UNIVERSITAT DE BARCELONA

MSc in Fundamental Principles of Data Science

3 Ethical Data Science

Fundamental and Practical Limits of ML

Jordi Vitrià

Limits to prediction

ML Aim:

If we model a phenomenon/system as a process by which some **input state** X is transformed into some **output state** Y, we can hope to learn an approximate **transformation function** $y = f(x)$ from **observed** past examples using machine learning/statistics.

Limits to prediction

If we model a phenomenon as a process by which some input state X is transformed into some output state Y, we can hope to learn a **transformation function** $y = f(x)$ from **observed** past examples using machine learning/statistics.

Method: We observe $P(X, Y)$ (i.i.d data) and model $\mathbb{E}(Y \mid X)$ by maximizing the empirical risk of the model (accuracy, likelihood).

The interpretation of $\mathbb{E}(Y \mid X)$ is: "**given that I have observed X, what can I say about Y?**"

Limits to prediction

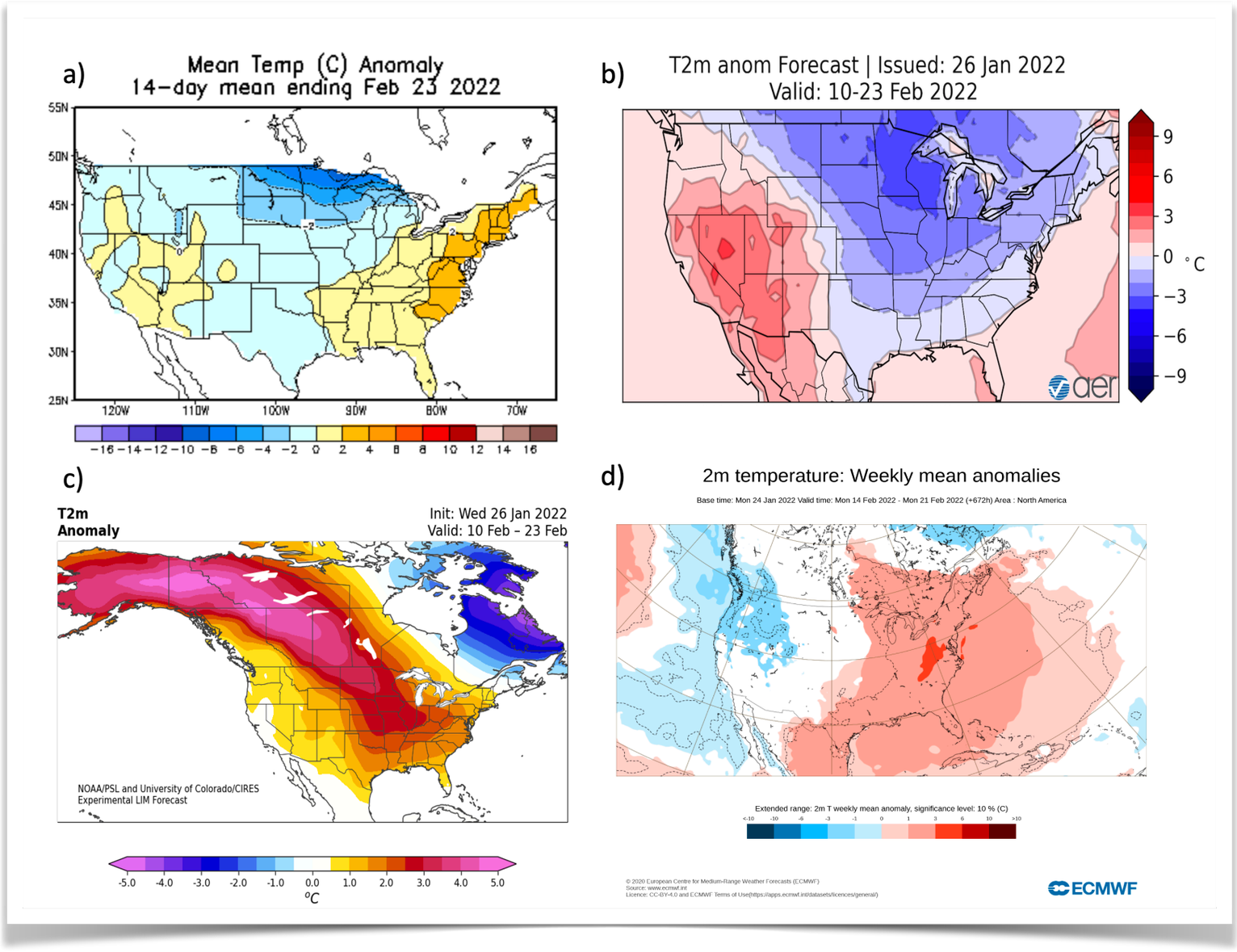
The term "i.i.d data" refers to "**Independent and Identically Distributed**" data.

In the context of statistics and machine learning, i.i.d is an assumption about the random variables that make up the dataset being used for analysis or modeling. Here's a breakdown of what this means:

Independent: Each data point (or random variable) in the dataset is assumed to be independent of the others. This means that the occurrence of one data point does not influence or change the probability of occurrence of another data point. For example, in a coin toss, each toss is independent of the previous ones.

