The forgotten practicalities of machine learning:
Dirty Data

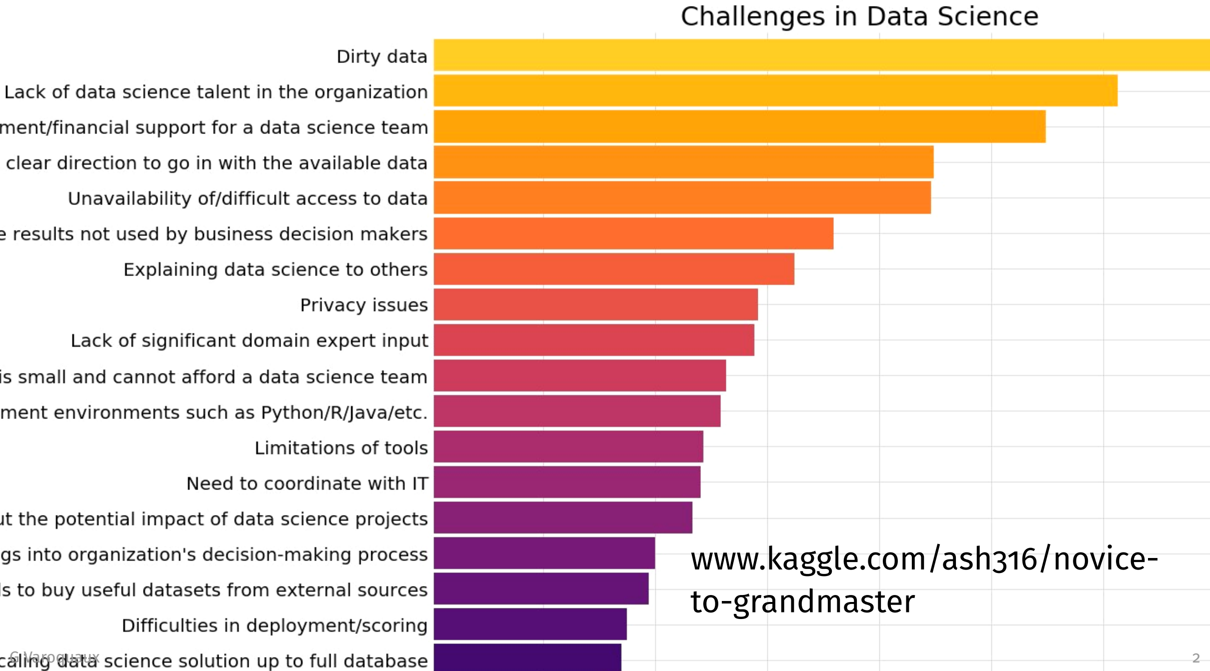
Gaël Varoquaux
Inria

About me

I spent $>$ 15 years data processing for science and applications

- PhD in physics
- Brain imaging, medical imaging, cognitive neuroscience
- Current focus: machine learning & health
- Co-founded `scikit-learn`

Dirty Data is the number one roadblock to data science



www.kaggle.com/ash316/novice-to-grandmaster

From “big” data to “dirty” data

- Plenty of data is needed
  - AI models are data hungry
  - for generalizable findings

- Increasing quantity degrades quality:
  - aggregation across multiple sources
  - opportunistic collection (not the right information)

From “big” data to “dirty” data

- Plenty of data is needed
  - AI models are data hungry
  - for generalizable findings

- Increasing quantity degrades quality:
  - aggregation across multiple sources
  - opportunistic collection (not the right information)

How to analyze the resulting mess?

Typical answer: curate it

Carefully create “cleaner” representations, easier to model

Dirtiness that breaks our toolbox

Machine learning Let $\mathbf{X} \in \mathbb{R}^{n \times p}$

Real-life data science

Gender	Experience	Age	Employee Position Title
M	10 yrs	42	Master Police Officer
F	23 yrs	NA	Social Worker IV
M	3 yrs	28	Police Officer III
F	16 yrs	45	Police Aide
M	13 yrs	48	Electrician I
M	6 yrs	36	Bus Operator
M	NA	62	Bus Operator
F	9 yrs	35	Social Worker III
F	NA	39	Library Assistant II
M	8 yrs	NA	Library Assistant I

Dirtiness that breaks our toolbox

Machine learning   Let $\mathbf{X} \in \mathbb{R}^{n \times p}$

Real-life data science

Gender	**Experience**	**Age**	**Employee Position Title**
M	10 yrs	42	Master Police Officer
F	23 yrs	NA	Social Worker IV
M	3 yrs	28	Police Officer III
F	16 yrs	45	Police Aide
M	[illegible]	[illegible]	[illegible]ctrician I
M	6 yrs	36	Bus Operator
M	NA	62	Bus Operator
F	9 yrs	35	Social Worker III
F	NA	39	Library Assistant II
M	8 yrs	NA	Library Assistant I

Dirty Categories/Entities ☹

Dirtiness that breaks our toolbox

Machine learning   Let $\mathbf{X} \in \mathbb{R}^{n \times p}$

Real-life data science

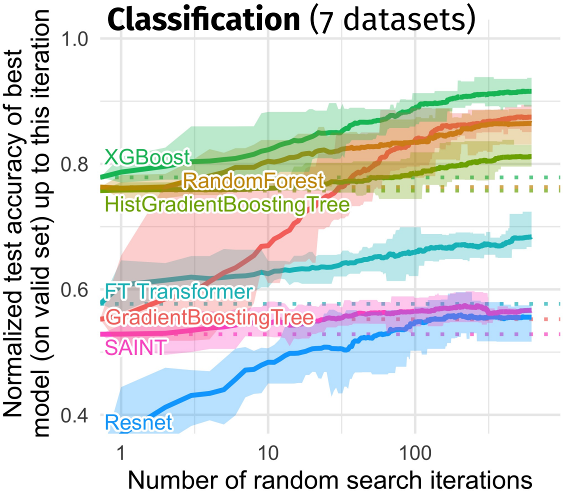
Gender	Experience	Age	Employee Position Title
M	10 yrs	42	Master Police Officer
F	23 yrs	NA	Social Worker IV
M	3 yrs	28	Police Officer III
F	16 yrs	45	Police Aide
M			Electrician I
M	6 yrs	36	Bus Operator
M	NA	62	Bus Operator
F	9 yrs	35	Social Worker III
F	NA	39	Library Assistant II
M	8 yrs	NA	Library Assistant I

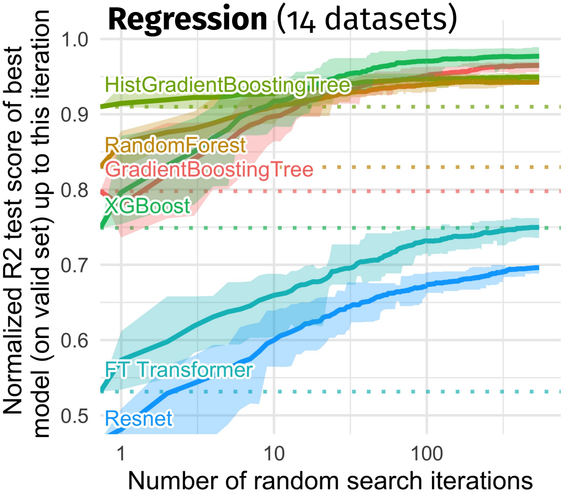
Missing values ☹

Tabular ~~deep~~ learning

[Grinsztajn... 2022]

Tabular data = columns have different meanings (age, sex, glucose)





Tree-based models are best

Settings: supervised learning as statistical modeling

- Given n pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ drawn <u>i.i.d.</u>
  find a function $f : \mathcal{X} \to \mathcal{Y}$ such that $f(x) \approx y$

<u>Notation</u>: $\hat{y} \stackrel{\text{def}}{=} f(x)$

Empirical risk minimization

- Loss function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$

- Estimation of f: $$f^\star = \operatorname*{argmin}_{f \in \mathcal{F}} \mathbb{E}\big[l(\hat{y}, y)\big]$$

For l: quadratic loss, $f^\star(x) = \mathbb{E}[y|x]$

Settings: supervised learning as statistical modeling

- Given n pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ drawn <u>i.i.d.</u>
  find a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ such that $f(x) \approx y$

<u>Notation</u>: $\hat{y} \stackrel{\text{def}}{=} f(x)$

Empirical risk minimization

- Loss function $l : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$
- Estimation of f: $f^\star = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \, \mathbb{E}\big[l(\hat{y}, y)\big]$

In practice, $\hat{\mathbb{E}}$ not $\mathbb{E}$
$\Rightarrow$ good choice of function class $\mathcal{F}$ (inductive bias, restricting model fit)
+ not an actual argmin (regularization, dropout, penalties...)

Course outline

1 Non-normalized discrete entities / categories

Entities: more learning, rather than more cleaning
Dirty categories from strings

2 Missing values

The classical missing-values framework
Rethinking imputation for prediction
Architectures for missing values

1 Non-normalized discrete entities / categories

Gender	Experience	Employee Position Title
M	10 yrs	Master Police Officer
F	23 yrs	Social Worker IV
M	3 yrs	Police Officer III
F	16 yrs	Police Aide
M	13 yrs	Electrician I
M	6 yrs	Bus Operator
M	29 yrs	Bus Operator
F	9 yrs	Social Worker III
F	6 yrs	Library Assistant II
M	8 yrs	Library Assistant I

With
- P. Cerda
- A. Cvetkov-Iliev

1 Non-normalized discrete entities / categories

Entities: more learning, rather than more cleaning

Dirty categories from strings

Example study: salaries across institutions

https://salaries.texastribune.org

Questions of interest

- How does experience impact salary for managers vs assistants?
- What is the typical pay gap between sexes?

