Compiling Python Programs into Differentially Private Ones

Johan Leduc

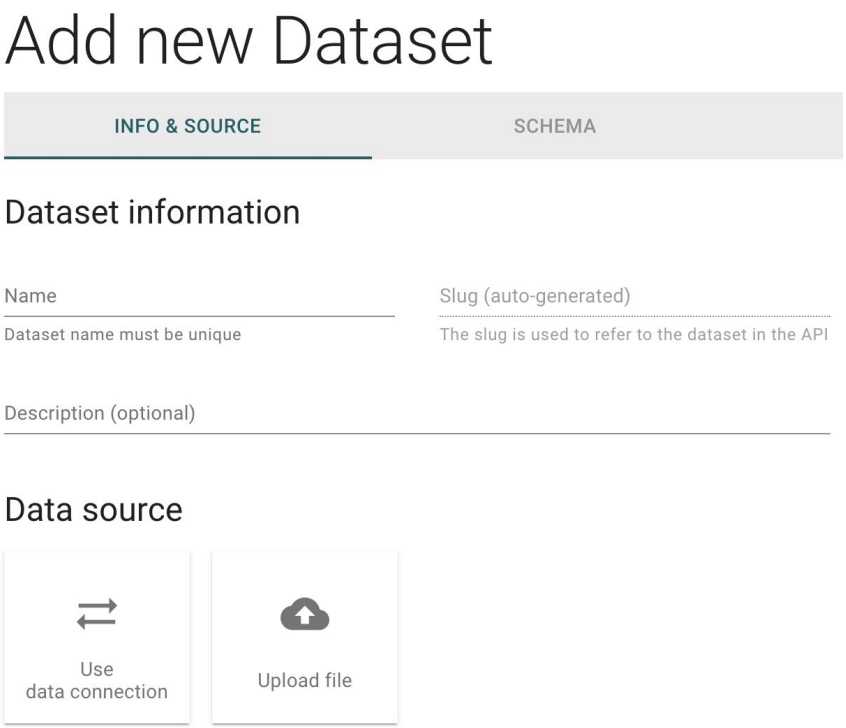
Plan

- Data onboarding
- Data job description
- Differentially Private compilation
- Application

Data onboarding

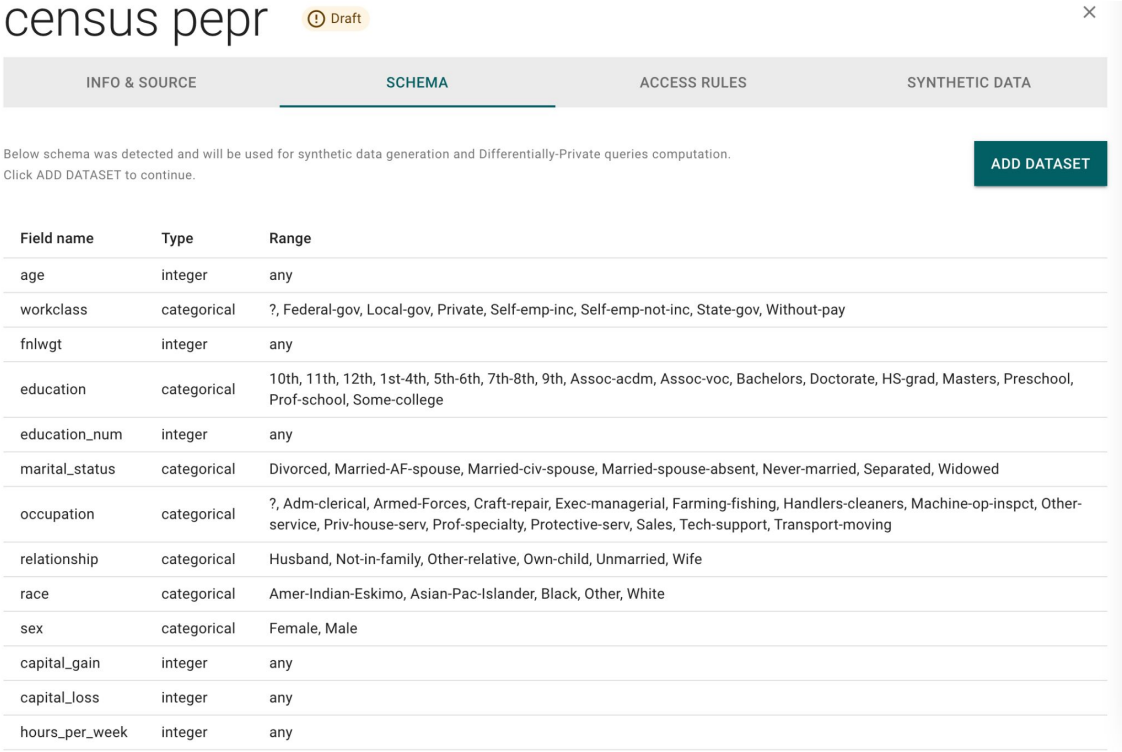
Data onboarding

1. Import sensitive data



Data onboarding

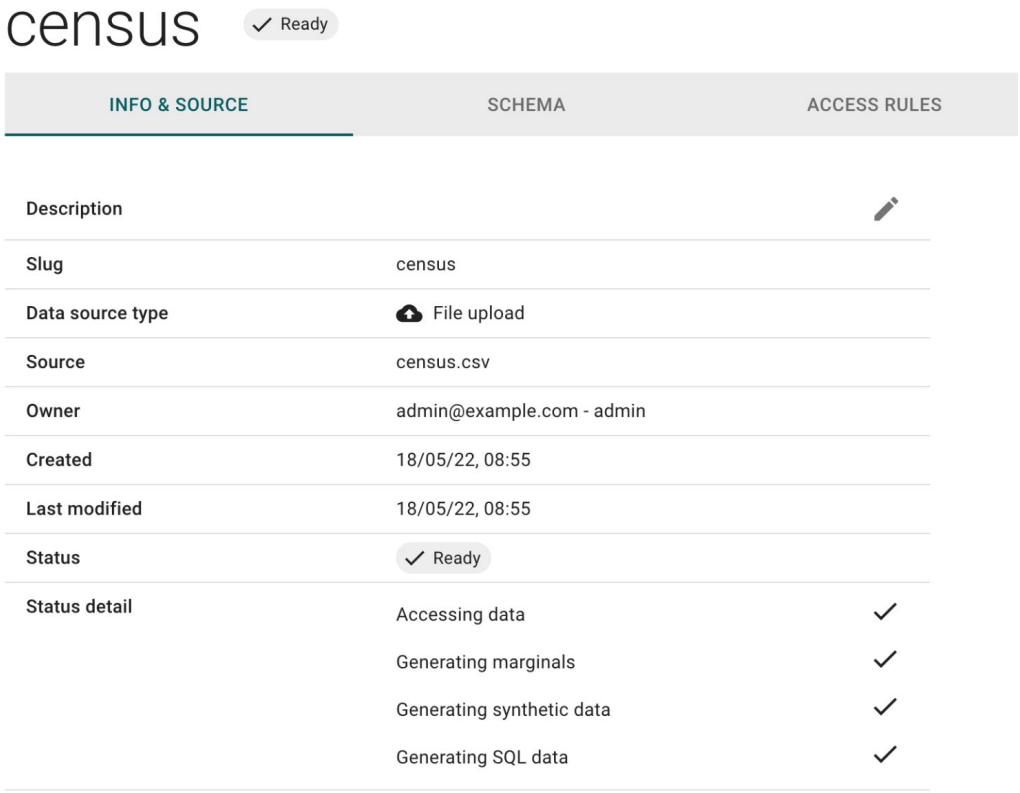
1. Import sensitive data
2. Validate schema