Identically Distributed: This means that each data point in the dataset is drawn from the same probability distribution and has the same statistical properties (such as mean, variance, etc.). It does not mean that all data points are the same, but rather that they share the same underlying distribution.

Limits to prediction



A **seven-day** forecast can accurately predict the weather about **80 percent** of the time and a **five-day** forecast can accurately predict the weather approximately **90 percent** of the time.

However, a **10-day**—or longer—forecast is only right about half the time.

Limits to prediction

Weather data is typically not i.i.d because it often violates both the **independence** and **identical distribution** assumptions for several reasons:

- **Temporal Dependence**: Weather observations are strongly dependent on preceding conditions. For example, the weather today is likely to be similar to the weather yesterday to some extent, especially in terms of temperature, precipitation, and atmospheric pressure. This sequential dependence means weather data points are not independent.

- **Seasonal Variations**: Weather data exhibits seasonal patterns, which means that its distribution can change over different times of the year. For instance, temperatures are generally higher in summer and lower in winter in many places. This seasonal effect means that the distribution of weather data is not identical throughout the year.

Limits to prediction

Getting truly i.i.d weather data is challenging due to the inherent temporal and spatial dependencies in weather phenomena. However, you can approximate i.i.d conditions in weather data for specific types of analyses:

- If your analysis can tolerate it, randomly sampling weather data from a wide range of locations and times might reduce dependencies.
- Instead of using raw weather data, you can use anomalies or deviations from a long-term mean. For example, calculating the deviation of daily temperatures from the 30-year average for that day can help to remove some of the seasonal and longer-term trends, making the data more homogenous.
- Etc.

Limits to prediction

Let's suppose we get iid data.

Is everything predictable <u>given enough data and powerful algorithms</u>?

- Are there fundamental limits?
- Which are the <u>practical</u> limits?

Fundamental Limits of ML

Fundamental limits to prediction

- The nondeterminism of phenomena of interest
- Inscrutability of of the world
  impossible to know, to understand
- Computational limits.
- Limits to collecting sufficient training examples (volume, independence, etc.)
- Etc.

If there were a vast intelligence — Laplace's Demon — that knew the exact state of the universe at any one moment, and knew all the laws of physics, and had arbitrarily large computational capacity, it could both predict the future and reconstruct the past with perfect accuracy.

Fundamental limits to prediction

Note that:

The laws of physics (Core Theory, QFT) are sufficient to predict the future state of the universe (at least the part of the universe that matters for humans) at any one moment given a complete representation of the current state.

Carroll, Sean M. "The Quantum Field Theory on Which the Everyday World Supervenes." *arXiv preprint arXiv:2101.07884* (2021).

- Determinism of the universe at the most fundamental level is compatible with non-determinism at higher levels of description (chemistry, biology, psychology, sociology, etc.)!
- The cause of this paradox is that higher levels of description are defined by states that correspond to multiple fundamental level states.

Practical Limits of ML

Practical limits to prediction

Practical limits:

- Sensitive dependence on inputs (butterfly's effect, **ill-posed problems**). This is possible even in linear models.
- Effects of unexpected/unpredictable events (a lottery jackpot; an accident). This corresponds to variables that interact with very low probability (the real problem of autonomous driving).
- Feedback loops (predicting Y causes changes in X).
- Drift: the statistical relationship between the input variables and the target may change over time.
- Unobservable/latent input features (intelligence, people's thoughts).

All these issues can cause **failures..**

When shouldn't be used prediction?

Sometimes, what is incorrectly framed as a **prediction** problem can be better understood as a problem of **explanation**, **intervention**, or **decision making**.

- **Explanation** is about generating scientific insight into how a process works rather than simply predicting its input-output behavior. We need a generative model of $P(X, Y)$, their statistical relationships are not sufficient (**causality**).

- **Intervention** is about figuring out how to change a process for the better rather than treating it as a given and confining oneself to making predictions. We need a generative model of $P(X, Y)$ (**causality**).

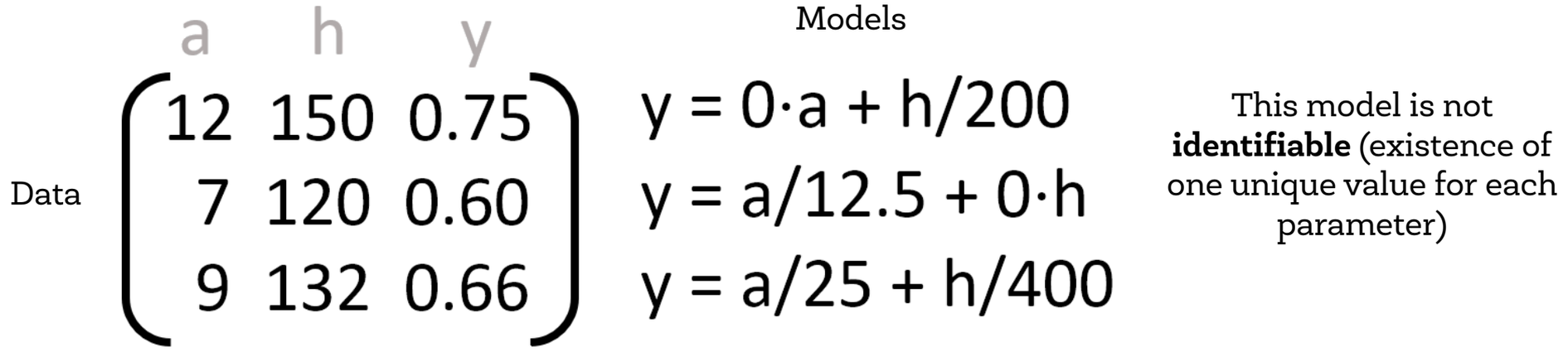
- **Decision making** recognizes that many considerations go into making good decisions **beyond maximizing predictive accuracy** (fairness, diversity, etc.).

