The Big Objective Here

Every metric will have both capabilities and limitations; no single metric will capture all possible definitions of utility.

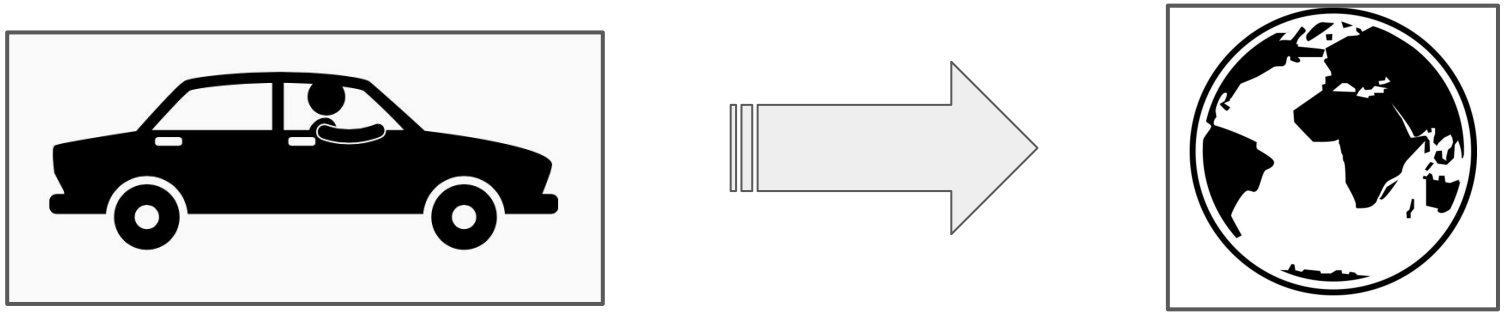
The overall objective of this challenge is to collect metrics that:

(1) Capture real world use cases and data stakeholder needs

(2) Are ***well defined***, and clearly written so that they are straightforward to implement correctly.

(3) Are ***well understood***, with analysis that explores both capabilities and limitations--blindspots, instability, biases, comparability properties….

Tips & Tricks: Defensive Driving for Metric Developers



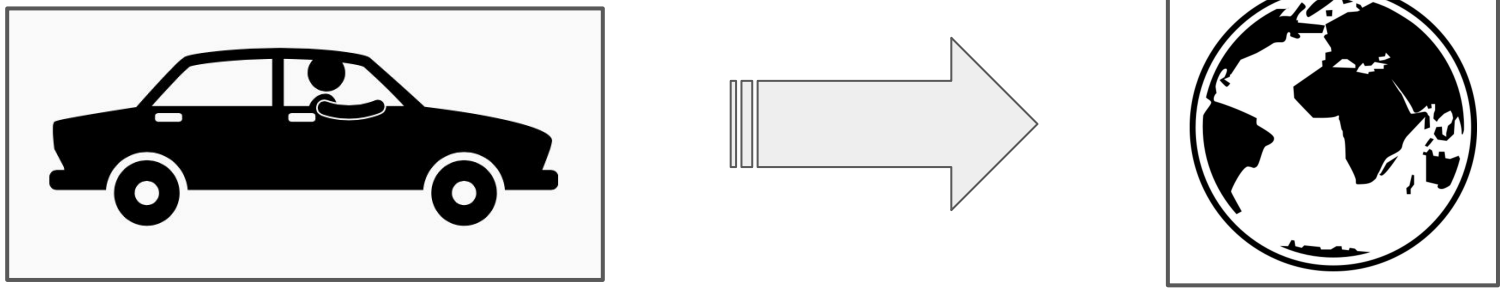
This is for real. We are really, really going to use these, and (if you consent) we are really, really going to put them in front of lots of other important people in the privacy research community so that they can use them too. You get a chance to do this write-up, we let you use all the color and pictures and words and pages and everything else you might want to use to get the word out about your idea, and then once you're done….

Your idea is going into other people's hands. It'll be passed around, pointed out over beers at conferences, mentioned briefly in undergrad lectures, cited in papers…. **and at some point it's going to get misused**.