[Cvetkov-Iliev... 2022]

Example study: an entity-matching challenge

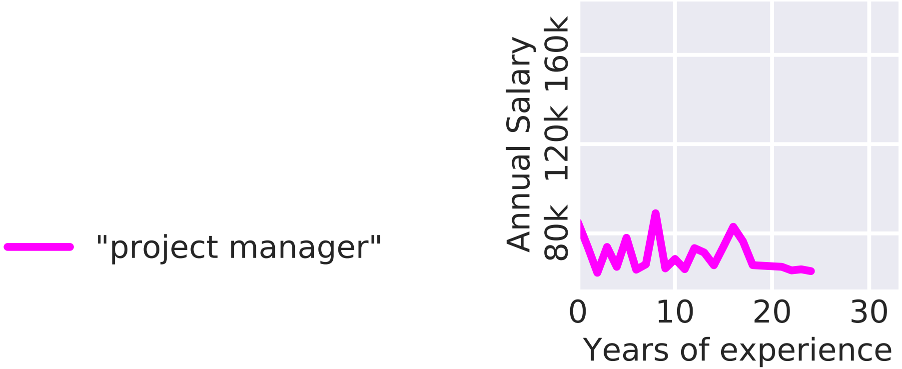
Analysis across institutions

Job Title	Experience	Salary
0712 - postdoctoral fellow	1	65k
data scientist	3	90k
senior research associate	8	110k

Job Title	Experience	Salary
professor	5	72k
sr research assoc	4	100k
postdoctoral research associate	2	49k

[Cvetkov-Iliev... 2022]

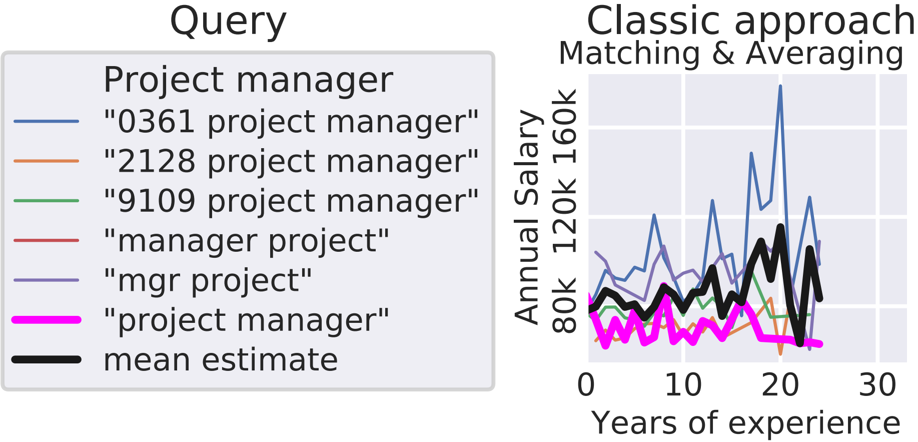
Example study: salary function of experience & position



All the instances of "project manager" in the data

[Cvetkov-Iliev... 2022]

Example study: salary function of experience & position

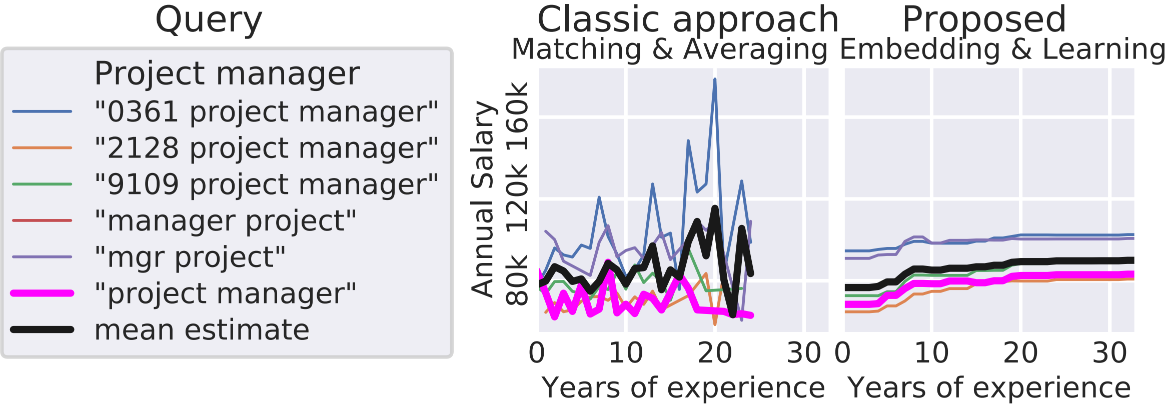


All the entries matched to “project manager” in the data

Manual entity matching using `openrefine`
3 days work, 1 000 matchs

[Cvetkov-Iliev... 2022]

Example study: salary function of experience & position



Estimates of $\mathbb{E}[\text{Salary} \,|\, \text{Job, Experience}]$

- Word embeddings of entries (`fasttext`)
- Machine learning to target Salary = f(Job, experience)

[Cvetkov-Iliev... 2022]

Example study: sex pay gap

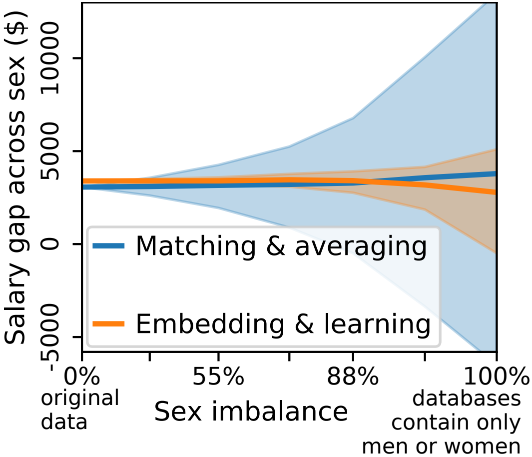
[Cvetkov-Iliev... 2022]

Compare pay of women vs men <u>all other things kept constant</u>

Machine-learning estimation:
Predict the counterfactual: salary of a woman, *had she been a man*

Doubly-robust causal inference

- $\mathbb{E}$[Salary | Sex, Job, Experience...]
- $\mathbb{P}$[Sex | Job, Experience...]



Consistency of results when matching men to women within vs across institutions

Example study: which approach is most *valid*

Cross-validation to measure quality of estimates

Estimation method	**Manual matching**	**Salary** (RMSE)	**Quantile** (MAE)	**Propensity** (Brier score)
Matching & averaging	Yes	55634	31802	0.231
Embedding & Learning	No	52683	28726	0.189
Embedding & Learning	Yes	50614	26713	0.184

Lower is better

$\Rightarrow$ Both cleaning & learning help

- But only learning is better than only cleaning
- Cleaning is 3 days manual labor ☹

[Cvetkov-Iliev... 2022]

More learning, rather than more cleaning

- Non-parametric flexible models capture errors better than cleaning

  [Cvetkov-Iliev... 2022]

  Supervised learning = modeling errors for a purpose

- Cleaning & parametric modeling are needed

  because we reason on model parameters

  But these models are imperfect simplification of reality

  Imperfection of modeling and cleaning compromise the validity of findings [Varoquaux 2021]

Analytics –beyond prediction– on top of supervised learning can enable easier, more valid, analysis

1 Non-normalized discrete entities / categories

Entities: more learning, rather than more cleaning

Dirty categories from strings

Non-normalized categories break statistical pipelines

- Categorical-ish data

- Standard statistical practice:
  one-hot encoding

  Breaks due to high-cardinality

  Looses links between entries

Employee Position Title
Master Police Officer
Social Worker IV
Police Officer III
Police Aide
Electrician I
Bus Operator
Bus Operator
Social Worker III
Library Assistant II
Library Assistant I

Traditional view: data curation & database normalization

Feature engineering

Employee Position Title
Master Police Officer
Social Worker III
Police Officer II
Social Worker II
Police Officer III

$\Longrightarrow$

Position	Rank
Police Officer	Master
Social Worker	III
Police Officer	II
Social Worker	II
Police Officer	III

Traditional view: data curation & database normalization

Feature engineering

Employee Position Title
Master Police Officer
Social Worker III
...

⟹

Position	Rank
Police Officer	Master
Social Worker	III
...	

Merging *entities*

- Output a “clean” database
- Difficult without supervision

- Potentially suboptimal

Pfizer Corporation Hong Kong $\overset{?}{=}$ Pfizer Pharmaceuticals Korea

Deduplication

Company name
Pfizer Inc.
Pfizer Pharmaceuticals LLC
Pfizer International LLC
Pfizer Limited
Pfizer Corporation Hong Kong Limited
Pfizer Pharmaceuticals Korea Limited
...

Traditional view: data curation & database normalization

Feature engineering

Employee Position Title
Master Police Officer
Social Worker III
...

$\Longrightarrow$

Position	Rank
Police Officer	Master
Social Worker	III
...	

Merging *entities*

- Output a “clean” database

Deduplication

Company name
Pfizer Inc.
Pfizer Pharmaceuticals LLC
...

Hard to make automatic and turn-key

Harder than supervised learning

$\Rightarrow$ the analytic question should guide the curation

Adding string similarity fixes statistical pipelines

On many real-life datasets [Cerda... 2018]

- a simple string similarity boosts statistical analysis
- more than deduplication

	London	Londres	Paris
Londres	0.3	1.0	0.0
London	1.0	0.3	0.0
Paris	0.0	0.0	1.0

string_distance(Londres, London)

Works best combined with a powerful model, such as gradient-boosted trees

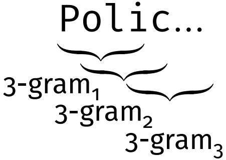
Modeling substrings

Drug Name
alcohol
ethyl alcohol
isopropyl alcohol
polyvinyl alcohol
isopropyl alcohol swab
62% ethyl alcohol
alcohol 68%
alcohol denat
benzyl alcohol
dehydrated alcohol

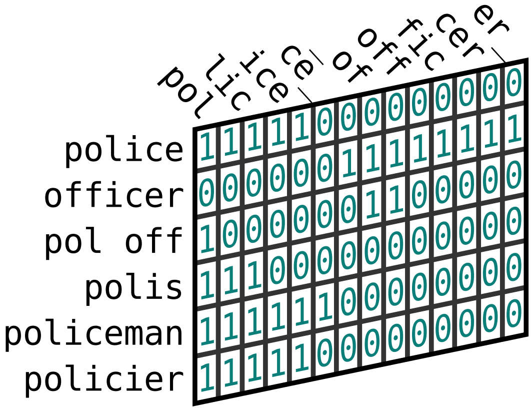
Employee Position Title
Police Aide
Master Police Officer
Mechanic Technician II
Police Officer III
Senior Architect
Senior Engineer Technician
Social Worker III

GapEncoder: Embedding via string forms

Factorizing sub-string count matrices



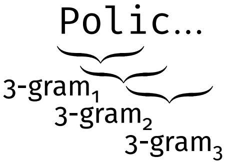
Models strings as a linear combination of substrings