When shouldn't be used prediction?

Explanation is about generating scientific insight into how a process works rather than simply predicting its input-output behavior.

Example: the multicollinearity problem.

Take the fictional toy example of predicting a child's reading ability (y) as a function of its age (a) and height (h). Let's assume age and height are perfectly correlated in our data, as in the example below. Now we can express y equivalently as:



This model is not **identifiable** (existence of one unique value for each parameter)

When shouldn't be used prediction?

Intervention is about figuring out how to change a process for the better rather than treating it as a given and confining oneself to making predictions.

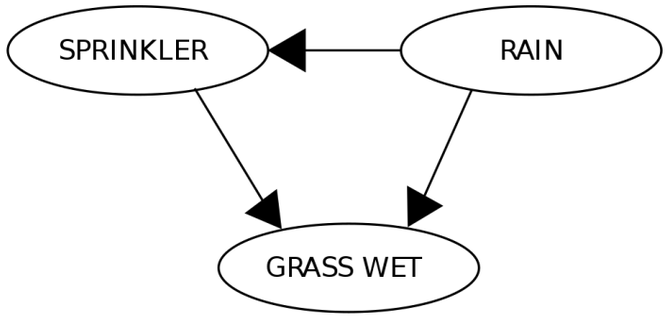
Data

$P(Wet, Rain, Sprinkler)$

Sprinkler	Rain	Wet
T	F	T
T	T	T
T	T	F
F	F	F
T	F	T
F	T	F
F	T	T
T	F	T
...	...	...

Generative Model

RAIN	SPRINKLER T	SPRINKLER F
F	0.4	0.6
T	0.01	0.99



RAIN	T	F
	0.2	0.8

SPRINKLER	RAIN	GRASS WET T	GRASS WET F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Observing $P(Wet, Rain, Sprinkler)$ does not determine the effect of an intervention $P(Rain \mid do(Wet))$. In general $P(Rain \mid do(Wet)) \neq P(Rain \mid Wet)$.

When shouldn't be used prediction?

Decision making recognizes that many considerations go into making good decisions **beyond maximizing predictive accuracy**, especially because the decisions themselves have causal effects.

When training a model:

- I want to minimize the Empirical Risk.
- **I want to maximize robustness against changes in data distribution.**
- **I want to be able of explaining my predictions.**
- **I want to measure and mitigate unwanted biases (discrimination).**
- **Etc.**

ML failures from a data-centric point of view

ML failures

ML fails when we are dealing with a predictive problem but, at inference time, $\mathbb{E}(Y|X)$ does not correspond to what happens in the real world.

ML failures

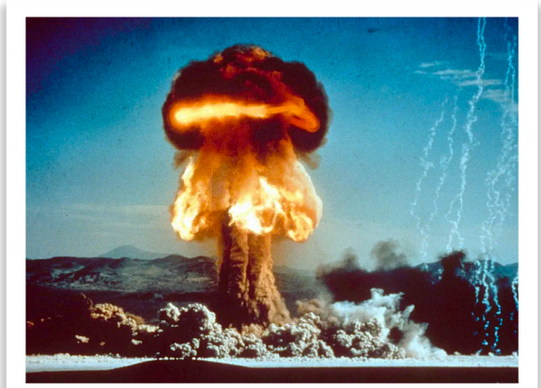
- **Data distribution shifts**: the model learns from a distribution that does not represent the world at inference time.

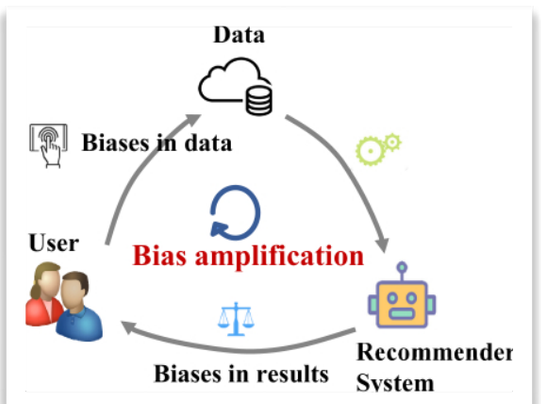
ML failures

- **Data distribution shifts**: the model learns from a distribution that does not represent the world at inference time.

Causes:

- External **changes in the data generation process**.
- **Degenerate feedback loops**: system's outputs cause changes in the inputs.