How do you make sure your metric survives intact in the grapevine of a rapidly changing, rapidly growing, bleeding edge R&D field? By trying to find and clearly identify all the potential pitfalls yourself, and include them with the metric's definition so that people using your metric understand not just *how to implement it*, but also *how it works* and *where it doesn't*.

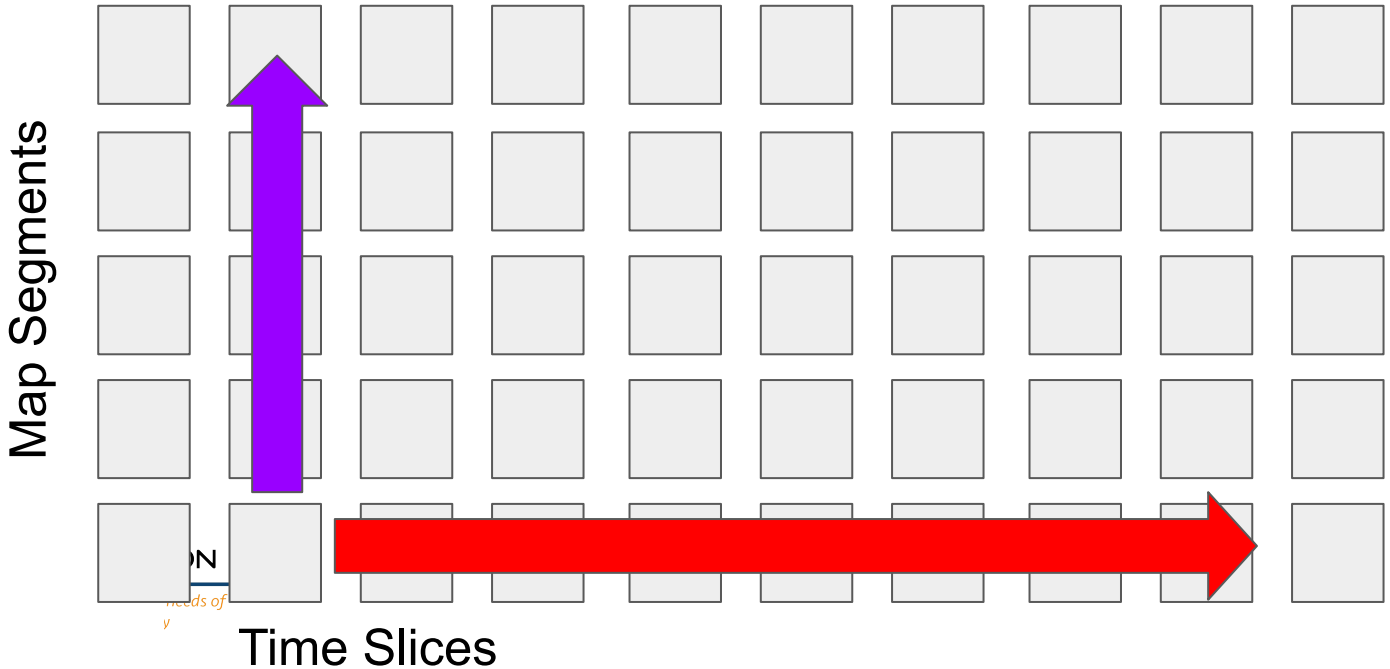
So here's some tips and things to keep in mind for how to do this.

Tips & Tricks: Time vs. Space



Evaluation Space:

Aggregation of Event Types by Time Slice and Map Segment

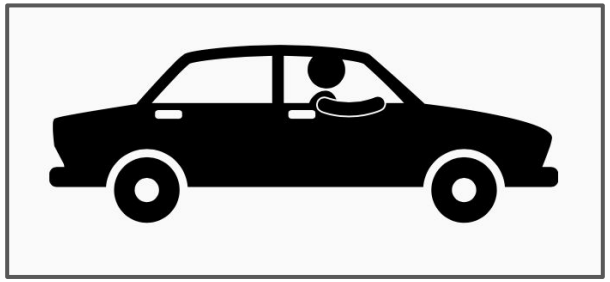


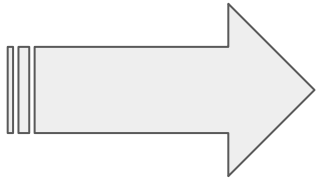
Remember the fun part of this challenge-- **Adventures in Space Time!**

How is your metric handling the difference between space and time? There will be geographic correlations in the data and temporal ones, and we want to make sure that all are preserved in the privatized data.

