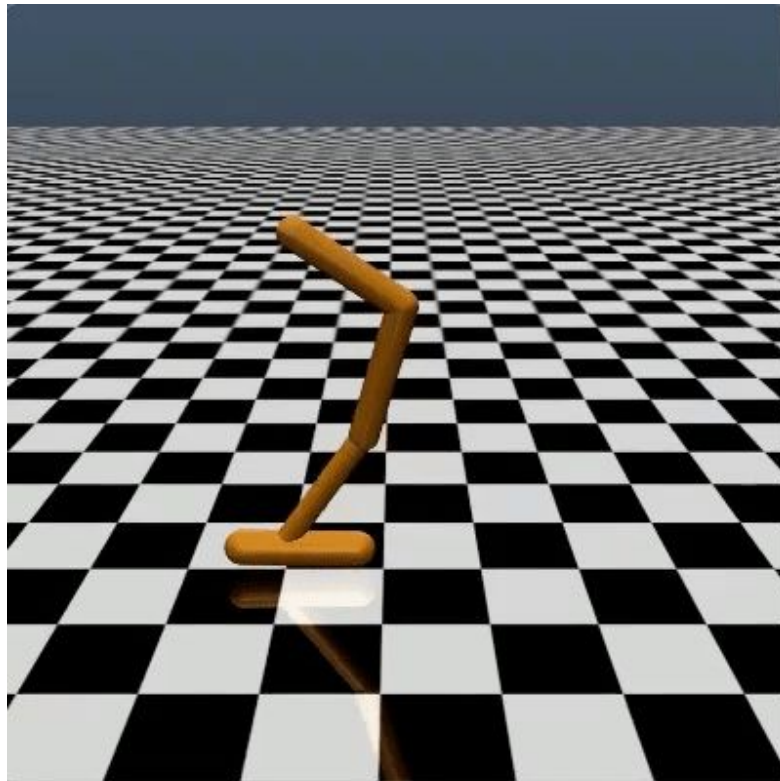
- Generate artificial data that can be used to train or validate ML models (e.g. test fairness of classifier)
- Synthetic data are easier to work with because they can be less biased and fully compliant with privacy/legislative requirements
- Caveat: If the synthetic data are not statistically representative of the real data, we introduce further bias

Example: Self-driving cars

- Loads of positive training data
- Very few negative data (collisions etc)



Simulators



Optional: Hands-on exercises
suggestions

Outline

- Data and model considerations, interplay between data and model
- Types of bias in data and related issues & solutions
- Uncertainty in data/models
- Data tools for industrial scale
- From raw data to ML features
- Time evolution considerations for data in ML (distribution shift, versioning...)
- Privacy and regulation
- Extra: Graph data representations for the industry