Field name	Type	Range
age	integer	any
workclass	categorical	?, Federal-gov, Local-gov, Private, Self-emp-inc, Self-emp-not-inc, State-gov, Without-pay
fnlwgt	integer	any
education	categorical	10th, 11th, 12th, 1st-4th, 5th-6th, 7th-8th, 9th, Assoc-acdm, Assoc-voc, Bachelors, Doctorate, HS-grad, Masters, Preschool, Prof-school, Some-college
education_num	integer	any
marital_status	categorical	Divorced, Married-AF-spouse, Married-civ-spouse, Married-spouse-absent, Never-married, Separated, Widowed
occupation	categorical	?, Adm-clerical, Armed-Forces, Craft-repair, Exec-managerial, Farming-fishing, Handlers-cleaners, Machine-op-inspct, Other-service, Priv-house-serv, Prof-specialty, Protective-serv, Sales, Tech-support, Transport-moving
relationship	categorical	Husband, Not-in-family, Other-relative, Own-child, Unmarried, Wife
race	categorical	Amer-Indian-Eskimo, Asian-Pac-Islander, Black, Other, White
sex	categorical	Female, Male
capital_gain	integer	any
capital_loss	integer	any
hours_per_week	integer	any

Data onboarding

1. Import sensitive data
2. Validate schema
3. Estimate marginals



Data onboarding

1. Import sensitive data
2. Validate schema
3. Estimate marginals
4. Generate synthetic data

census ✓ Ready

INFO & SOURCE | SCHEMA | ACCESS RULES | SYNTHETIC DATA

DOWNLOAD SYNTHETIC DATA

age	workclass	fnlwgt	education	education_num	marital_status	occupation	relationship
29	Self-emp-inc	349673	Some-college	13	Married-civ-spouse	Sales	Husband
27	Private	188503	Some-college	8	Married-civ-spouse	Exec-managerial	Husband
65	Private	28378	Bachelors	14	Never-married	Prof-specialty	Not-in-family
42	Private	111973	5th-6th	9	Never-married	Prof-specialty	Husband
57	Private	139883	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband
45	Private	136226	10th	10	Married-civ-spouse	Craft-repair	Husband
55	Private	148310	Bachelors	10	Married-civ-spouse	Prof-specialty	Not-in-family
66	Private	54801	Some-college	8	Never-married	Transport-moving	Husband
55	Private	177423	Bachelors	8	Married-civ-spouse	Prof-specialty	Husband
48	Private	126997	HS-grad	7	Married-civ-spouse	Machine-op-inspct	Not-in-family
73	Local-gov	183191	Prof-school	8	Divorced	Sales	Husband
36	Local-gov	180676	HS-grad	2	Divorced	Transport-moving	Not-in-family
38	Private	121979	HS-grad	12	Married-civ-spouse	Prof-specialty	Husband

Data job description

Data job description

1. Connect to the server
2. Get the dataset symbolic representation

```python
from sarus import Client

client = Client(
    url="http://sarus.tech:5000",
    email="datascientist@company.com",
    password="password",
)

dataset = client.dataset(slugname="census")
```

Data job description

1. Connect to the server
2. Get the dataset symbolic representation
3. Describe derived datasets

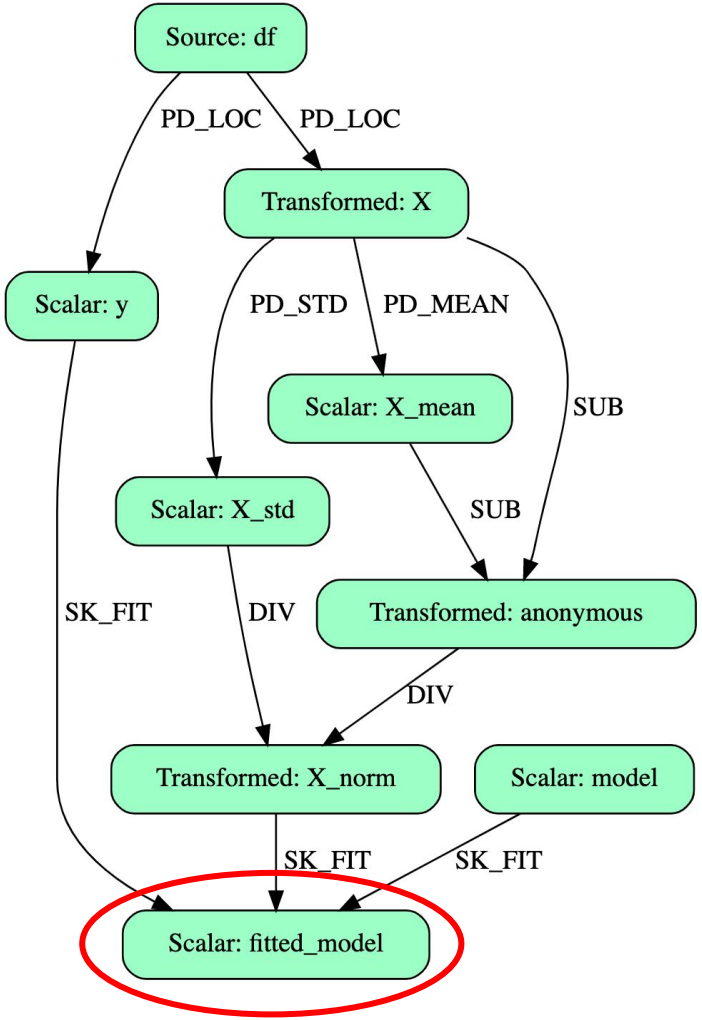
→ Don't reinvent the wheel

→ Stubs registering the pipeline

```
from sarus.sklearn.svm import SVC

df = dataset.as_pandas()
X = df.loc[:, ["age", "education_num"]]
y = df.income

X_mean = X.mean()
X_std = X.std()
X_norm = (X - X_mean) / X_std

model = SVC()
fitted_model = model.fit(X=X_norm, y=y)
```