What part of this problem are you tackling? Are you focused only on map segments, and simply averaging across time? Or are you looking at trends through time and only averaging across map segments? Or are you handling both together?

Tips & Tricks: Ordinal vs Categorical





Data features come in two basic types:

Ordinals that have a natural order to them like numbers, dollar amounts, ages, poverty percentages, times, years, and even highest grade of education.

Categoricals that have no natural ordering: sex, race, language, ancestry, favorite websites, event code, map segment (with caveats).

How does your metric use these two types of variables? Does it only work with one type or the other? (← that's fine). As always, be clear.

Actual Data Table

Age (Number)	Gender (M/F)	Income (Number)	Attended University (T/F)
23	M	$73K	F
32	F	$65K	T
45	M	$84K	T
68	F	$112K	T
54	F	$91K	F

Synthetic Data Table

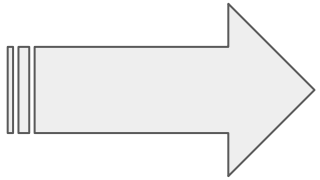
Age (Number)	Gender (M/F)	Income (Number)	Attende University (T/F)
23	M	$73K	F
32	F	$65K	T
45	M	$84K	T
68	F	$112K	T
54	F	$91K	F

Three Marginals Output from Step 1: *Actual and Synthetic Person Data Sources*

Gender (M/F)	Income (Number)	Attended University (T/F)	Actual Count	Synthetic Count
M	$0-33K	F		
F	$0-33K	F		
M	$0-33K	T		
F	$0-33K	T		
M	$34-66K	F		

3-marginal metric from the NIST Differential Privacy Synthetic Data Challenge
Uses binning to treat numerical variables like categorical variables.

Tips & Tricks: Ordinal vs Categorical *Error*



NOTE!

Ordinals have a natural definition of error, how far apart two values are, (A - B). .

Categoricals don't necessarily. You can look at things like edit distance, counts of the number of records with each value (as in pie chart and marginal-based techniques), or using them as ***class values in classification techniques***.

Understanding clearly how your metric operates on these two feature types is important.

Actual Data Table

Age (Number)	Gender (M/F)	Income (Number)	Attended University (T/F)
23	M	$73K	F
32	F	$65K	T
45	M	$84K	T
68	F	$112K	T
54	F	$91K	F

Synthetic Data Table

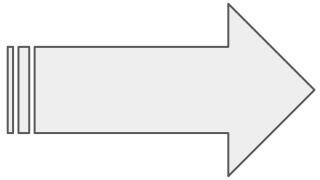
Age (Number)	Gender (M/F)	Income (Number)	Attende University (T/F)
23	M	$73K	F
32	F	$65K	T
45	M	$84K	T
68	F	$112K	T
54	F	$91K	F

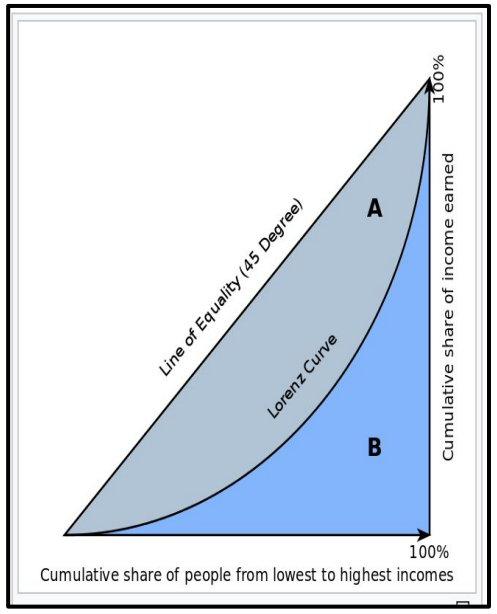
Three Marginals Output from Step 1: *Actual and Synthetic Person Data Sources*

Gender (M/F)	Income (Number)	Attended University (T/F)	Actual Count	Synthetic Count
M	$0-33K	F		
F	$0-33K	F		
M	$0-33K	T		
F	$0-33K	T		
M	$34-66K	F		

3-marginal metric from the NIST Differential Privacy Synthetic Data Challenge
Uses binning to treat numerical variables like categorical variables.