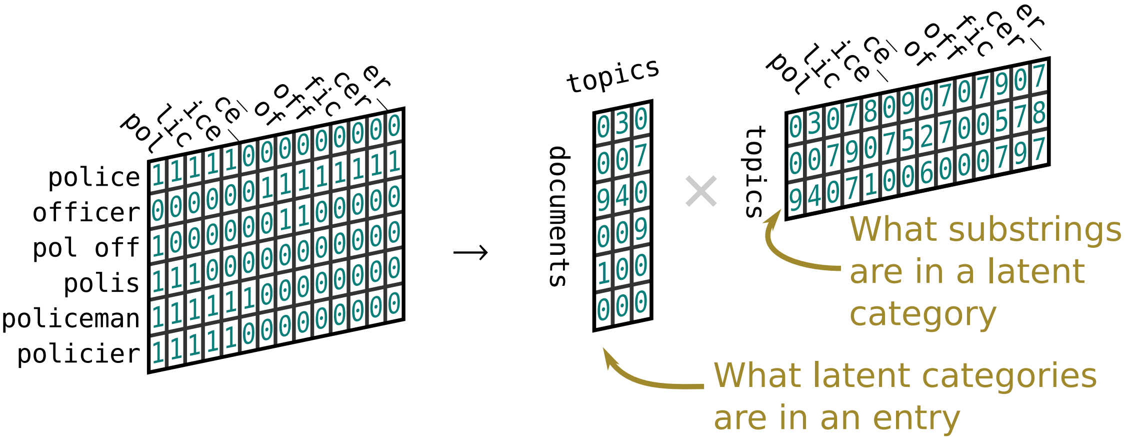
[Cerda and Varoquaux 2020]

GapEncoder: Embedding via string forms

Factorizing sub-string count matrices



Models strings as a linear combination of substrings



[Cerda and Varoquaux 2020]

GapEncoder: Gamma-Poisson factorization

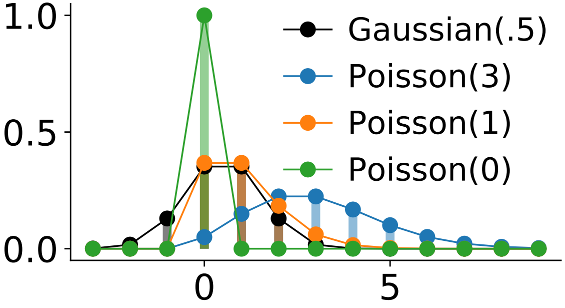
X is a matrix of counts

- Topic modeling [Canny 2004]
- String entries [Cerda and Varoquaux 2020]

$\Longrightarrow$ Poisson loss, instead of squared loss

$$\mathbb{P}(\mathbf{x}_j|\mathbf{w}_j) = \text{Poisson}\big(\mathbf{w}_j\big) = {}^{1}/_{x_j!}\, \mathbf{w}_j^{x_j} e^{-\mathbf{w}_j}$$

Counts are not well approximated by a Gaussian



GapEncoder: Gamma-Poisson factorization

X is a matrix of counts

- Topic modeling [Canny 2004]
- String entries [Cerda and Varoquaux 2020]

$\implies$ Poisson loss, instead of squared loss

$$\mathbb{P}(\mathbf{x}_j|\mathbf{u},\mathbf{V}) = \text{Poisson}\big((\mathbf{u}\,\mathbf{V})_j\big) = {}^1\!/_{x_j!}\,(\mathbf{u}\,\mathbf{V})_j^{x_j} e^{-(\mathbf{u}\,\mathbf{V})_j}$$

u are loadings,
modeled as random with a Gamma prior[1]

$$\mathbb{P}(u_i) = \frac{u_i^{\alpha_i - 1} e^{-u_i/\beta_i}}{\beta_i^{\alpha_i}\,\Gamma(\alpha_i)}$$

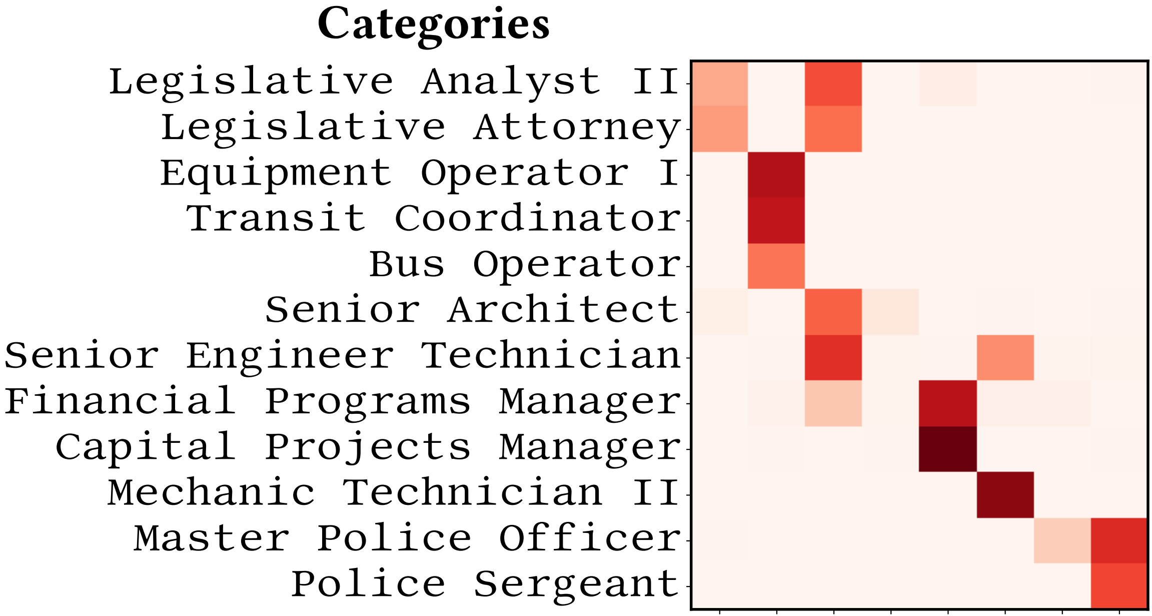
Maximum a posteriori estimation:

$$\hat{\mathbf{U}},\hat{\mathbf{V}} = \underset{\mathbf{U},\mathbf{V}}{\text{argmin}} -\sum_j \Big(\log \mathbb{P}(\mathbf{x}_j|\mathbf{u},\mathbf{V}) + \sum_i \log \mathbb{P}(u_i)\Big)$$

Stochastic MM optimization = robust [Cerda and Varoquaux 2020]

[1]Because it is the conjugate prior of the Poisson, and because it imposes soft sparsity and raises rotational invariance

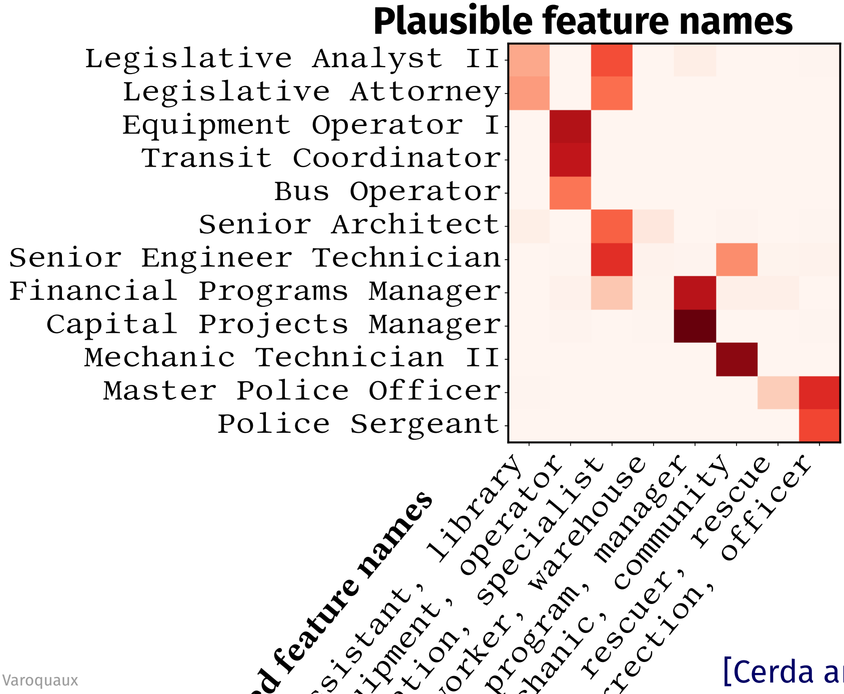
GapEncoder: String embeddings capturing latent categories



Code: dirty-cat.github.io

[Cerda and Varoquaux 2020]

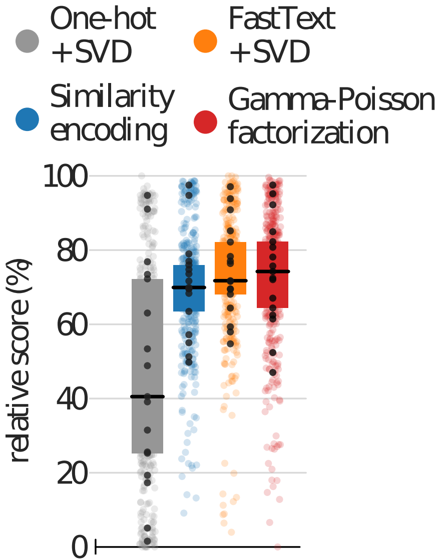
GapEncoder: String embeddings capturing latent categories

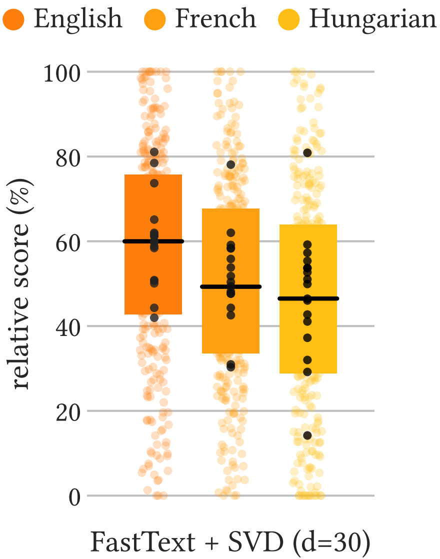


[Cerda and Varoquaux 2020]

Representations tailored to the data

fasttext: almost as good as GapEncoder, if in **the right language**





[Cerda and Varoquaux 2020]