Data job description

```python
from sarus.sklearn.svm import SVC

df = dataset.as_pandas()
X = df.loc[:, ["age", "education_num"]]
y = df.income

X_mean = X.mean()
X_std = X.std()
X_norm = (X - X_mean) / X_std

model = SVC()
fitted_model = model.fit(X=X_norm, y=y)
```



Compilation

Differential privacy

Notion of adjacent datasets

$$\mathcal{D} = \{\text{data}, \qquad \}$$

$$\mathcal{D}' = \{\text{data},$$  $$\}$$

Differential privacy

Differentially private mechanism

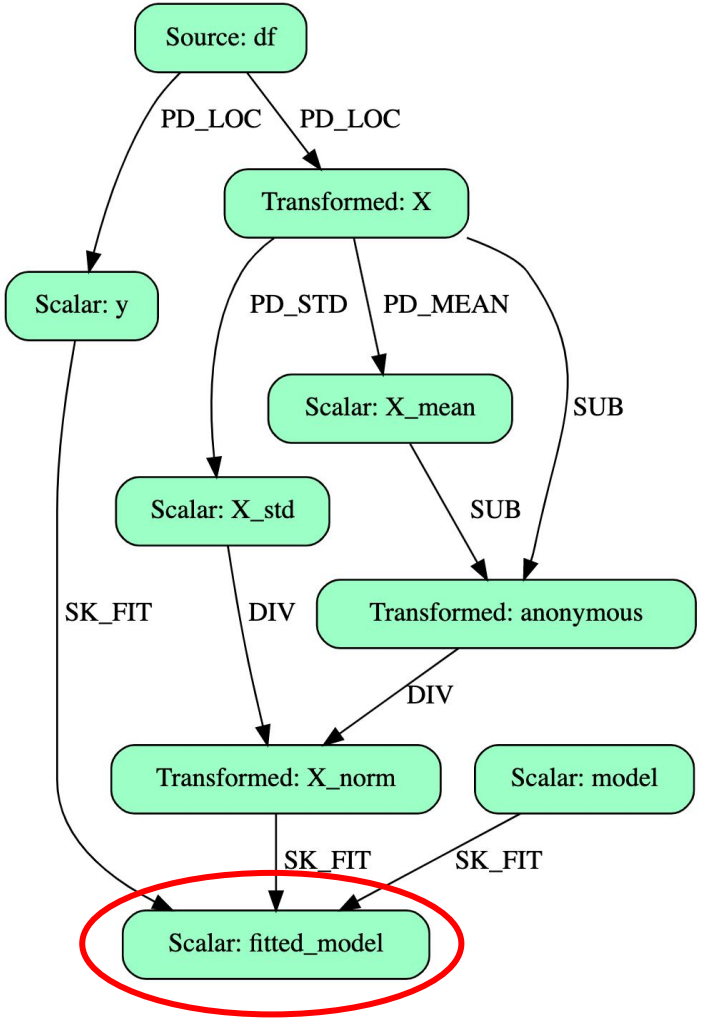
$$\Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{S}] \leq e^{\epsilon} \Pr[\mathcal{A}(\mathcal{D}') \in \mathcal{S}] + \delta$$

Identify the protected entity

1. Protected Entity (PE) identified during onboarding
2. A dataset is **PE preserving** if each row is linked to ***at most one PE***
    a. There is a natural adjacency notion (add or remove 1 PE)
    b. Every other row is considered public
3. Transformations are **PE preserving** if they preserve this property
    a. e.g. column selection is PE preserving
    b. e.g. aggregates (mean, std) are not PE preserving as they mix informations from many PE together

How to compile ?

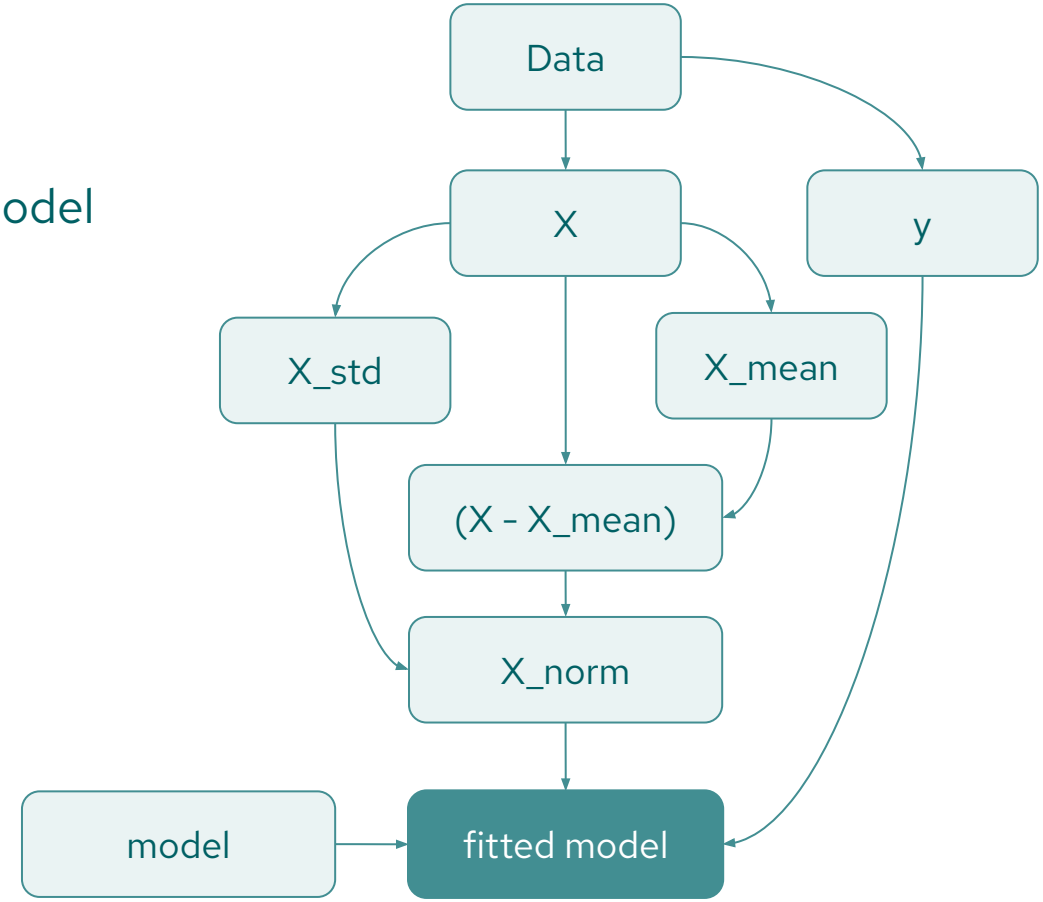
```
from sarus.sklearn.svm import SVC

df = dataset.as_pandas()
X = df.loc[:, ["age", "education_num"]]
y = df.income

X_mean = X.mean()
X_std = X.std()
X_norm = (X - X_mean) / X_std

model = SVC()
fitted_model = model.fit(X=X_norm, y=y)
```