Tips & Tricks: Generalization and Configuration





https://en.wikipedia.org/wiki/Gini_coefficient

Income Inequality metric from the
NIST Differential Privacy Synthetic Data Challenge

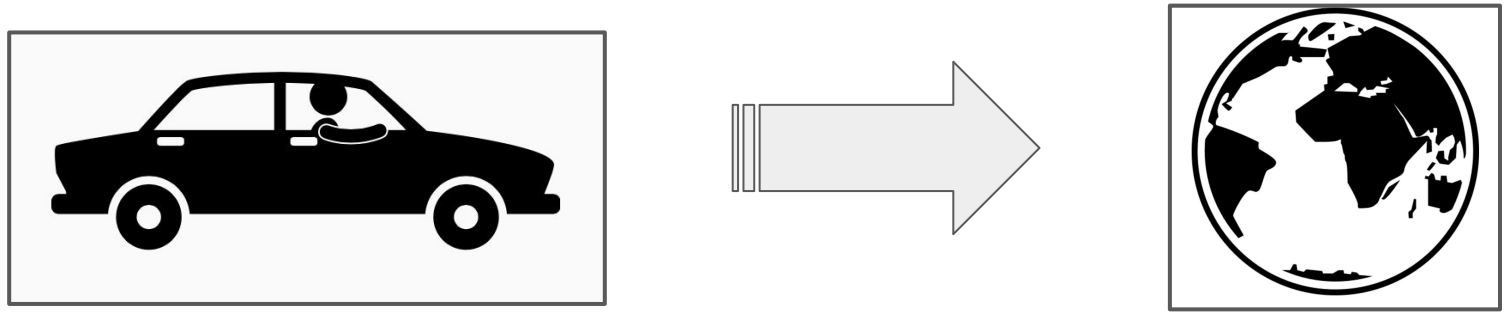
Do you have a great, specific real world use case in mind, such an income inequality, the pay gap, or anti-gerrymanding analytics… but it's highly dependent on the schema containing a specific set of features?

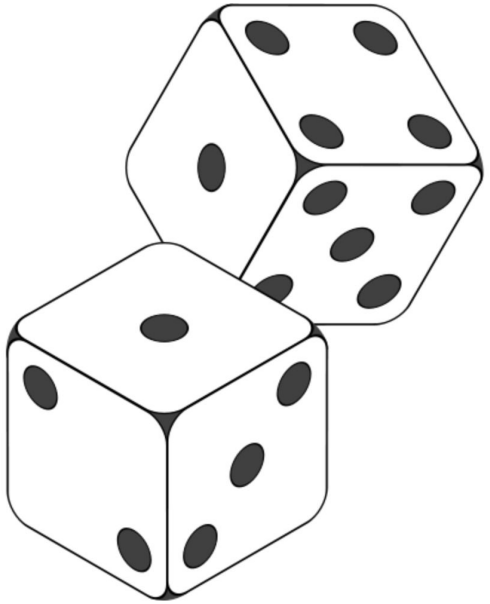
Consider generalizing it! If a use case generally runs on income, can it be run on any financial variable? Or even any numerical variable?

If a use case generally runs on sex or race, can it also be run on any demographic variable?

Merics that can be configured to run on many different schema can provide more comprehensive analysis and much better coverage.

Tips & Tricks: Randomization





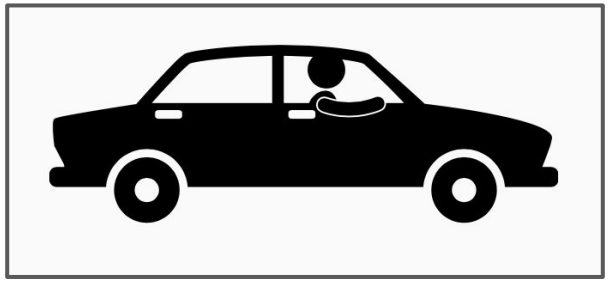
If you have a great idea for a comprehensive metric to evaluate the data, but it takes to long to run, and tends to choke and die if there's too many features or too many records--

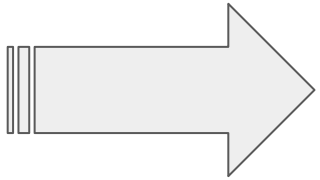
Consider Randomization!