ML failures

- **Edge Cases**: a ML learning can fail in a number of edge cases, making catastrophic mistakes.

GIZMOS

The Cyclist Problem

Self-driving cars aren't good at detecting cyclists. The latest proposed fix is a cop-out.

BY CHRISTINA BONNINGTON

FEB 03, 2018 • 8:07 AM

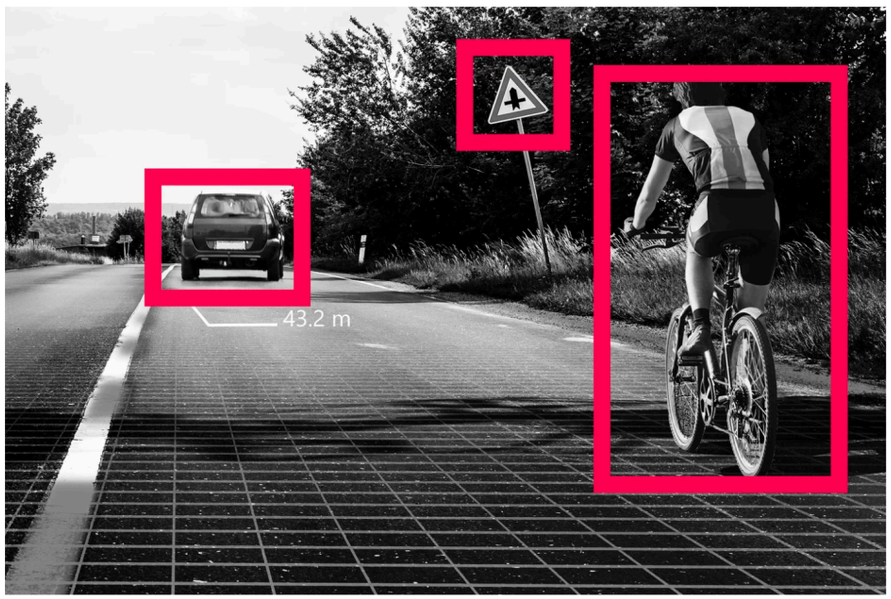


Photo illustration by Natalie Matthews-Ramo/Slate. Photos by Thinkstock.

Data Distribution Shifts

- The distribution of the data the model is trained on, $P(X, Y)$, is called **source distribution**.
- The distribution of the data the model runs inference on is called the **target distribution**.
- $P(X, Y)$ can be decomposed in two ways:
  - $P(X, Y) = P(X)P(Y|X)$
  - $P(X, Y) = P(Y)P(X|Y)$

Data Distribution Shifts

Data distribution shifts are:

- **Covariate shift** is when $P(X)$ changes, but $P(Y \mid X)$ remains the same.
- **Label Shift** is when $P(Y)$ changes, but $P(X \mid Y)$ remains the same.
- **Concep drift** is when $P(Y \mid X)$ changes, but $P(X)$ remains the same.

Covariate Shift

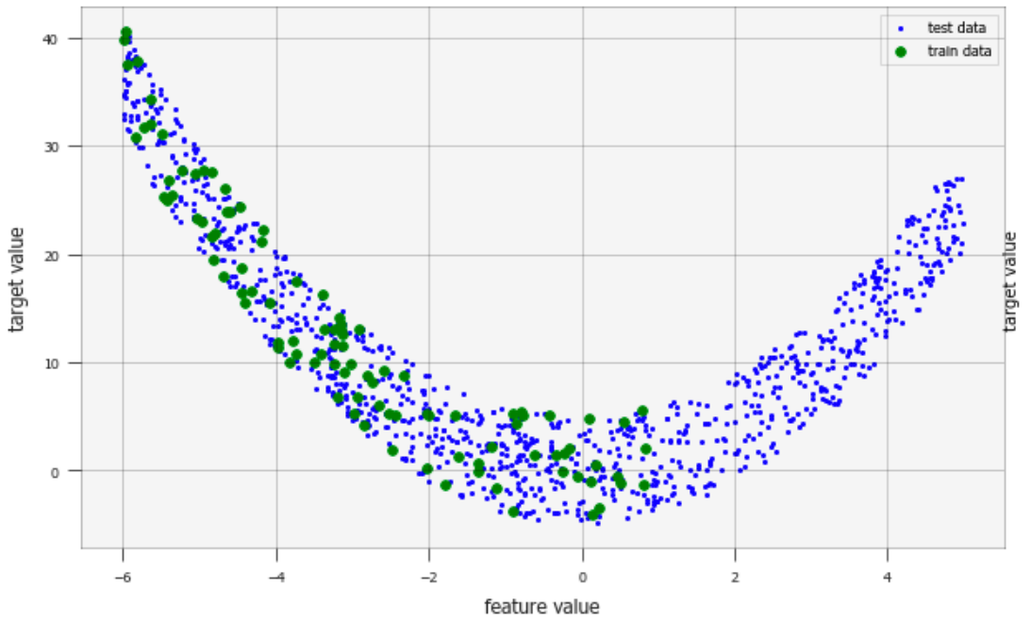
Statistics: a covariate is a variable that can influence the outcome of a given statistical trial.