Application

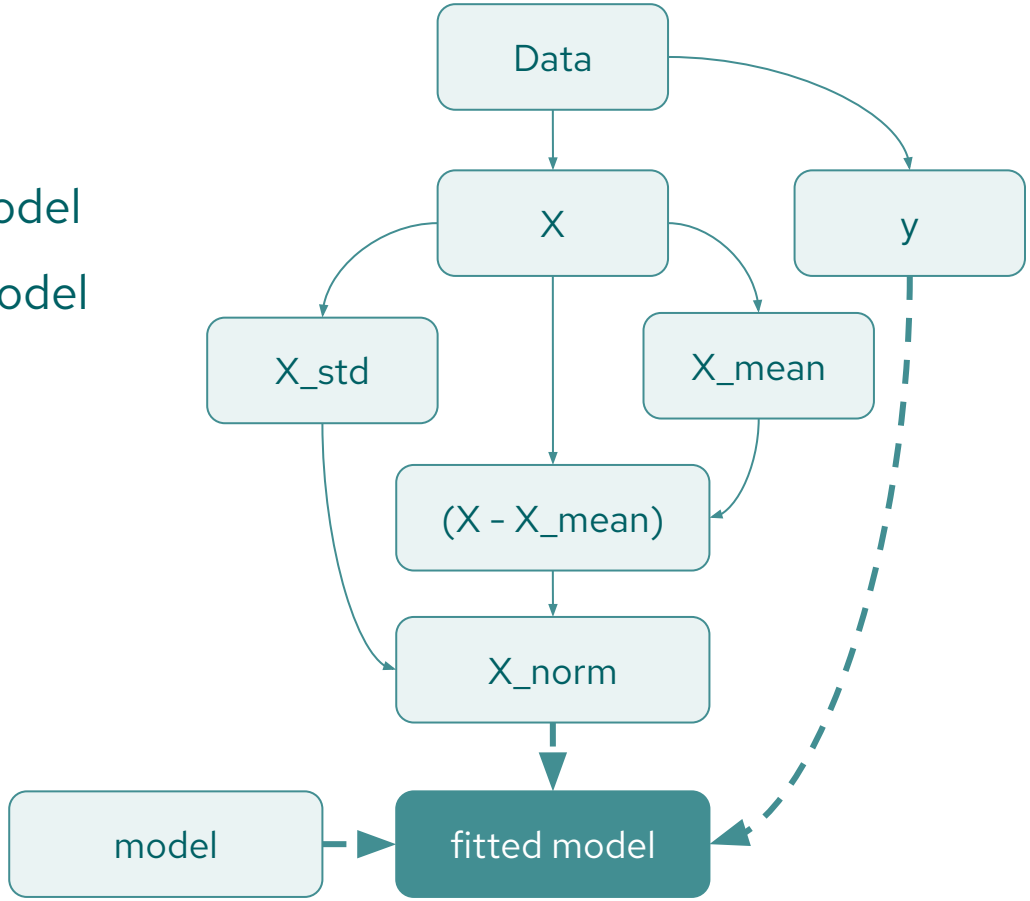
Application

1. Compilation of a fitted ML model



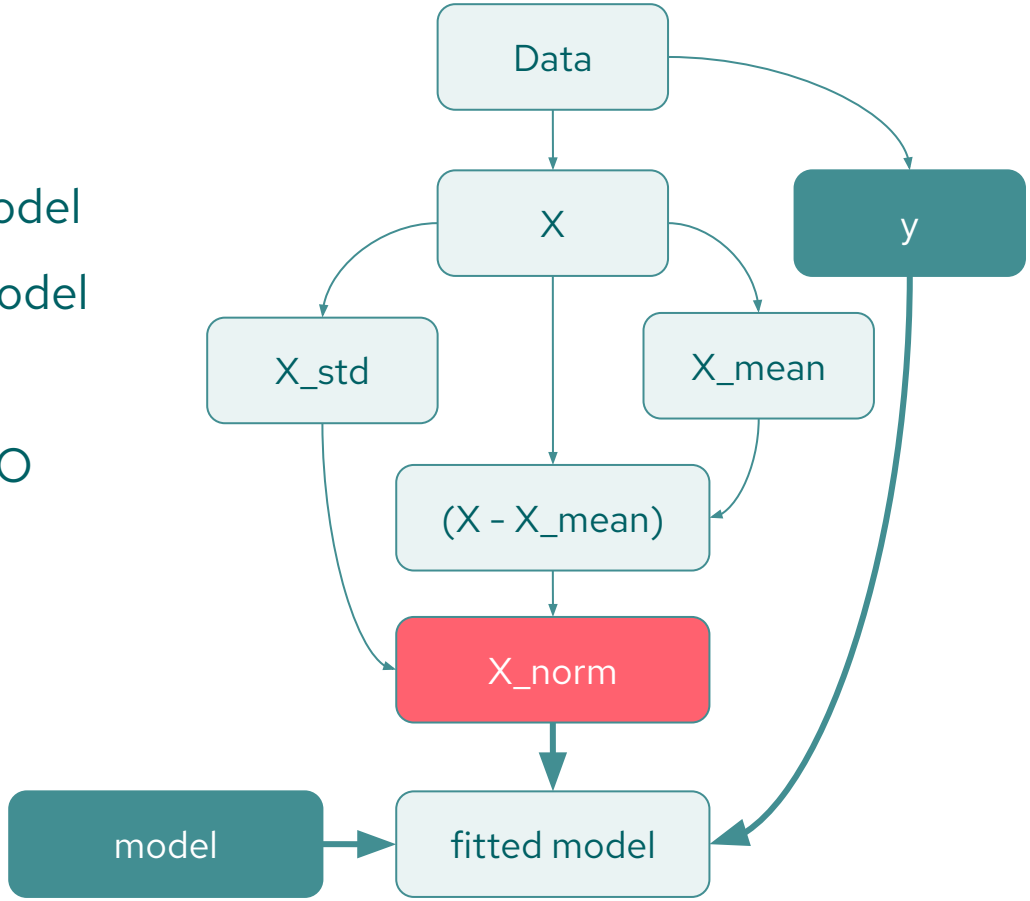
Application

1. Compilation of a fitted ML model
2. Apply DP-SDG to train the model



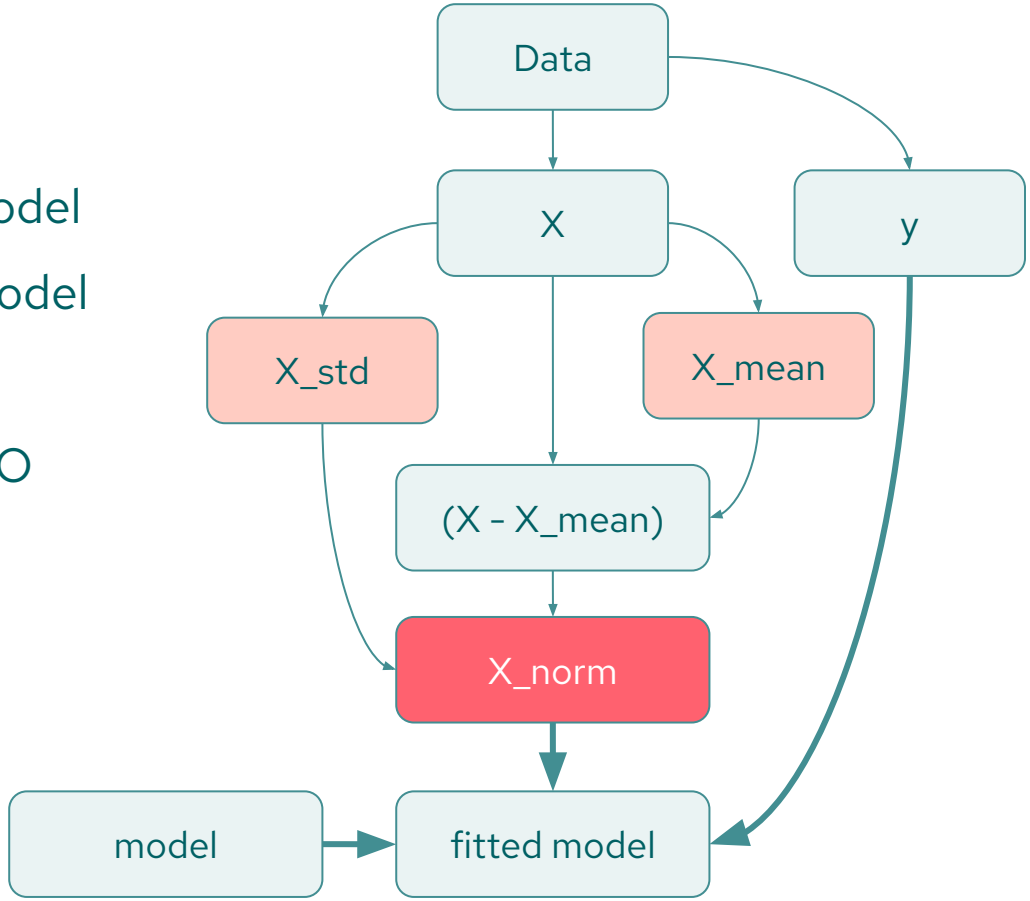
Application