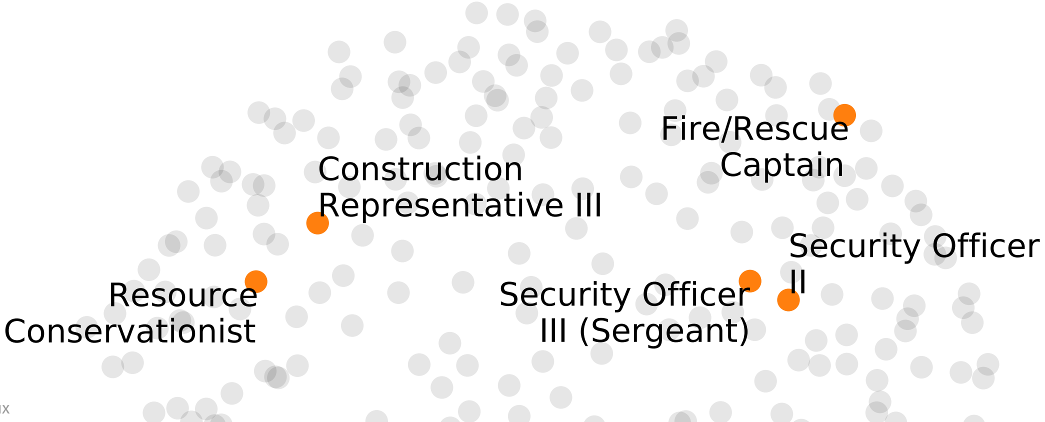
Incorrect entities

Embedding discrete objects into vector spaces is crucial

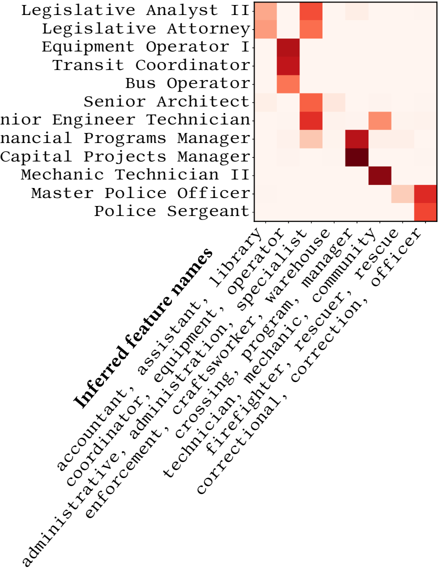
Forces rethinking the analytic pipeline

(flexible models, rather than binning & averaging)

Enables to capture errors as noise



Dirty Categories: Non-normalized entities



Analysis without cleaning
by representing string form to model

GapEncoder – Gamma Poisson encoder:
- Low-dimensional representation
- Interpretable: recovers latent categories

```
from dirty_cat import GapEncoder
enc = GapEncoder()
X = enc.fit_transform(categorical_cols)

enc = SuperVectorizer()
X = enc.fit_transform(dataframe)
```

Code: dirty-cat.github.io

[Cerda and Varoquaux 2020]

2 Missing values

Ubiquitous in health
and social sciences

Gender	Experience	Age
M	10 yrs	42
F	23 yrs	NA
M	3 yrs	28
F	16 yrs	45
M	13 yrs	48
M	6 yrs	36
M	NA	62
F	9 yrs	35
F	NA	39
M	8 yrs	NA

With
- M. Le Morvan
- E. Scornet
- J. Josse

2 Missing values

The classical missing-values framework

Rethinking imputation for prediction

Architectures for missing values

Classical missing-values theory

[Rubin 1976, Josse... 2019]

Model **a)** a distribution f_θ for the complete data $\mathbf{x}$

b) a random process g_ϕ generating a mask $\mathbf{m}$

(full likelihood) $$\mathcal{L}_1(\theta, \phi) = \prod_{i=1}^{n} \int f_\theta(\mathbf{x}_{i,o}, \mathbf{x}_{i,m})\, g_\phi(\mathbf{m}_i | \mathbf{x}_{i,o}, \mathbf{x}_{i,m})\, \mathrm{d}\mathbf{x}_{i,m}$$

Expectation over missing-values mechanism

(ignoring missing mechanism) $$\mathcal{L}_2(\theta) = \prod_{i=1}^{n} \int f_\theta(\mathbf{x}_{i,o}, \mathbf{x}_{i,m})\, \mathrm{d}\mathbf{x}_{i,m}$$

Classical missing-values theory

[Rubin 1976, Josse... 2019]

Model **a)** a distribution f_θ for the complete data $\mathbf{x}$

b) a random process g_ϕ generating a mask $\mathbf{m}$

(full likelihood) $$\mathcal{L}_1(\theta,\phi) = \prod_{i=1}^{n} \int f_\theta(\mathbf{x}_{i,o}, \mathbf{x}_{i,m})\, g_\phi(\mathbf{m}_i | \mathbf{x}_{i,o}, \mathbf{x}_{i,m})\, \mathrm{d}\mathbf{x}_{i,m}$$

Expectation over missing-values mechanism

(ignoring missing mechanism) $$\mathcal{L}_2(\theta) = \prod_{i=1}^{n} \int f_\theta(\mathbf{x}_{i,o}, \mathbf{x}_{i,m})\, \mathrm{d}\mathbf{x}_{i,m}$$

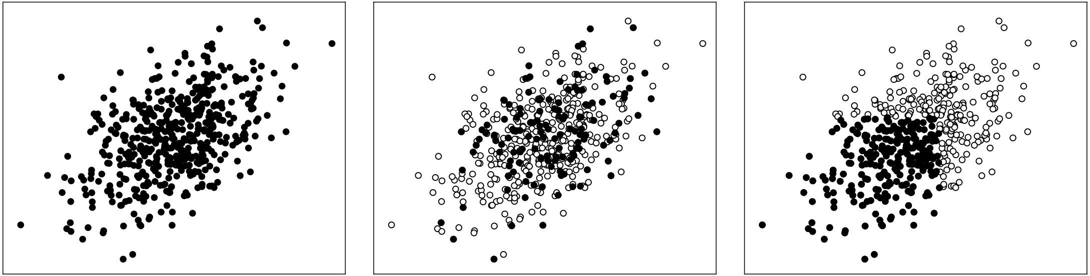
Theorem: In MAR, maximizing $\mathcal{L}_1$ and $\mathcal{L}_2$ give same $\hat{\theta}$

Definition: **Missing at random** situation (MAR)[1]

$$\text{observed}(\mathbf{x}', \mathbf{m}_i) = \text{observed}(\mathbf{x}_i, \mathbf{m}_i) \quad \Rightarrow \quad g_\phi(\mathbf{m}_i | \mathbf{x}') = g_\phi(\mathbf{m}_i | \mathbf{x}_i)$$

[1] for non-observed values, the probability of missingness does not depend on this non-observed value

Ignorable missingness



Complete MAR MNAR (censored)

Missing Not at Random situation (MNAR)

Missingness **not ignorable** $\Rightarrow$ Hard

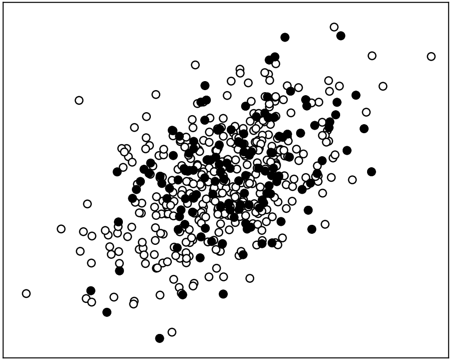
must explicitly model the mechanism

MAR grounds the validity of common statistical procedures

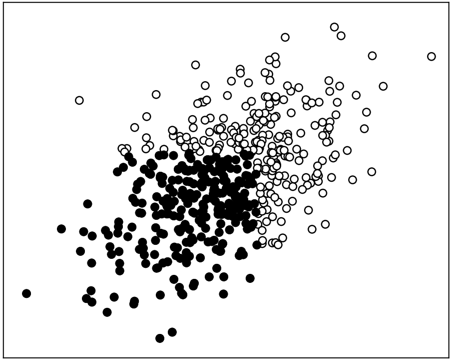
- Expectation Maximization
- Imputation + plug-in estimation

Classic theory

Missing at Random central to statistical practice



MAR



MNAR

2 Missing values

- The classical missing-values framework
- Rethinking imputation for prediction
- Architectures for missing values

Imputation procedures that work out of sample

Mean imputation special case of univariate imputation

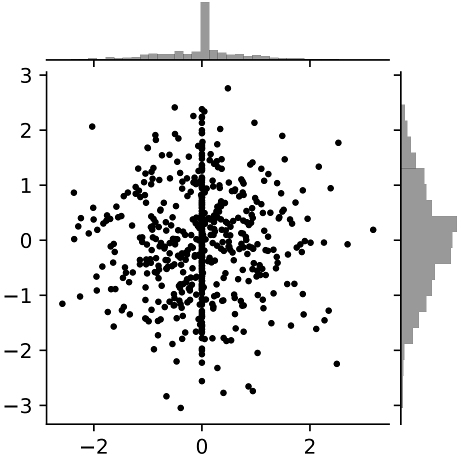
Replace NA by the mean of the feature

`sklearn.impute.SimpleImpute`

Classic statistics point of view

Mean imputation is disastrous: it disorts the distribution

"Congeniality" conditions: imputation must preserve data properties used by later analysis steps



Imputation procedures that work out of sample

Mean imputation special case of univariate imputation

Replace NA by the mean of the feature

`sklearn.impute.SimpleImpute`

Conditional imputation

- Modeling one feature as a function of others
- Possible implementation:
  iteratively predict one feature as a function of other
- Classic implementations in R: MICE, missforest

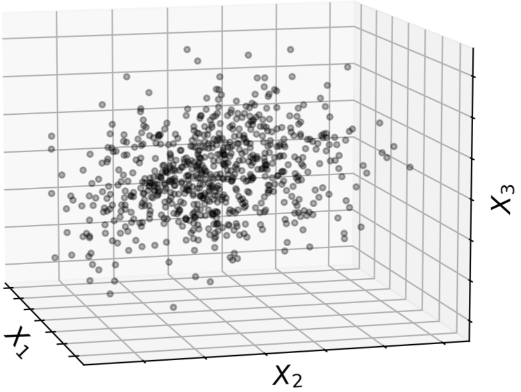
`sklearn.impute.IterativeImputer`

bad computational scalability

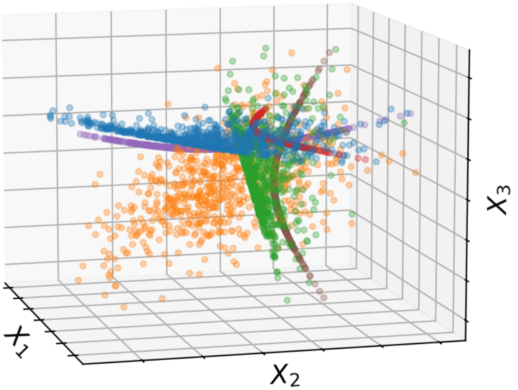
[Le Morvan... 2021]

Theorem (informal): a universally consistent learner trained on imputed data $\Phi(\widetilde{X})$ is Bayes consistent (optimal prediction) for <u>all missing data mechanisms</u> and <u>almost all imputation functions</u>

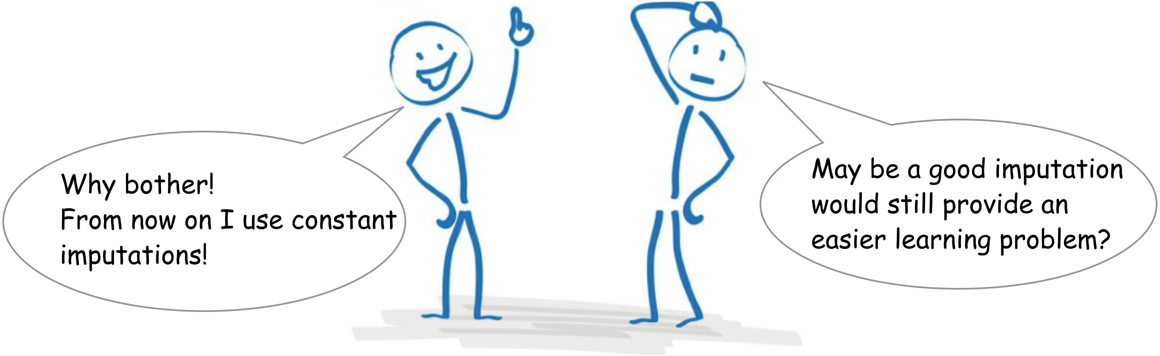
Asymptotically, imputing well is not needed to predict well.