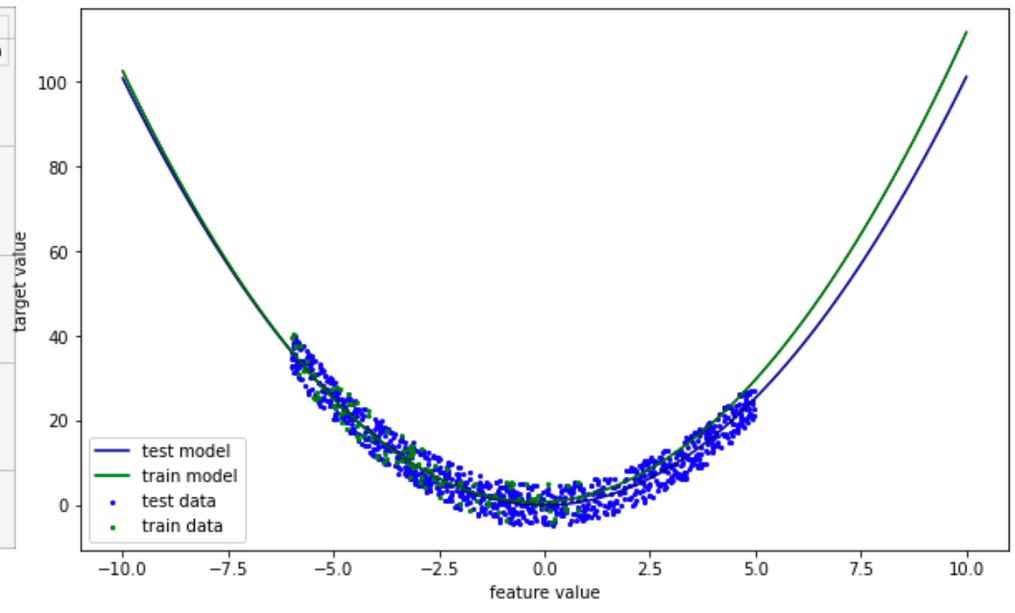
Supervised ML: input features are covariates.

Covariate shift: **Input distribution changes, but for a given input, output is the same.**

Covariate Shift

$P(X)$ changes, but for a given input, $P(Y \mid X)$ is the same.





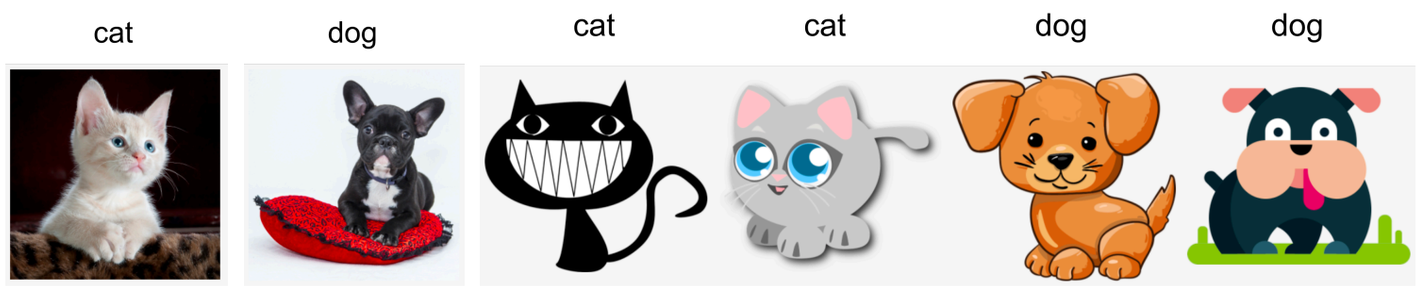
New incoming data can invalidate the current model.

Covariate Shift

$P(X)$ changes, but for a given input, $P(Y \mid X)$ is the same.

Training

Test



Covariate Shift

$P(X)$ changes, but for a given input, $P(Y \mid X)$ is the same.

Example:

- Predicts P(cancer | patient_data)
- $P_{training}$(age > 40) > $P_{inference}$(age > 40)
- $P_{training}$(cancer | age > 40) = $P_{inference}$(cancer | age > 40)

There are several causes. E.g. women >40 are encouraged by doctors to get check-ups.

Covariate Shift

$P(X)$ changes, but for a given input, $P(Y \mid X)$ is the same.

Example:

- Predicts P(cancer | patient_data)
- $P_{training}$(age > 40) > $P_{inference}$(age > 40)
- $P_{training}$(cancer | age > 40) = $P_{inference}$(cancer | age > 40)

Training: If knowing in advance how the production data will differ from training data, use **importance weighting**.

Production: unlikely to know how a distribution will change in advance.

Importace weighting

In supervised machine learning, it is important to train an estimator on balanced data so the model is equally informed on all classes.

To balance the classes, we can inform the estimator to adjust how it calculates loss. Using weights, we can force as estimator to learn based on more or less importance ('weight') given to a particular class.

Weights scale the loss function. As the model trains on each point, the error will be multiplied by the weight of the point. The estimator will try to minimize error on the more heavily weighted classes, because they will have a greater effect on error, sending a stronger signal. Without weights set, the model treats each point as equally important.