By randomly subsampling features or records, you can create a metric that gets a rapid high-level snapshot of the whole data set quality without exhaustively checking every possible combination.

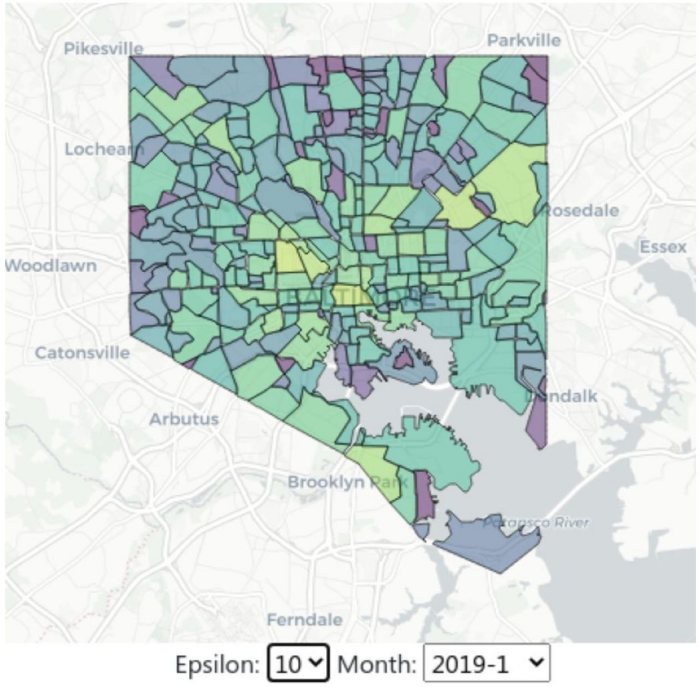
Be careful to explore sampling ratios and stability, though! (more later)

Tips & Tricks: Snapshot and Deep Dive





The **Interactive Map** allows you to see your scores geographically (across all map segments). Here we see that dense urban neighborhoods closer to the city center, which generally contain more records, have better scores than rural and suburban neighborhoods where records may be more sparse. These are challenges that will need to be creatively overcome to achieve good performance on the Sprint 1 task.



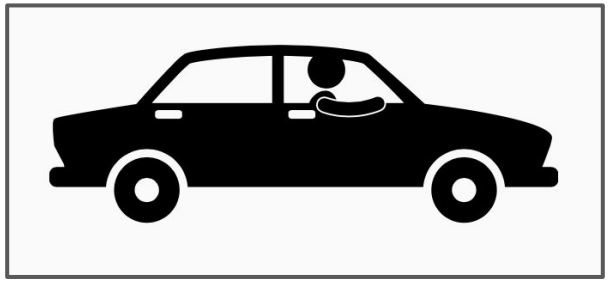
Snapshot vs Deep Dive!

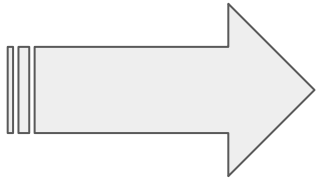
Often metrics can be designed to either give a single total data quality score for a privatized temporal map (Snapshot Mode), or to investigate and pinpoint sources of disparity between the privatized and ground truth data (Deep Dive Mode).

How does your metric produce its single score?

Can you unroll your aggregation or refocus your metric to give more detailed information about specific points of failure?

Tips & Tricks: Snapshot and Deep Dive





The **Temporal Scores Chart** allows you to select a given neighborhood and see the change in your pie chart scores in that neighborhood over each of the time segments. Here we see the scores are relatively uniform across months for our baseline privacy algorithm. However, a privacy algorithm that leverages the temporal aspect of the problem, for example by aggregating counts across multiple time segments, might see more interesting variation here.