Complete data

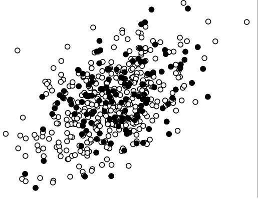


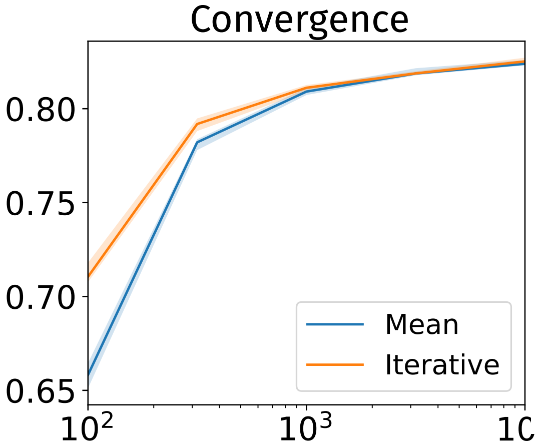
Imputed data (manifolds)

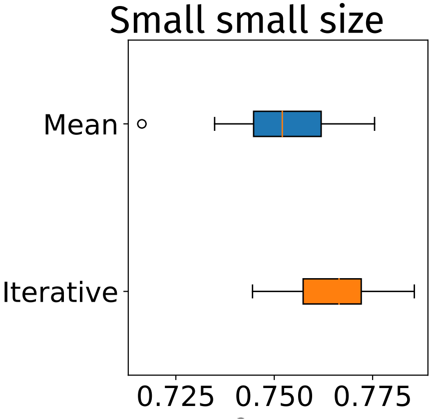

Why bother!
From now on I use constant imputations!
May be a good imputation would still provide an easier learning problem?

Simple simulations

Simulation: MCAR + Gradient boosting







Notebook: github – @nprost / supervised_missing

Imputation is not enough: predictive missingness

Pathological case [Josse... 2019]

y depends only on wether data is missing or not

eg tax fraud detection

theory: MNAR = “Missing Not At Random”

⚠ Imputing makes prediction impossible ⚠

Solution

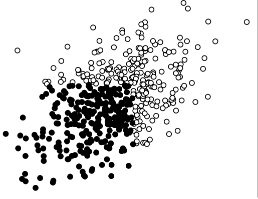
Add a missingness indicator: extra feature to predict

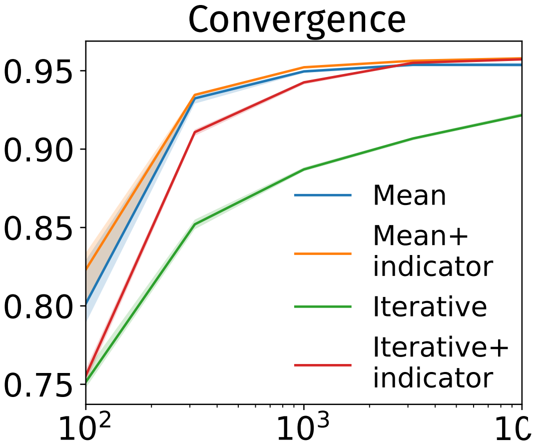
`...SimpleImpute(add_indicator=True)`

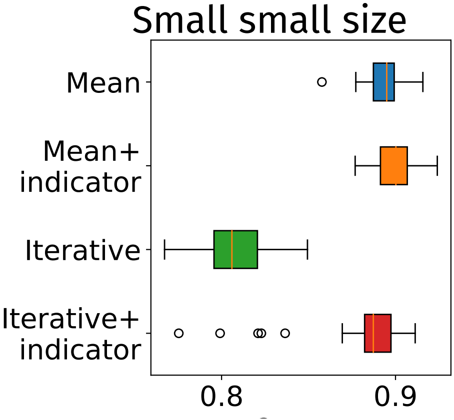
`...IterativeImputer(add_indicator=True)`

Simple simulations

Simulation: Censoring MNAR + Gradient boosting





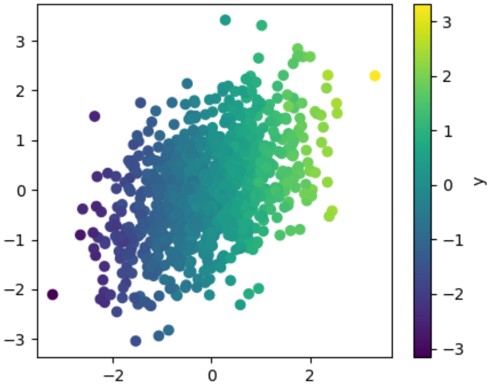


Notebook: github – @nprost / supervised_missing

Imputation imperfection make regression hard

Simple intutions: `http://dirtydata.science/python/`

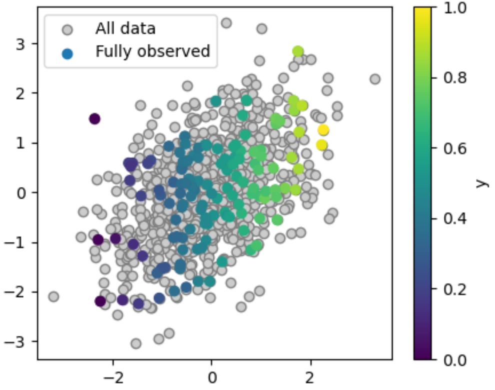
Fully-observed data



Imputation imperfection make regression hard

Simple intutions: `http://dirtydata.science/python/`

MCAR data

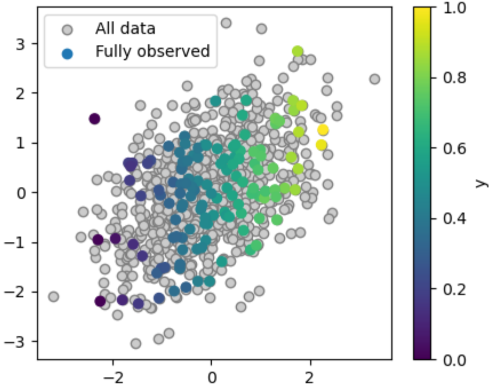


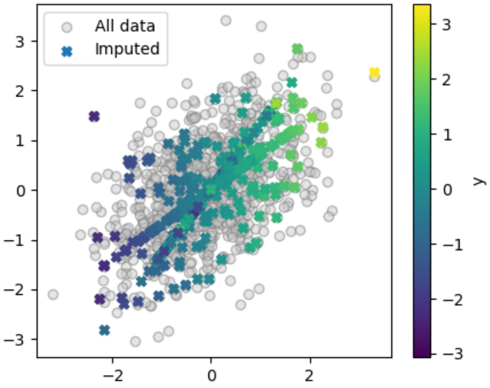
Imputation imperfection make regression hard

Simple intutions: `http://dirtydata.science/python/`

MCAR data

imputed

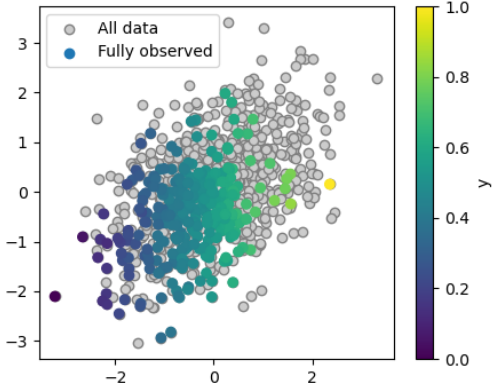




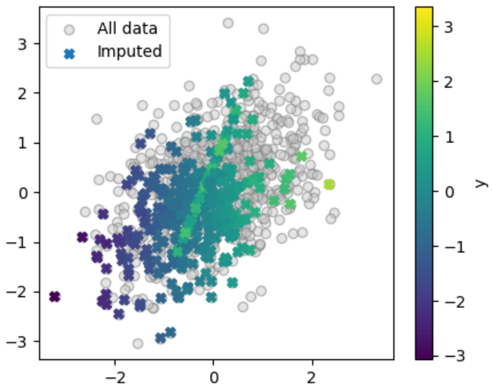
Imputation imperfection make regression hard

Simple intutions: `http://dirtydata.science/python/`

MNAR data



imputed

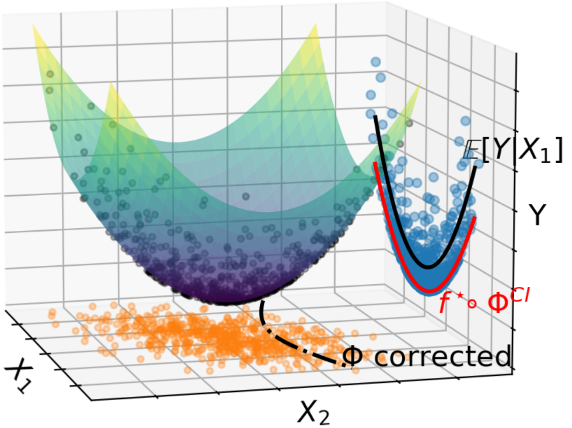


What a good imputation? [Le Morvan... 2021]

Chaining oracles: $f^\star \odot \Phi^{CI}$,
where Φ^{CI} is the oracle imputation $\mathbb{E}[X_{mis}|X_{obs}]$
$f^\star$ optimal predictor without missing values

$\Rightarrow$ Not consistent

Curvature turns omitted variance into bias



What a good imputation?