Example: Logistic regression

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(-(y_i \log(\hat{y}_i)) + (1 - y_i)\log(1 - \hat{y}_i))$$

$$WeightedLoss = \frac{1}{N}\sum_{i=1}^{N}(-w_0(y_i \log(\hat{y}_i)) + w_1(1 - y_i)\log(1 - \hat{y}_i))$$

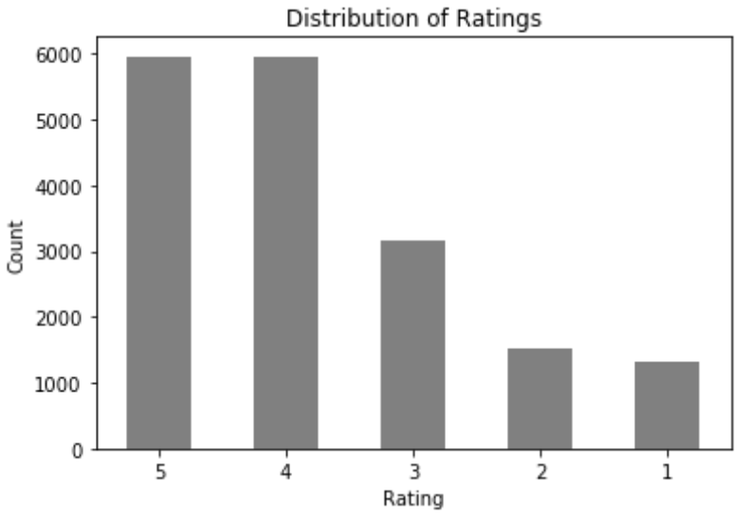
Importace weighting

In supervised machine learning, it is important to train an estimator on balanced data so the model is equally informed on all classes.

To balance the classes, we can inform the estimator to adjust how it calculates loss. Using weights, we can force as estimator to learn based on more or less importance ('weight') given to a particular class.

Weights scale the loss function. As the model trains on each point, the error will be multiplied by the weight of the point. The estimator will try to minimize error on the more heavily weighted classes, because they will have a greater effect on error, sending a stronger signal. Without weights set, the model treats each point as equally important.

Example: Multiclass

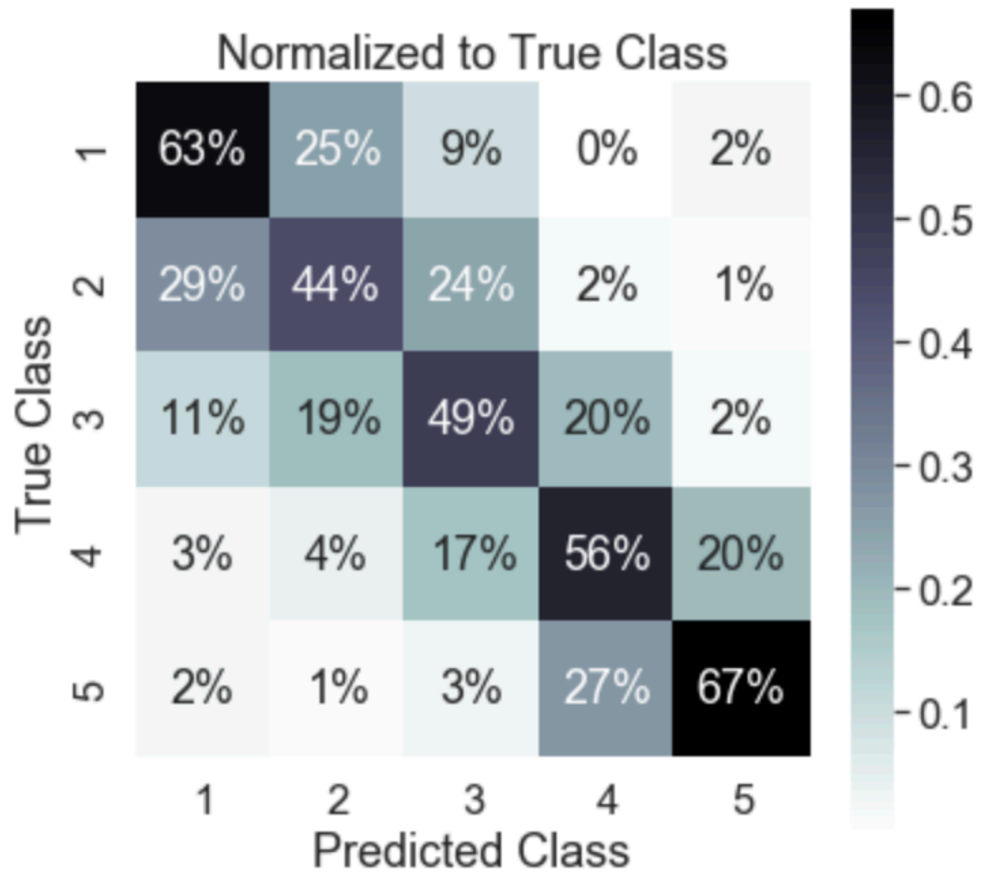



```
Class Distribution (%)


1     7.431961
2     8.695045
3    17.529658
4    33.091417
5    33.251919
```

```
Class Weights: 5 classes


{1: 2.691079812206573,
 2: 2.3001605136436596,
 3: 1.140923566878981,
 4: 0.6043863348797975,
 5: 0.6014690451206716}
```

Importace weighting

Non-weighted sample data, strongly favors majority classes



Weighted sample data, better train on minority classes

Label Shift

$P(Y)$ changes, but for a given output, $P(X \mid Y)$ is the same.