Remington (213)

year	month	score
2019	1	0.6073
2019	2	0.6631
2019	3	0.6635
2019	4	0.6849
2019	5	0.6235
2019	6	0.6508
2019	7	0.6536
2019	8	0.5685
2019	9	0.5944
2019	10	0.6263
2019	11	0.6558
2019	12	0.6480

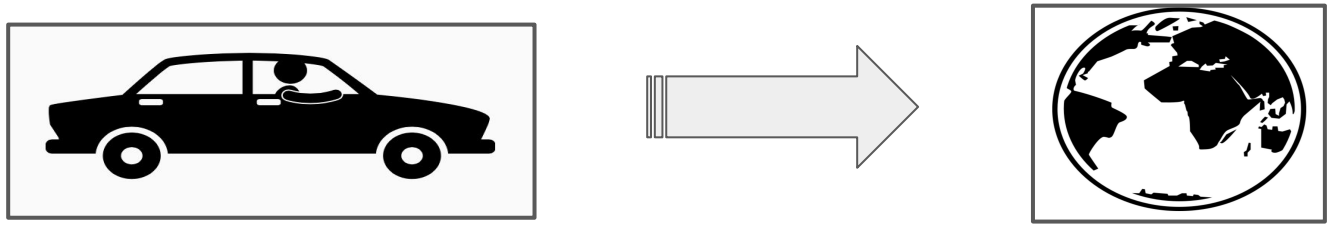
Snapshot vs Deep Dive!

Often metrics can be designed to either give a single total data quality score for a privatized temporal map (Snapshot Mode), or to investigate and pinpoint sources of disparity between the privatized and ground truth data (Deep Dive Mode).

How does your metric produce its single score?

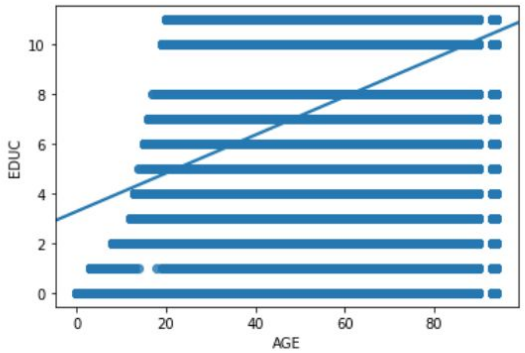
Can you unroll your aggregation or refocus your metric to give more detailed information about specific points of failure?

Tips & Tricks: Checking Blindspots (and Decision Boundaries)



Age vs. Education

Ground Truth

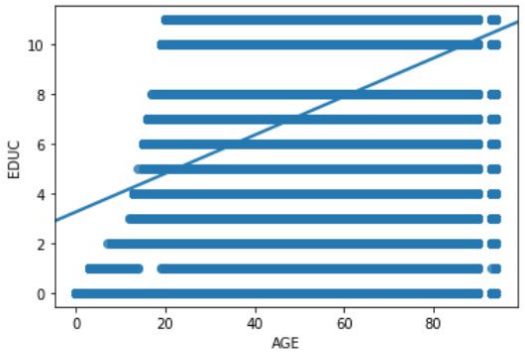


Coefficients:
[[0.07710821]]

Intercept:
[3.27240961]

R-squared:
0.27983908685879266

Good

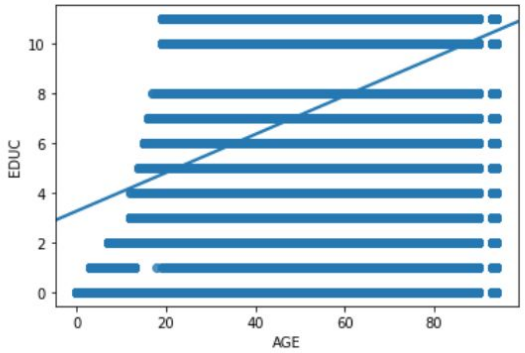


Coefficients:
[[0.07751626]]

Intercept:
[3.25282563]