1. Compilation of a fitted ML model
2. Apply DP-SDG to train the model
   a. DP alternative OK
   b. Inputs PE preserving ? NO



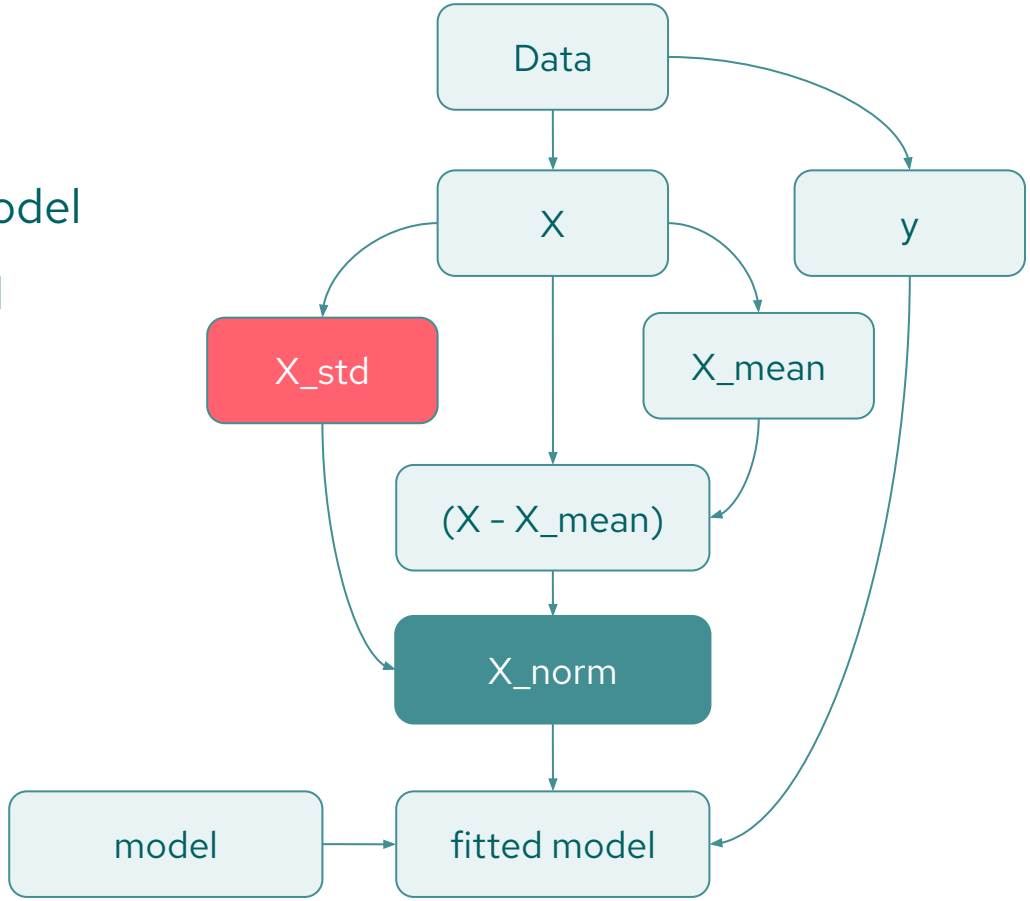
Application

1. Compilation of a fitted ML model
2. Apply DP-SDG to train the model
   a. DP alternative OK
   b. Inputs PE preserving ? NO