[Le Morvan... 2021]

1) **Chaining oracles:** 👎 fails

Curvature turns omitted variance into bias

2) **Conditional imputation** $\Phi^{CI} = \mathbb{E}[X_{\text{mis}}|X_{\text{obs}}]$:

$\Rightarrow$ optimal prediction function discontinuous

What a good imputation?

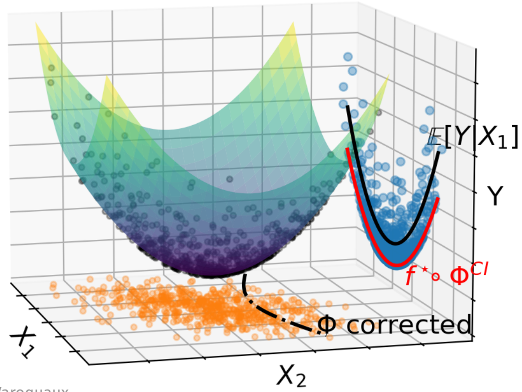
[Le Morvan... 2021]

1) **Chaining oracles:** 👎 fails

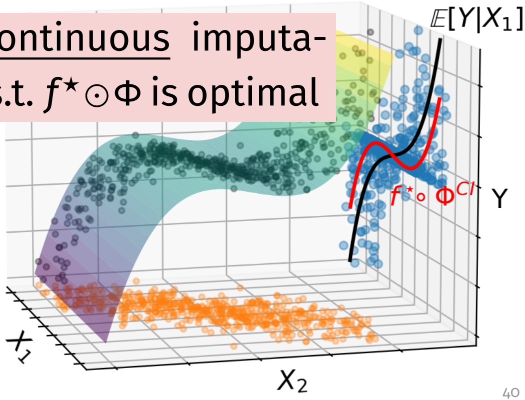
Curvature turns omitted variance into bias

2) **Conditional imputation** Φ^{CI} $\Rightarrow$ discontinous regression function

3) **Fixing** $f^\star$ may lead to discontinuous imputations Φ



No <u>continuous</u> imputation s.t. $f^\star \odot \Phi$ is optimal



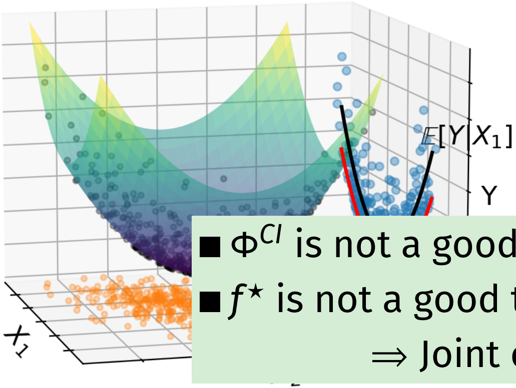
What a good imputation? [Le Morvan... 2021]

1) **Chaining oracles:** 👎 fails

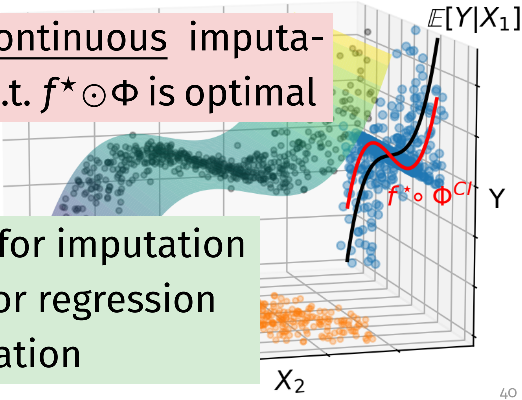
Curvature turns omitted variance into bias

2) **Conditional imputation** Φ^{CI} $\Rightarrow$ discontinous regression function

3) **Fixing** $f^\star$ may lead to discontinuous imputations Φ



No <u>continuous</u> imputation s.t. $f^\star \odot \Phi$ is optimal



- Φ^{CI} is not a good target for imputation
- $f^\star$ is not a good target for regression

$\Rightarrow$ Joint optimization

Rethinking imputation

- A good imputation is one that makes the regression easy
- Close to conditional imputation, but not
- Can work even in MNAR
- Even for interpretation: imputation imperfections propagate

[Le Morvan... 2021]

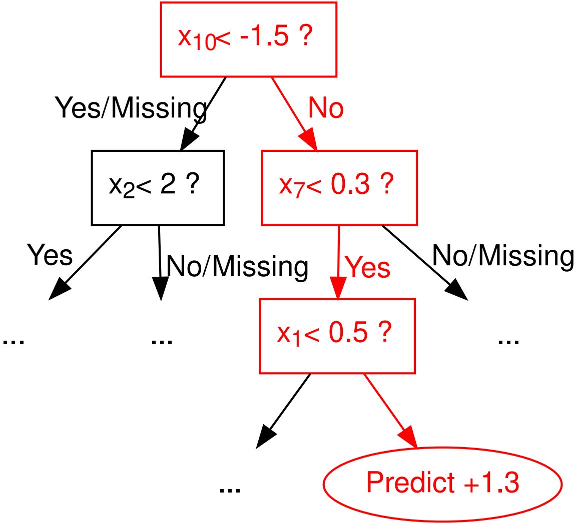
2 Missing values

- The classical missing-values framework
- Rethinking imputation for prediction
- Architectures for missing values

Tree models with missing values

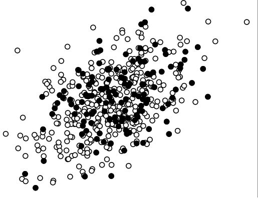
MIA (Missing Incorporated Attribute)
[Josse... 2019]

The learner readily handles missing values



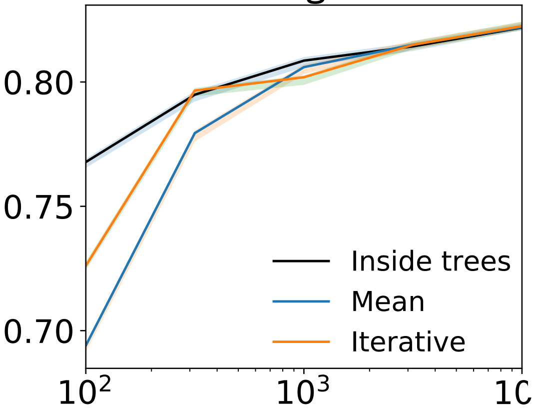
`XGBoost`
`sklearn.ensemble.HistGradientBoostingClassifier`

Simple simulations

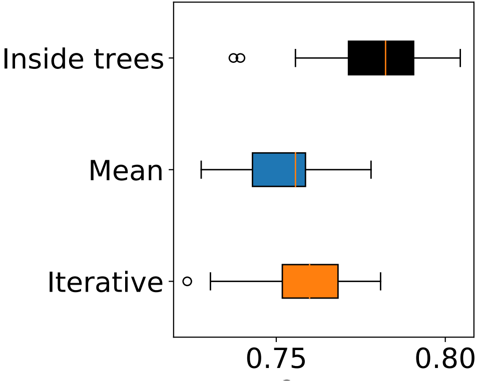


Simulation: MCAR + Gradient boosting

Convergence

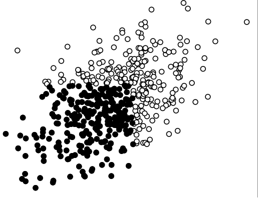


Small small size

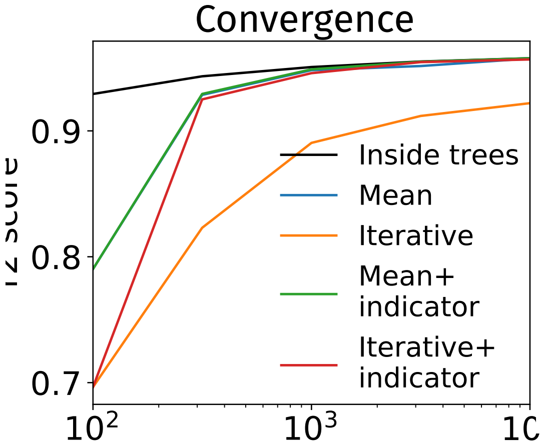


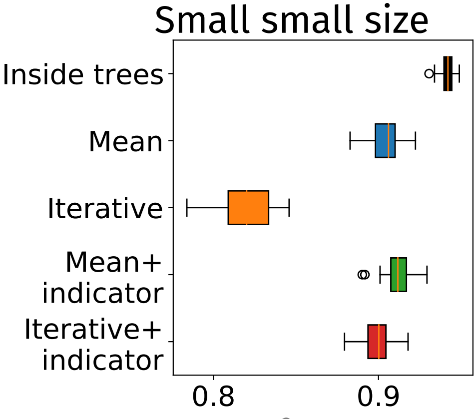
Notebook: github – @nprost / supervised_missing

Simple simulations



Simulation: Censoring MNAR + Gradient boosting





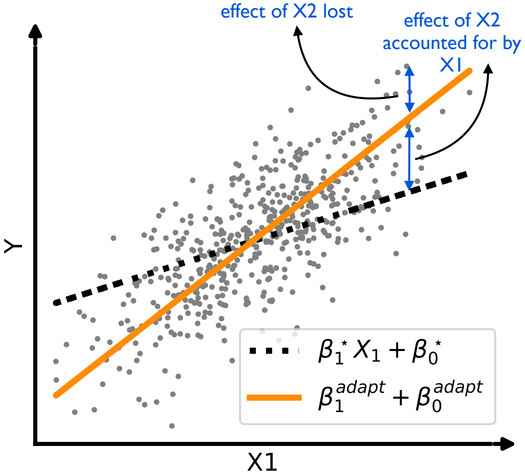
Notebook: github – @nprost / supervised_missing

Continuous predictors with missing values: intuitions

$$Y = \beta_1^\star X_1 + \beta_2^\star X_2 + \beta_0^\star$$

$$\mathrm{cor}(X_1, X_2) = 0.5.$$

If X_2 is missing, the coefficient of X_1 should **compensate for the missingness of X_2**



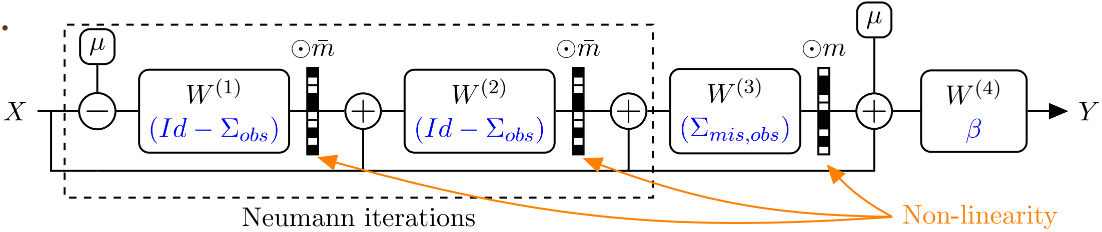
The difficulty of supervised learning with missing values is to handle **up to** 2^d missing data patterns