R-squared:
0.2825870270758932

Mediocre

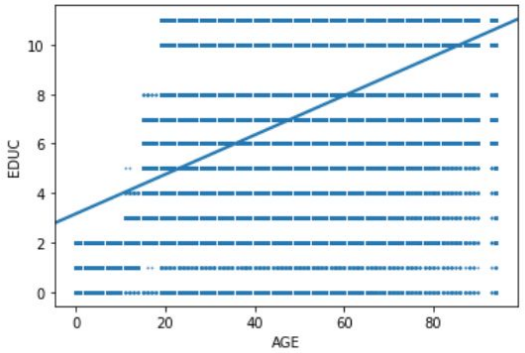


Coefficients:
[[0.07741602]]

Intercept:
[3.25825086]

R-squared:
0.2820458675394747

Poor



Coefficients:
[[0.07959727]]

Intercept:
[3.16305652]

R-squared:
0.29869253706847976

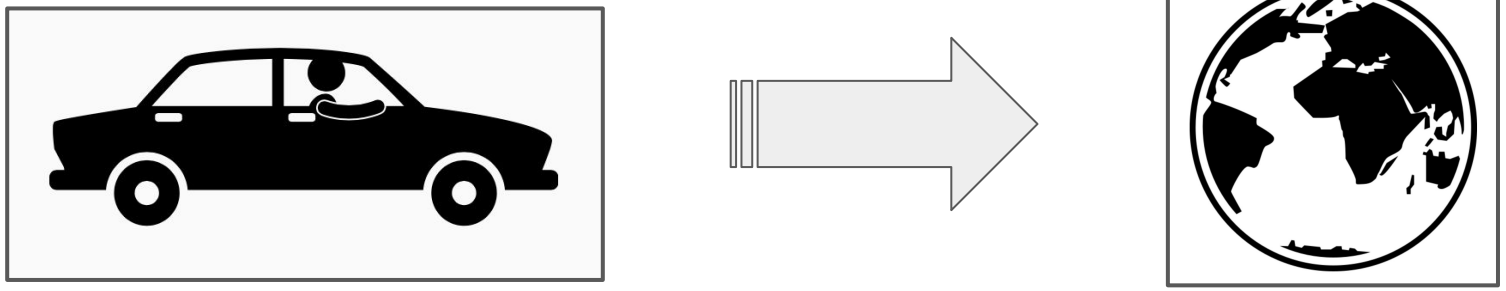
All reasonable metrics provide imperfect discriminative power, and that's fine-- **Do you know where your metrics blindspots are?**

Do you use binning on numerical variable, or threshold cut-offs like the pie chart metric? Bin sizes and thresholds are decision boundaries that create blind spots.

How does your metric aggregate information? Does it take an average, find a precentile, or fit a curve? What type of details is it glossing over when it does this?

Does your metric project data into euclidean (cartesian/vector) space? What information might be lost in that projection.

Tips & Tricks: Checking Edge Cases



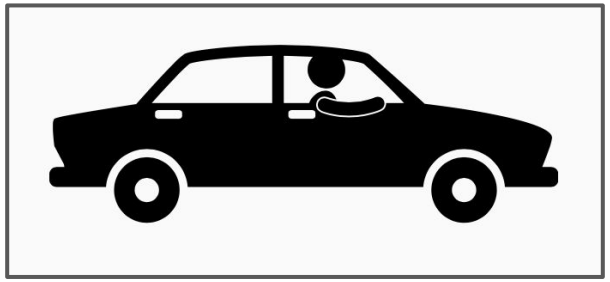
neighborhood	year	month	0	1	2	...	171	172	173
0	2019	1	0	0	0	...	0	0	0
0	2019	2	0	0	0	...	0	0	0
0	2019	3	0	0	0	...	0	0	0
...	...	...	...	...	...	...	...	...	...
277	2019	10	0	0	0	...	0	0	0
277	2019	11	0	0	0	...	0	0	0
277	2019	12	0	0	0	...	0	0	0

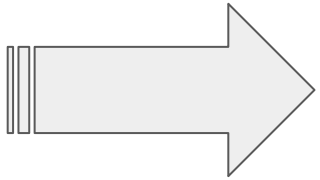
What happens to your metric when the ground truth is full of zeros, and the privatized data isn't? What about when there's only a single record? What happens when the privatized data has many, many more records than the ground truth?