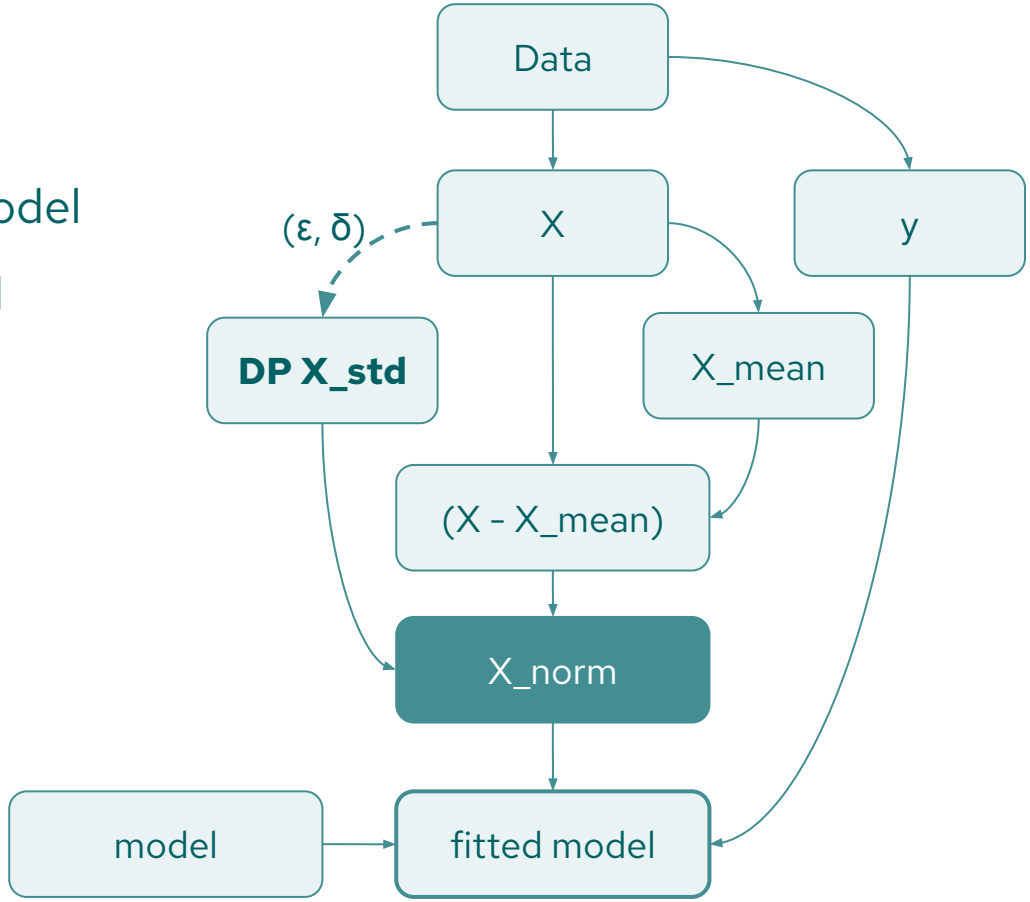
Application

1. Compilation of a fitted ML model
2. Compile PE preserving `X_std`



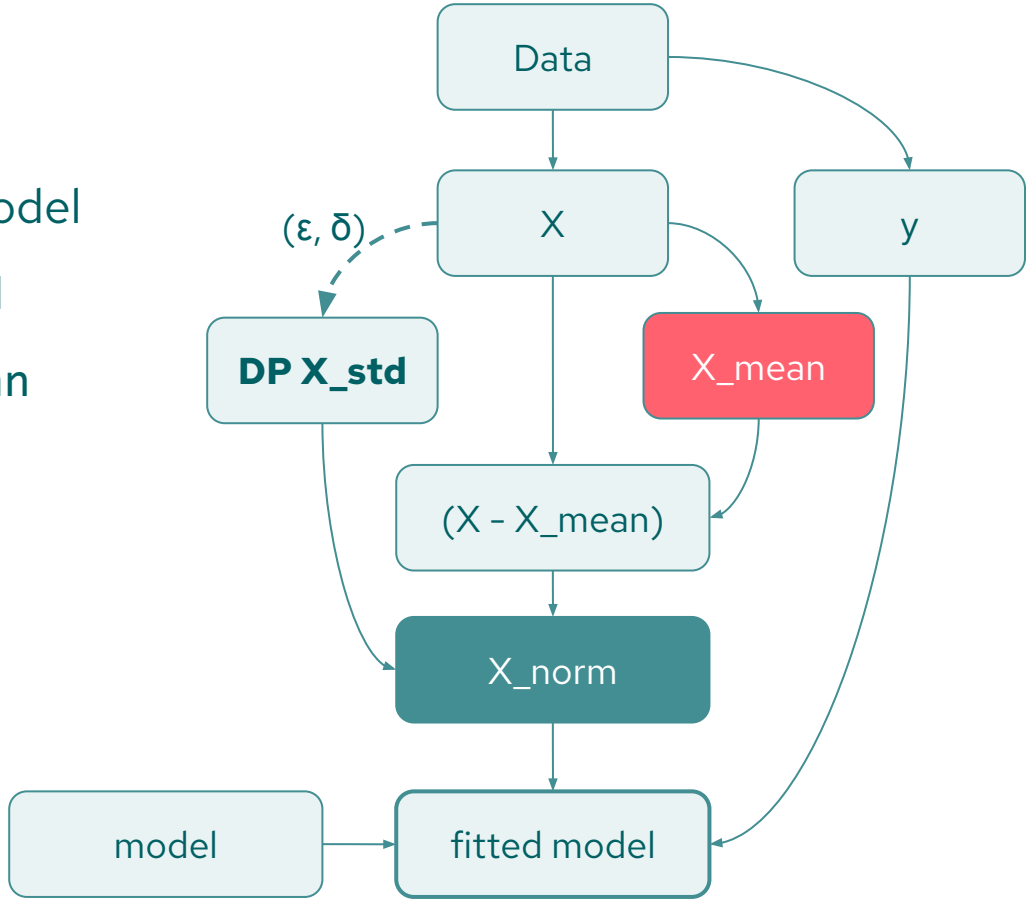
Application

1. Compilation of a fitted ML model
2. Compile PE preserving `X_std`



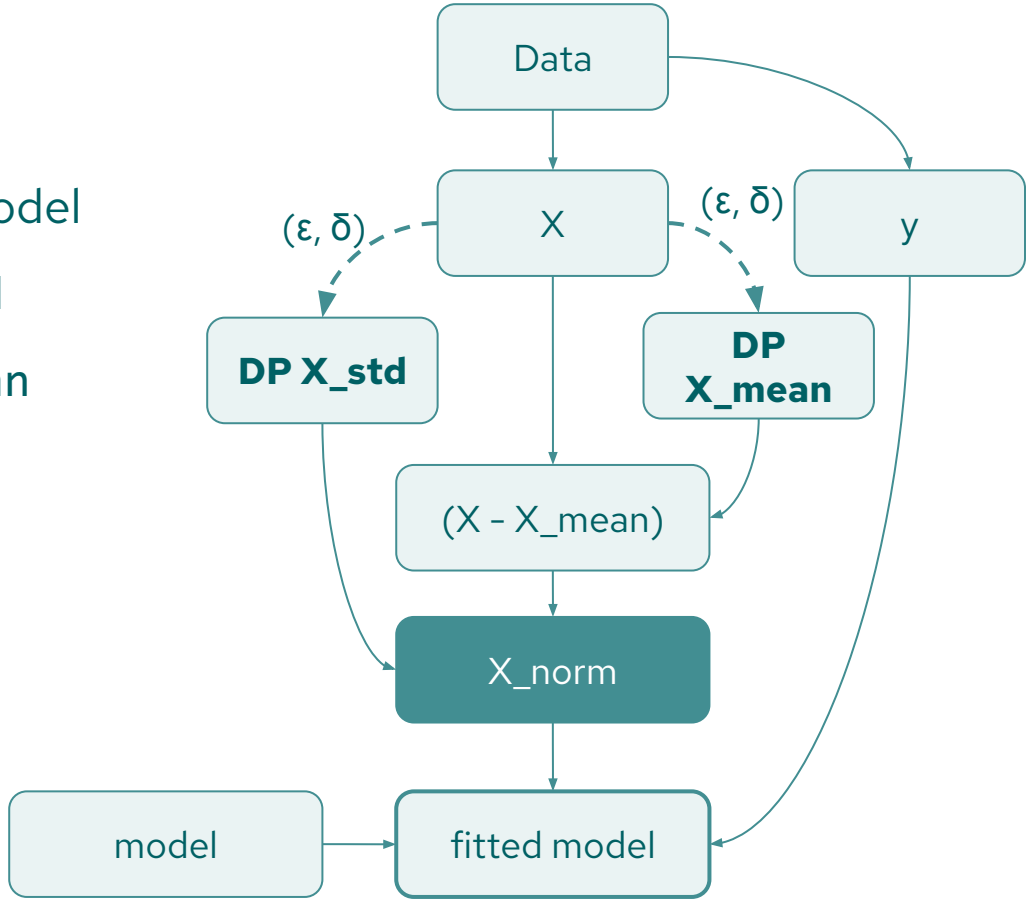
Application

1. Compilation of a fitted ML model
2. Compile PE preserving `X_std`