Output distribution changes but for a *given output*, input distribution stays the same.

Label Shift

$P(Y)$ changes, but <u>for a given output</u>, $P(X \mid Y)$ is the same.

Output distribution changes but for *a given output*, input distribution stays the same.

Exemple:

- Predicts $P(Y = disease \mid X = symptoms)$
- The prevalence of diseases, $P(Y)$, are changing over time.

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)}$$

Concept Drift

$P(X)$ remains the same, but $P(Y \mid X)$ changes.

Same input, expecting different output.

Concept Drift

$P(X)$ remains the same, but $P(Y \mid X)$ changes.

Same input, expecting different output.

Example (non stationary distribution):

- Predicts P(€|features of a house in BCN)
- P(features of a house in BCN) remains the same.
- Covid causes people to leave BCN, housing prices drop.
- P(€1M | features of a house in BCN):
  - Pre-covid: high
  - During-covid: low

Other drifts: Bergson's paradox

US Universities pick students based on a number of attributes.

Two commonly considered attributes are high school GPA and SAT scores.

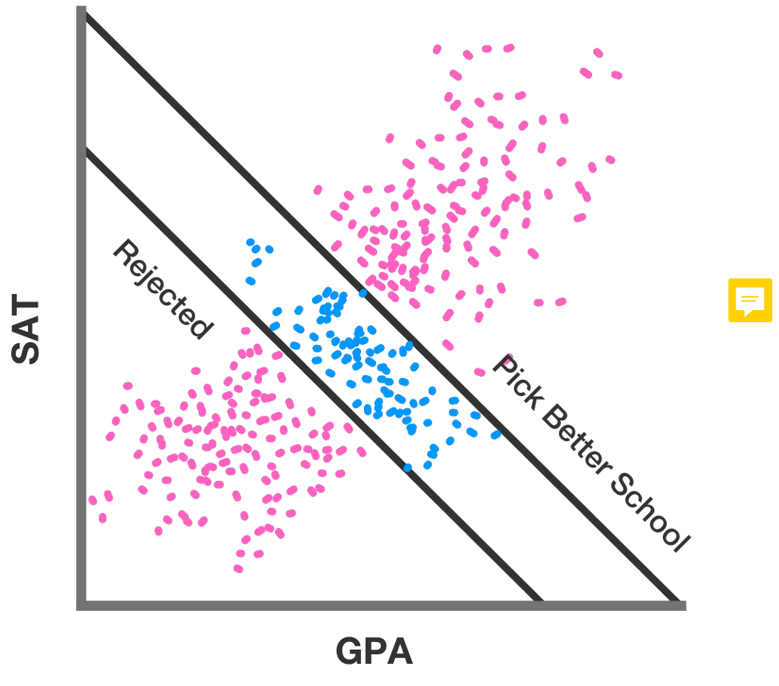
The SAT is a standardized test widely used for college admissions in the United States.

We want to measure the correlation GPA-SAT by using data from a random school. The prior hypothesis is that there is a positive correlation...

Other: Bergson's paradox

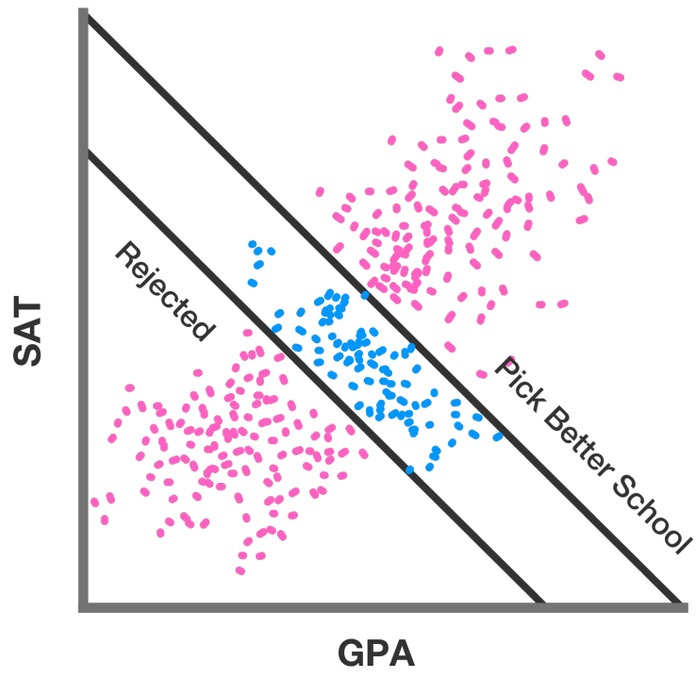
The admissions committee **accepts** students who have either a sufficiently high GPA, a sufficiently high SAT score, or some combination of the two.

