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

<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>integer</td>
<td>any</td>
</tr>
<tr>
<td>fnlwgt</td>
<td>integer</td>
<td>any</td>
</tr>
<tr>
<td>education</td>
<td>categorical</td>
<td>10th, 11th, 12th, 1st-4th, 5th-6th, 7th-8th, 9th, Assoc-acdm, Assoc-voc, Bachelors, Doctorate, HS-grad, Masters, Preschool, Prof-school, Some-college</td>
</tr>
<tr>
<td>education_num</td>
<td>integer</td>
<td>any</td>
</tr>
<tr>
<td>marital_status</td>
<td>categorical</td>
<td>Divorced, Married-AF-spouse, Married-civ-spouse, Married-spouse-absent, Never-married, Separated, Widowed</td>
</tr>
<tr>
<td>occupation</td>
<td>categorical</td>
<td>?; Adm-clerical, Armed-Forces, Craft-repair, Exec-managerial, Farming-fishing, Handlers-cleaners, Machine-op-inspect, Other-service, Pr-House-serv, Prof-specialty, Protective-serv, Sales, Tech-support, Transport-moving</td>
</tr>
<tr>
<td>relationship</td>
<td>categorical</td>
<td>Husband, Not-in-family, Other-relative, Own-child, Unmarried, Wife</td>
</tr>
<tr>
<td>race</td>
<td>categorical</td>
<td>Amer-Indian-Eskimo, Asian-Pac-Islander, Black, Other, White</td>
</tr>
<tr>
<td>sex</td>
<td>categorical</td>
<td>Female, Male</td>
</tr>
<tr>
<td>capital_gain</td>
<td>integer</td>
<td>any</td>
</tr>
<tr>
<td>capital_loss</td>
<td>integer</td>
<td>any</td>
</tr>
<tr>
<td>hours_per_week</td>
<td>integer</td>
<td>any</td>
</tr>
</tbody>
</table>
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
Data job description
Data job description

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

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

```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)
```
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

\[ \mathcal{D} = \{ \text{data}, \} \]

\[ \mathcal{D}' = \{ \text{data}, \} \]
Differential privacy

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

```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)
```
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
1. Compilation of a fitted ML model
2. Compile PE preserving $X_{\text{std}}$
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 $X_{\text{std}}$
3. Compile PE preserving $X_{\text{mean}}$
Application

1. Compilation of a fitted ML model
2. Compile PE preserving $X_{\text{std}}$
3. Compile PE preserving $X_{\text{mean}}$

Diagram:
- Data
- $X$
- $y$
- $(\varepsilon, \delta)$
- $X_{\text{mean}}$
- $X_{\text{std}}$
- $(X - X_{\text{mean}})$
- $X_{\text{norm}}$
- Model
- Fitted model
Application

1. Compilation of a fitted ML model
2. Compile PE preserving $X_{\text{std}}$
3. Compile PE preserving $X_{\text{mean}}$
4. $X_{\text{norm}}$ is PE preserving
Application

1. Compilation of a fitted ML model
2. Compile PE preserving $X_{\text{std}}$
3. Compile PE preserving $X_{\text{mean}}$
4. $X_{\text{norm}}$ 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