3. Compile PE preserving `X_mean`



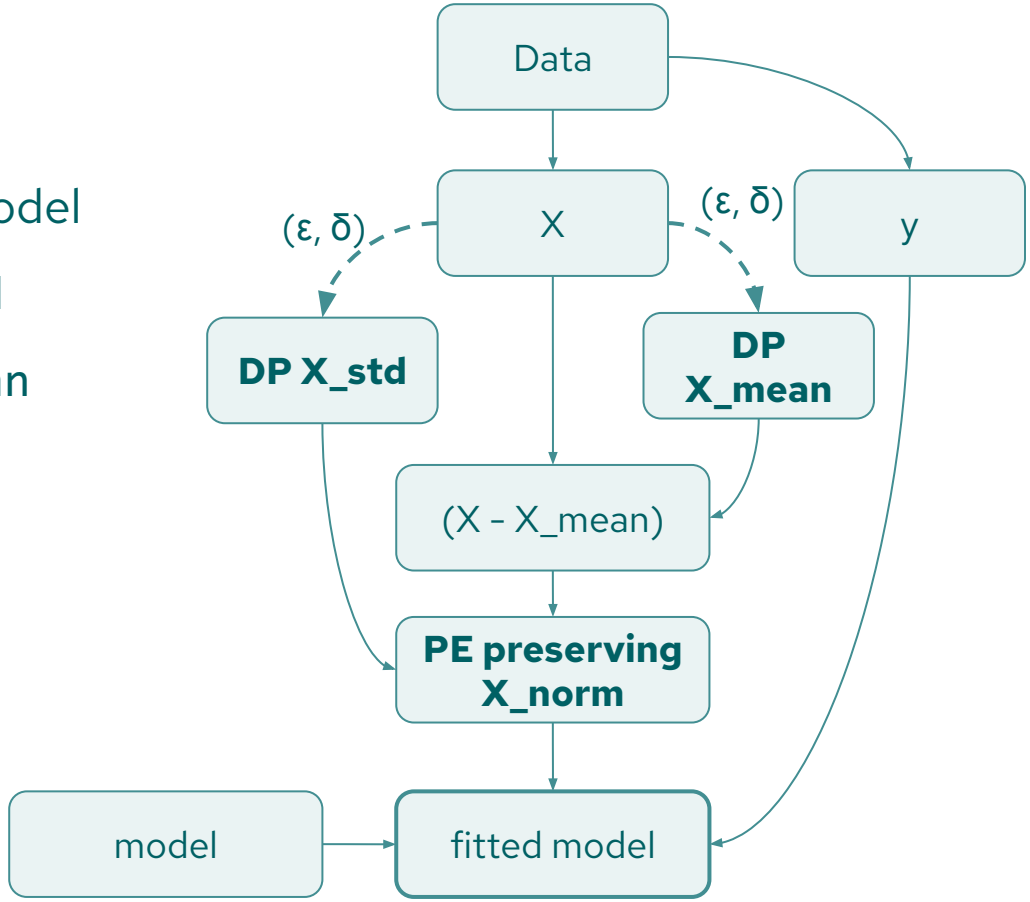
Application

1. Compilation of a fitted ML model
2. Compile PE preserving `X_std`
3. Compile PE preserving `X_mean`



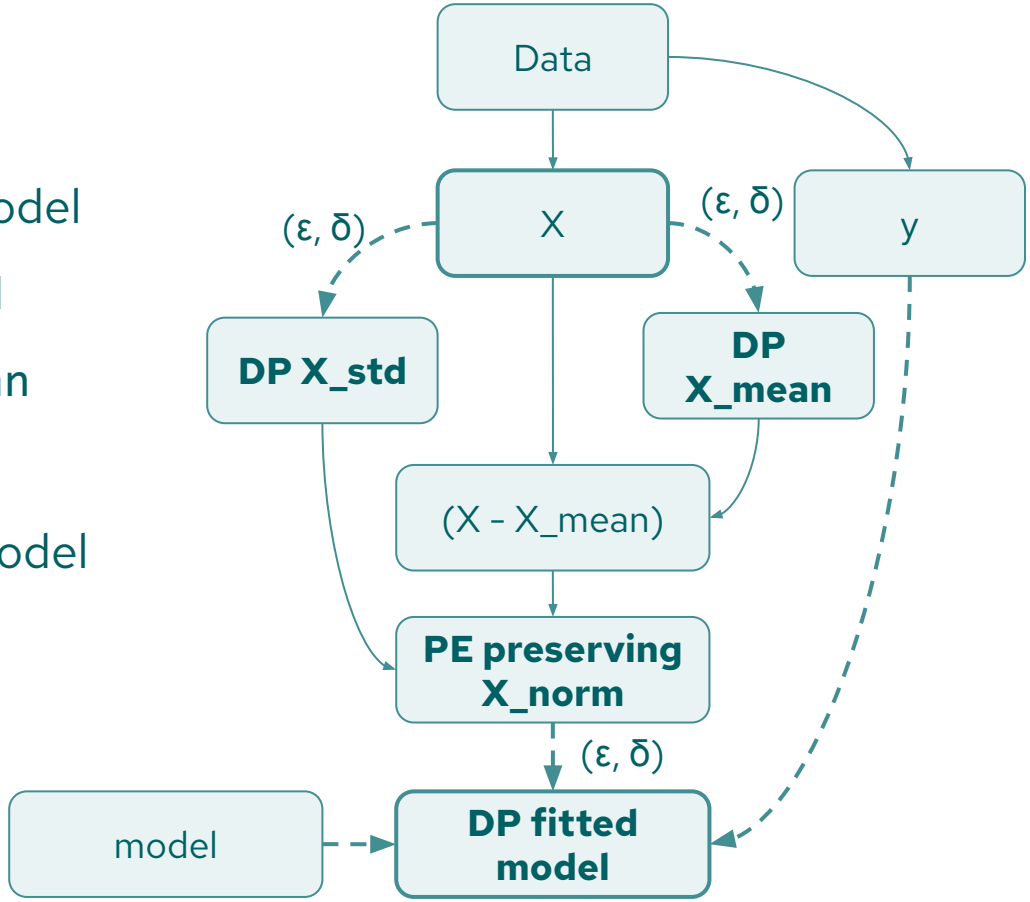
Application

1. Compilation of a fitted ML model
2. Compile PE preserving `X_std`
3. Compile PE preserving `X_mean`
4. `X_norm` is PE preserving



Application