However, applicants who have both high GPAs and high SAT scores will likely get into a higher-tier school and **not attend**, even if they are accepted.



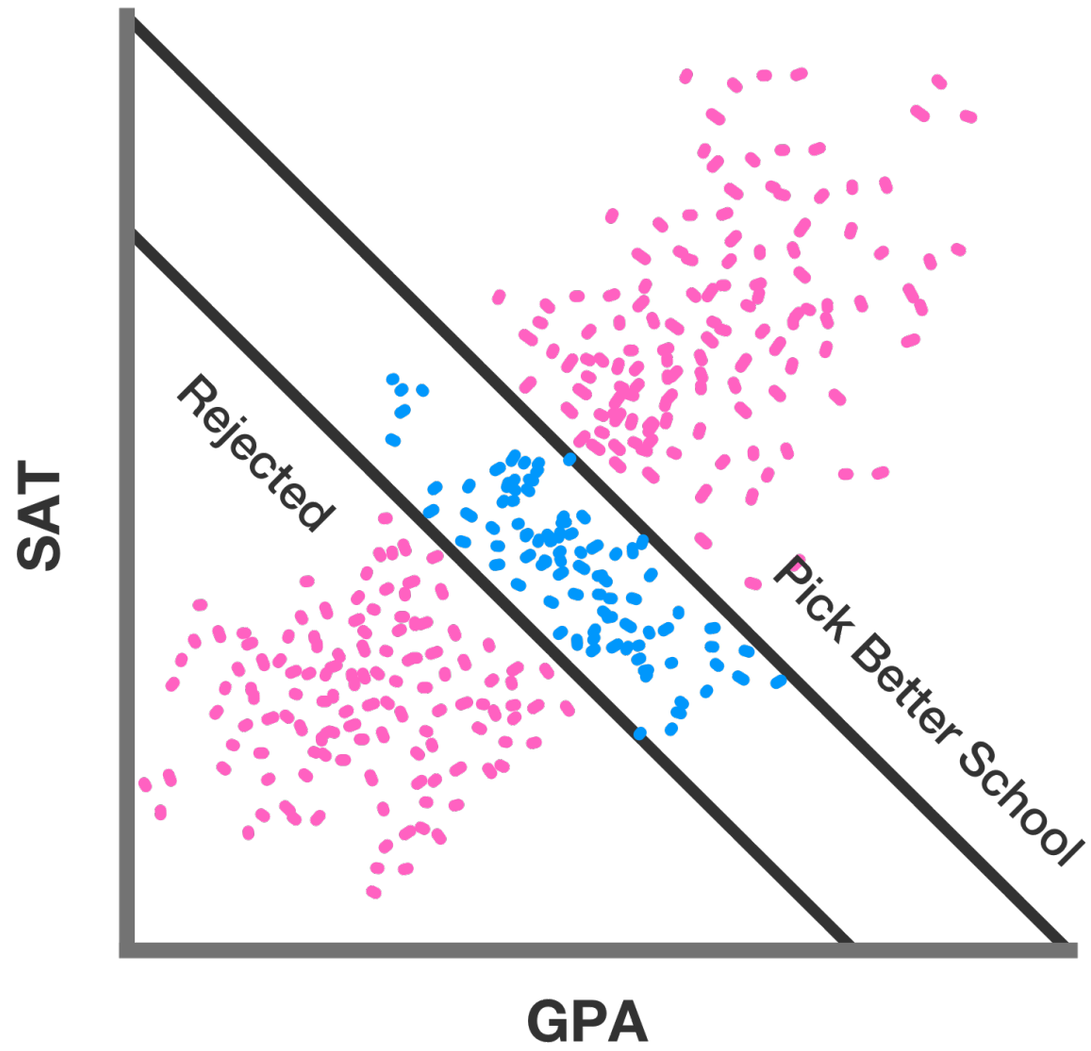
Other: Bergson's paradox

Data show a downward trend (negative correlation) even though the overall population (red and blue dots) show an upward trend (positive correlation). This trend reversal is the "paradox," though there is nothing truly paradoxical about it.



Other: Bergson's paradox

$P(X), P(Y), P(Y|X), P(X|Y)$ change!



How to detect Data Distribution Shifts?

Data distribution shifts are only a problem if they cause your model's performance to degrade.

You have to monitor your model's accuracy related metrics!

How to detect Data Distribution Shifts?

How to determine that two distributions are different?

1. **Compare statistics**: mean, median, variance, quantiles, skewness, kurtosis,...
   How: Compute mean & variance of a feature during training and compare them to the same values computed in production.
   Not universal: only useful for distributions where these statistics are meaningful.
   Inconclusive: if statistics differ, distributions differ. If statistics are the same, distributions can still differ.
2. **Two-sample hypothesis test**.
   How: Determine whether the difference between two populations is statistically significant (using the Kolmogorov–Smirnov test).
   Doesn't make assumptions about distribution.
   Only works with one-dimensional data.

How to address Data Distribution Shifts?

1. **Train model using a <u>massive</u> dataset** (hopefully including diverse data distributions).
2. **Retrain model with new data from new distribution (**fine-tuning**).** Need to figure out not just when to retrain models, but also how and what data.