$\Rightarrow$ Suitable "weight sharing" across patterns

NeuMiss: Designing a missing-values architecture [Le Morvan... 2020]

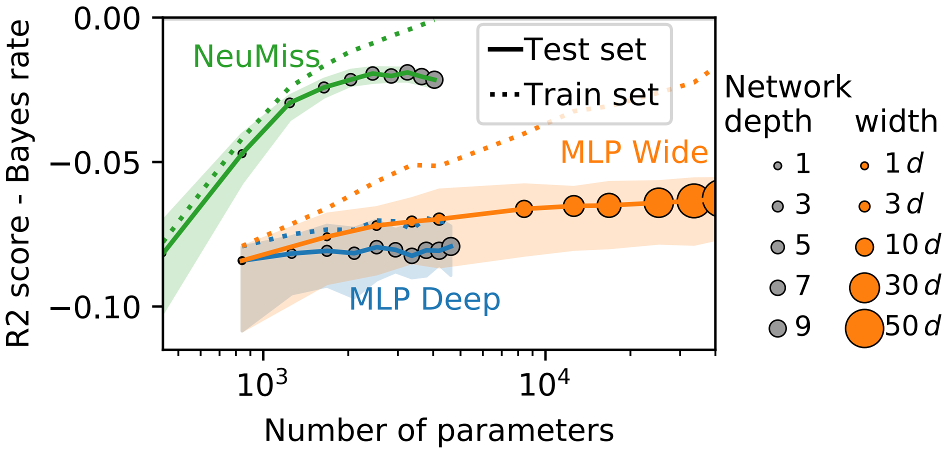
1. Write the form of Bayes predictor in linear, Gaussian settings:
   linear function, with $\ldots\Sigma_{mis,obs}(\Sigma_{obs})^{-1}X_{obs}\ldots$
   in MAR and MNAR (Gaussian self masking)

2. Make it differentiable
   Difficulty: learning Σ_{obs}^{-1}, for any missing data pattern
   Approximate : Σ_{obs}^{-1} by unrolling a NeuMann series

3. 

New non-linearity: multiplication by the missingness mask

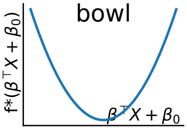
NeuMiss Empirical results: approximation efficiency [Le Morvan... 2020]

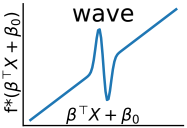


NeuMiss needs less samples to approximate well

(and predict well)

NeuMiss as differentiable imputation: non-linear settings

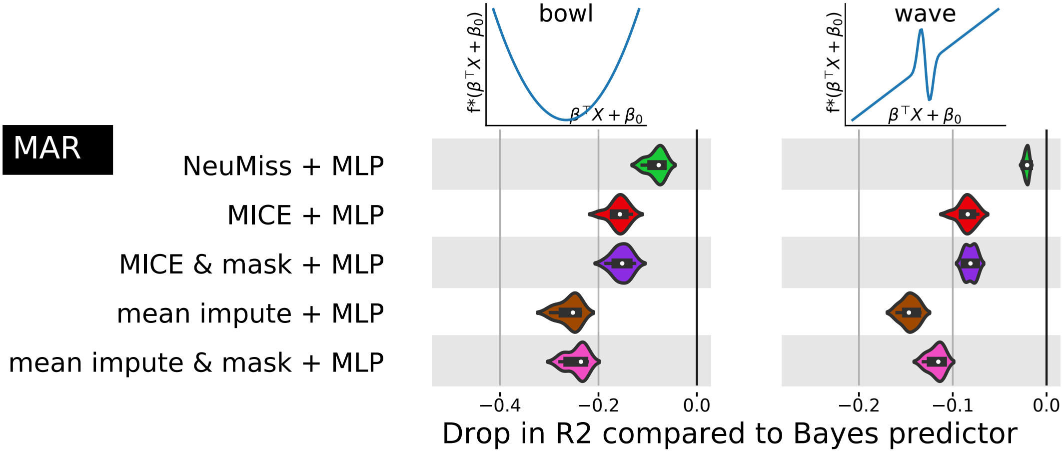




- Using NeuMiss as a block chained with an MLP
- Joint optimization of imputation & regression

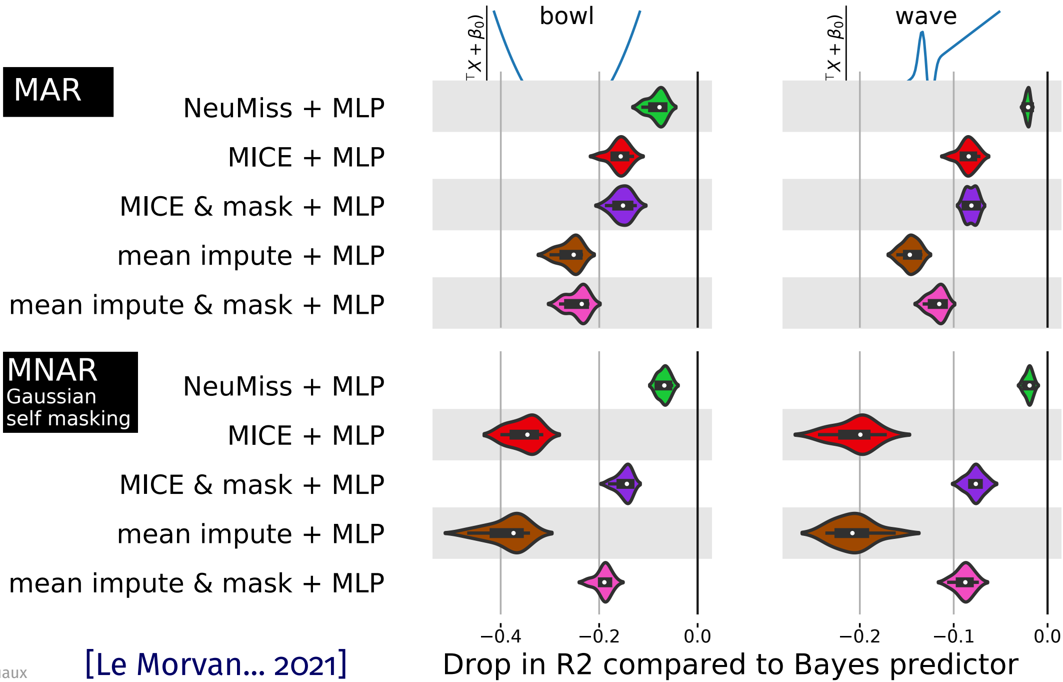
[Le Morvan... 2021]

NeuMiss as differentiable imputation: non-linear settings



[Le Morvan... 2021]

NeuMiss as differentiable imputation: non-linear settings

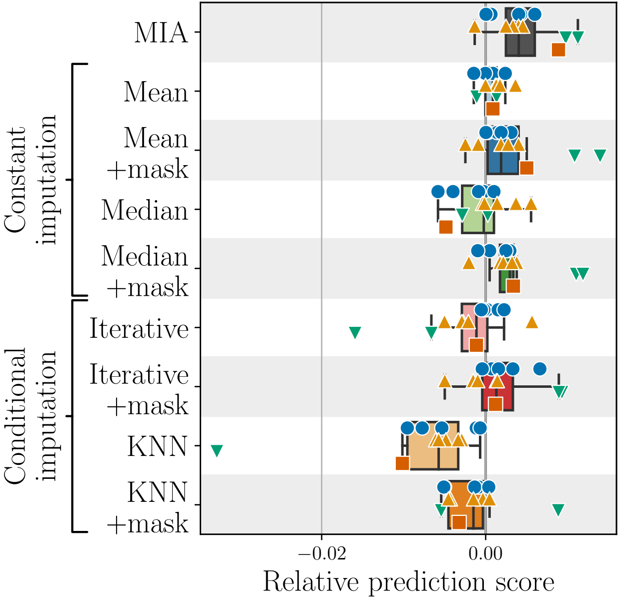


[Le Morvan... 2021]

Real-life benchmarks

[Perez-Lebel... 2022b]

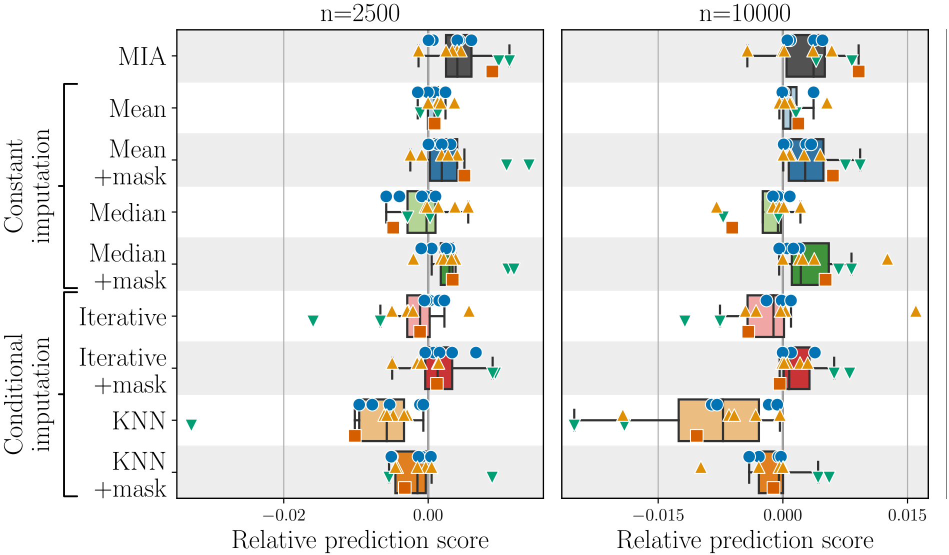
- 13 real-life prediction tasks
- 4 health databases



Real-life benchmarks

[Perez-Lebel... 2022b]