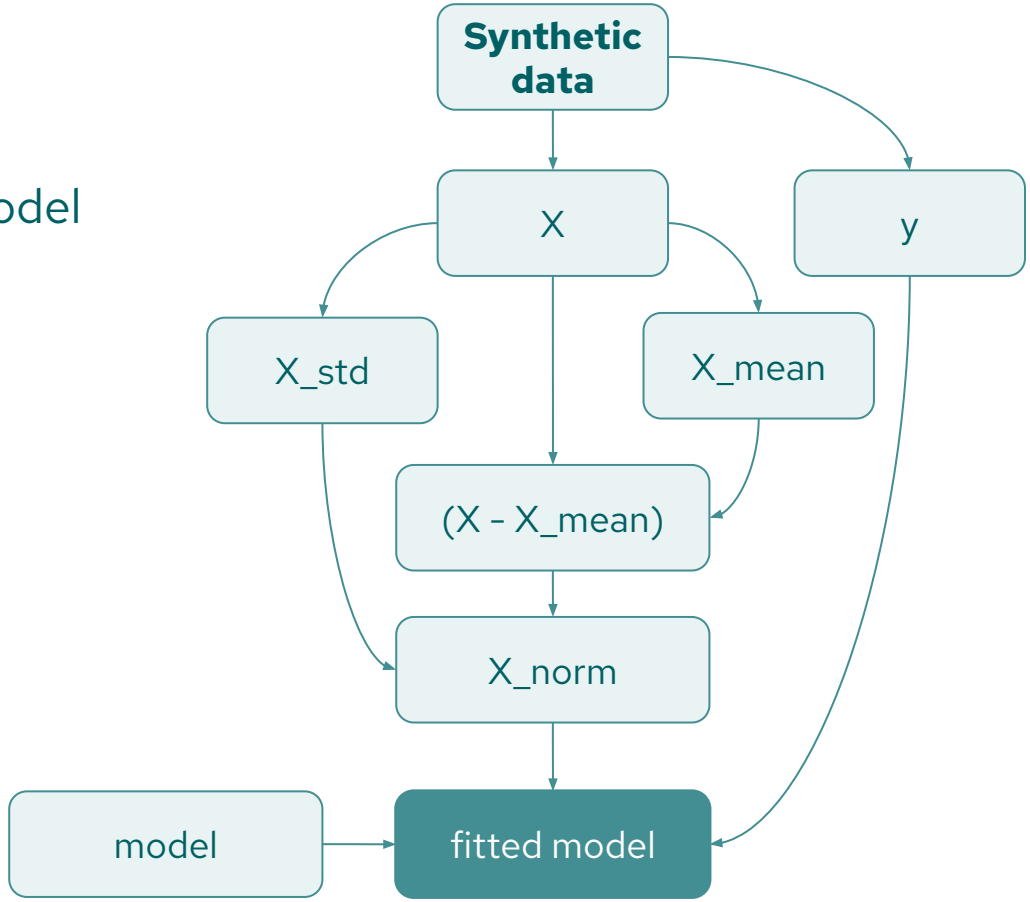
1. Compilation of a fitted ML model
2. Compile PE preserving `X_std`
3. Compile PE preserving `X_mean`
4. `X_norm` is PE preserving
5. Apply DP-SDG to train the model



Application

1. Compilation of a fitted ML model

#joker: synthetic data alternative



Thank you for your attention!

Johan Leduc
jl@sarus.tech

Nicolas Grislain
ng@sarus.tech