What if the input schema only has a single numerical feature, and the rest are categorical? What if it only has one categorical feature and the rest are numerical?

Doing a good debugging on your metric is a good idea to avoid unexpected and alarming behavior down the road. Think carefully through how your metric behaves at extreme or unusual inputs. Make sure you clearly identify any assumptions you're making about what inputs are valid.

Tips & Tricks: Checking Stability





$$\frac{3}{4} = 0.75$$

$$\frac{399}{400} = 0.9975$$

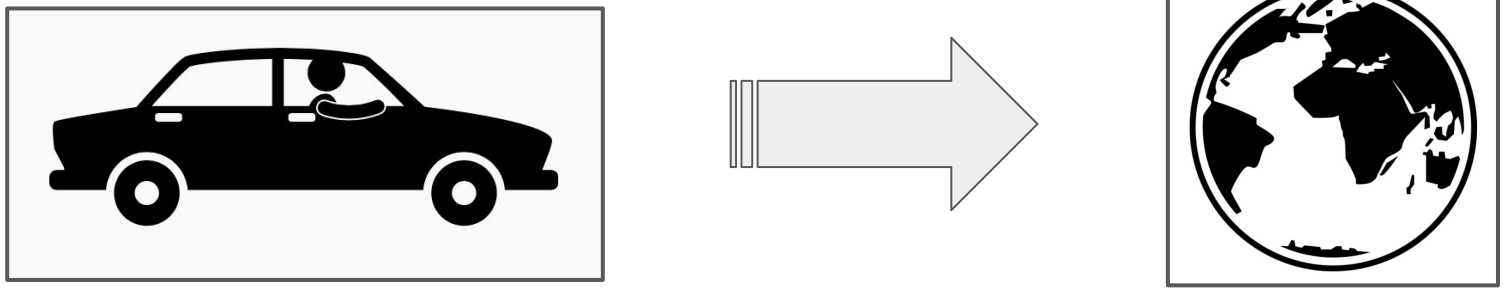
Ratios get strange when the numbers are small.
Randomization, if you're using too small a sampling ratio, can produce wildly different answers depending on what sample you get.

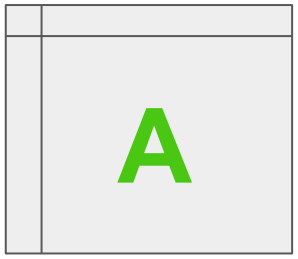
How stable is your metric?

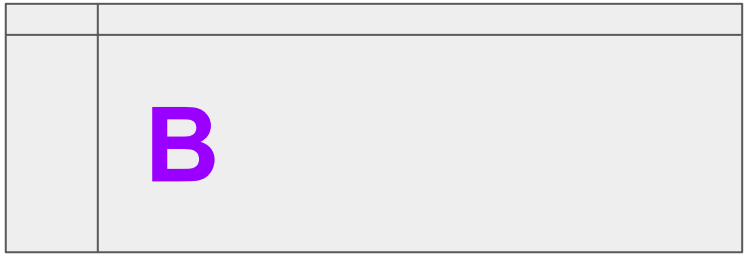
Run it multiple times on the same input (if randomized) and check the distribution. See how it behaves on data sets at the extremes (very sparse data, very dense data).

It doesn't need to work perfectly everywhere, but we need to understand in what contexts the results are stable and dependable, and in what contexts we may need to run multiple trials, or go with a different metric.

Tips & Tricks: Checking Comparability







Take a look at your metric and check this real quick-- How do the numbers change depending on the size of the input data? The number of possible record types? The number of numerical features vs. categorical features? How many zeros (sparseness) there is in the ground truth data?

When you get a score of 700 on a data-set in Schema A, and a score of 600 on a data-set in Schema B, does it really mean that the second data set is worse quality? Or does it just mean that the second data-set is *larger*?

How do your metric scores change dependent on the schema of the data, independent of the data quality itself?

It's fine if your metric isn't comparable between different data schemas, but understanding those properties is important to ensuring your metric isn't accidentally misused to produce misleading or invalid performance rankings.