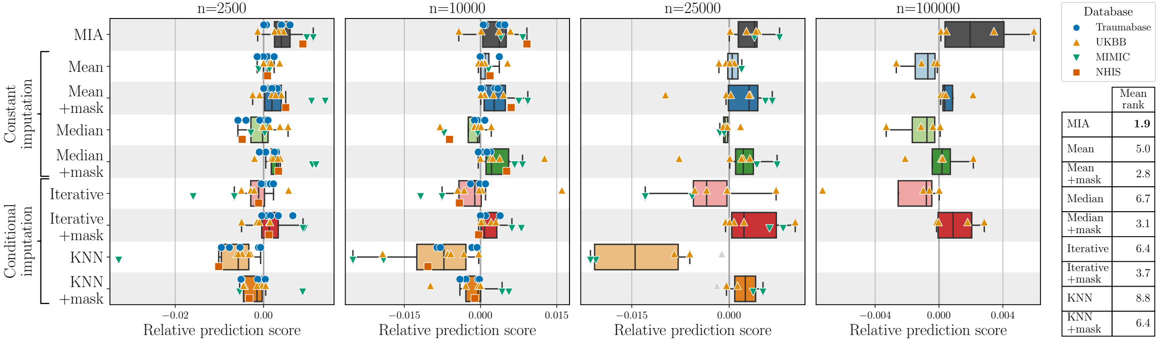
- 13 real-life prediction tasks
- 4 health databases



Real-life benchmarks

[Perez-Lebel... 2022b]

- 13 real-life prediction tasks
- 4 health databases

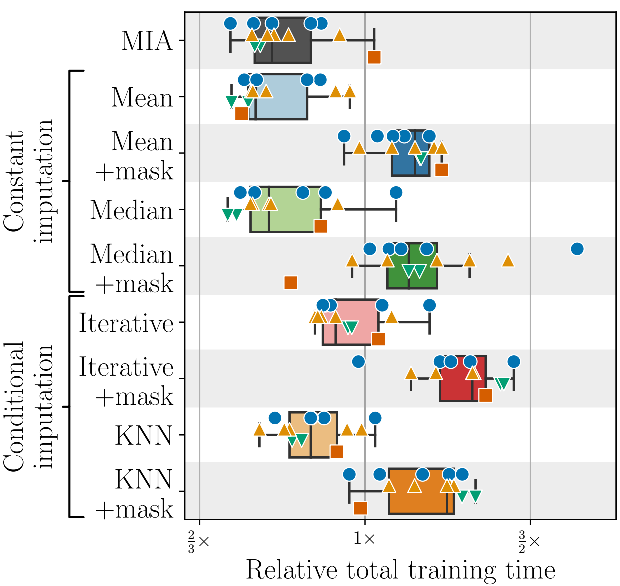


	Mean rank
MIA	**1.9**
Mean	5.0
Mean +mask	2.8
Median	6.7
Median +mask	3.1
Iterative	6.4
Iterative +mask	3.7
KNN	8.8
KNN +mask	6.4

- Adding mask improves ⇒ evidence of MNAR
- KNN-imputer not good, MIA pretty good

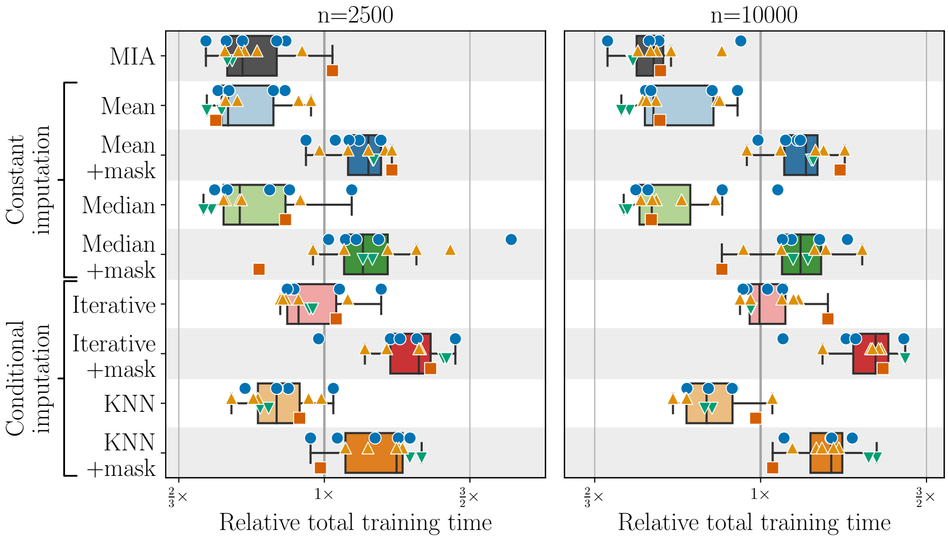
Real-life benchmarks computational cost [Perez-Lebel... 2022b]

- 13 real-life prediction tasks
- 4 health databases



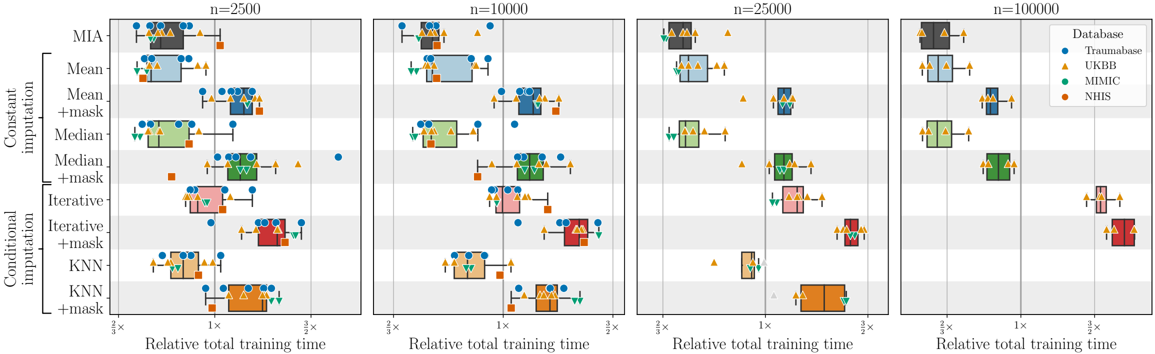
Real-life benchmarks computational cost [Perez-Lebel... 2022b]

- 13 real-life prediction tasks
- 4 health databases



Real-life benchmarks computational cost [Perez-Lebel... 2022b]

- 13 real-life prediction tasks
- 4 health databases



- Imputation comes with high cost –at least $O\left(np^2 \min(n, p)\right)$
- KNN-imputer not good, MIA pretty good

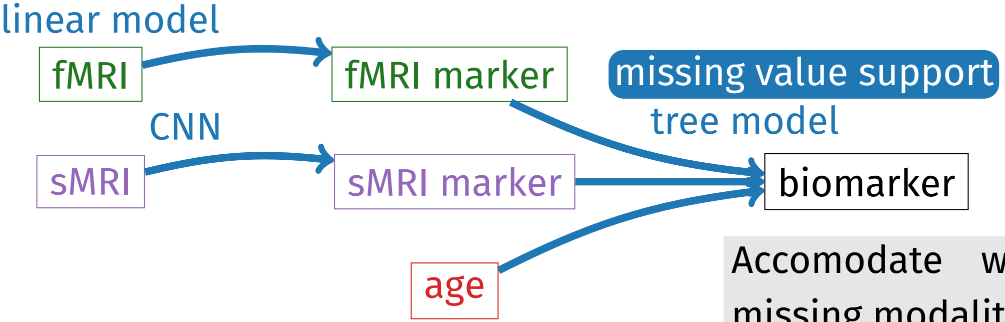
A tangent in medical imaging

Modality-specific models

- On each modality fit a suitable model (deep-learning, linear....)

Non-linear model stacking

- Combine the **predicted** outcome values with other variables (*eg* clinical) as the input of tree model



Accomodate well for missing modality

Supervised learning with missing values

Beyond parametric models

- MAR assumption no longer needed
- conditional imputation not a consistent oracle

NeuMiss networks: approximating the probabilistic model

- optimizable predictor with missing values / imputation
- more scalable than EM; robust to missingness mechanism

In practice: Real-life benchmarks: [Perez-Lebel... 2022a]

- Real databases are MNAR
- Conditional imputation not tractable

Use trees with missing incorporated attribute

`scikit-learn: HistGradientBoostingRegressor`

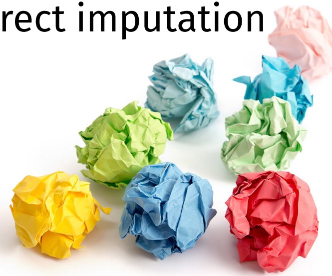
Summary – dirty-data analytics

More learning, less cleaning

- Finding a simple “cleaned” truth is hard or unrealistic
- Exposing glitches to supervised learning, not curating
- The validity of the outcome ensures that of the analysis

Leads to new statistical tradeoffs

- Finding latent fuzzy –continuous– categories
- Missing values analysis valid without MAR / correct imputation



Soda research group: Positions available

https://team.inria.fr/soda/

@GaelVaroquaux

References I

J. Canny. Gap: A factor model for discrete data. In *SIGIR*, page 122, 2004.

P. Cerda and G. Varoquaux. Encoding high-cardinality string categorical variables. *Transactions in Knowledge and Data Engineering*, 2020.

P. Cerda, G. Varoquaux, and B. Kégl. Similarity encoding for learning with dirty categorical variables. *Machine Learning*, pages 1–18, 2018.

A. Cvetkov-Iliev, A. Allauzen, and G. Varoquaux. Analytics on non-normalized data sources: more learning, rather than more cleaning. *IEEE Access*, 2022.

D. A. Engemann, O. Kozynets, D. Sabbagh, G. Lemaître, G. Varoquaux, F. Liem, and A. Gramfort. Combining magnetoencephalography with magnetic resonance imaging enhances learning of surrogate-biomarkers. *eLife*, 9:e54055, 2020.

L. Grinsztajn, E. Oyallon, and G. Varoquaux. Why do tree-based models still outperform deep learning on tabular data? *arXiv:2207.08815*, 2022.

J. Josse, N. Prost, E. Scornet, and G. Varoquaux. On the consistency of supervised learning with missing values. *arXiv preprint arXiv:1902.06931*, 2019.

References II

M. Le Morvan, J. Josse, T. Moreau, E. Scornet, and G. Varoquaux. Neumiss networks: differential programming for supervised learning with missing values. In *Advances in Neural Information Processing Systems 33*, 2020.

M. Le Morvan, J. Josse, E. Scornet, and G. Varoquaux. What's a good imputation to predict with missing values? *NeurIPS*, 2021.

A. Perez-Lebel, G. Varoquaux, M. Le Morvan, J. Josse, and J.-B. Poline. Benchmarking missing-values approaches for predictive models on health databases. *GigaScience*, 2022a.

A. Perez-Lebel, G. Varoquaux, M. Le Morvan, J. Josse, and J.-B. Poline. Benchmarking missing-values approaches for predictive models on health databases. *GigaScience*, 11, 2022b.

D. B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

G. Varoquaux. Ai as statistical methods for imperfect theories. In *NeurIPS 2021-35th Conference on Neural Information Processing Systems. Workshop: AI for Science*, 2021.