Security Information System

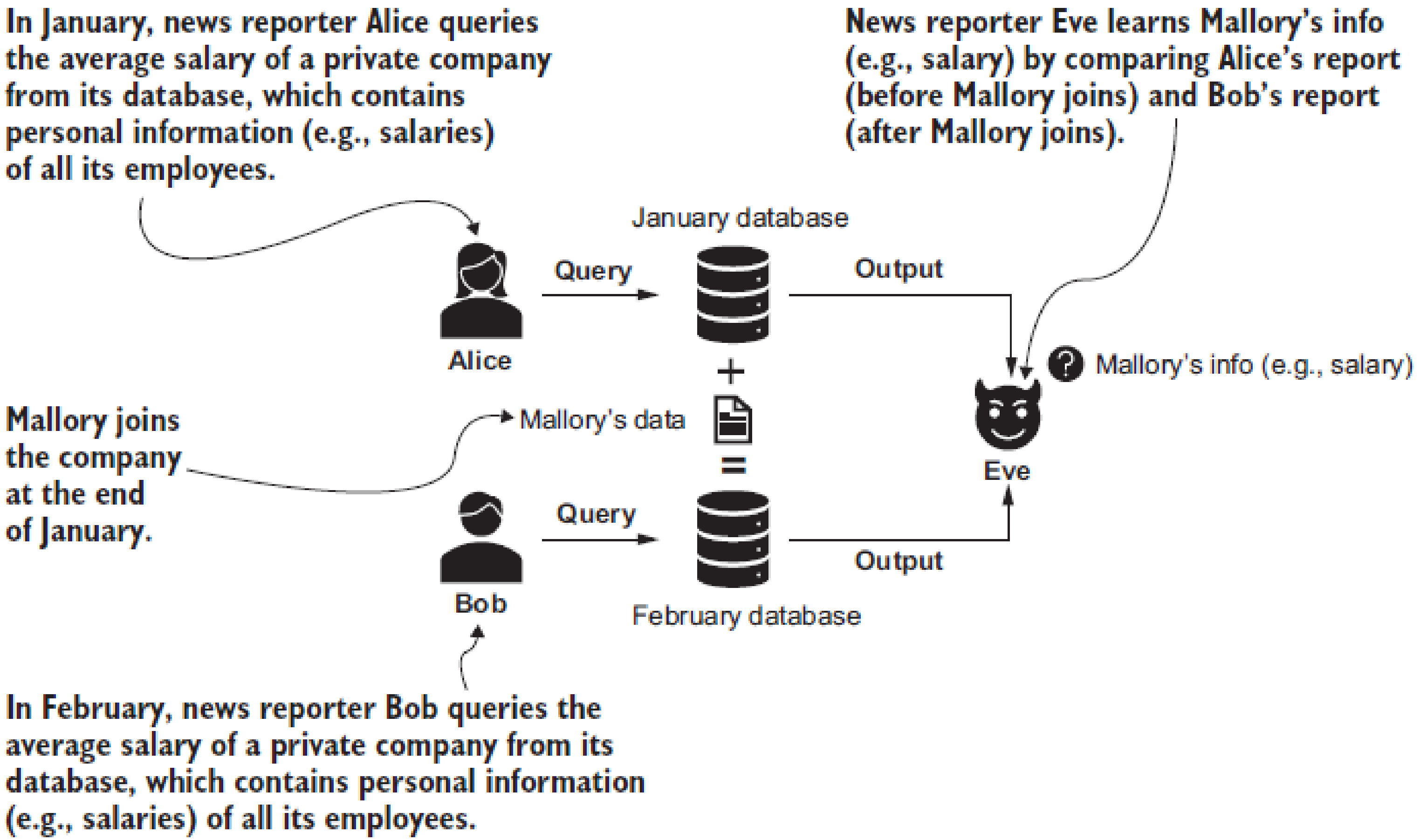
Chapter 7:
Differential Privacy

Outline

- Introduction
- Differential privacy for machine learning

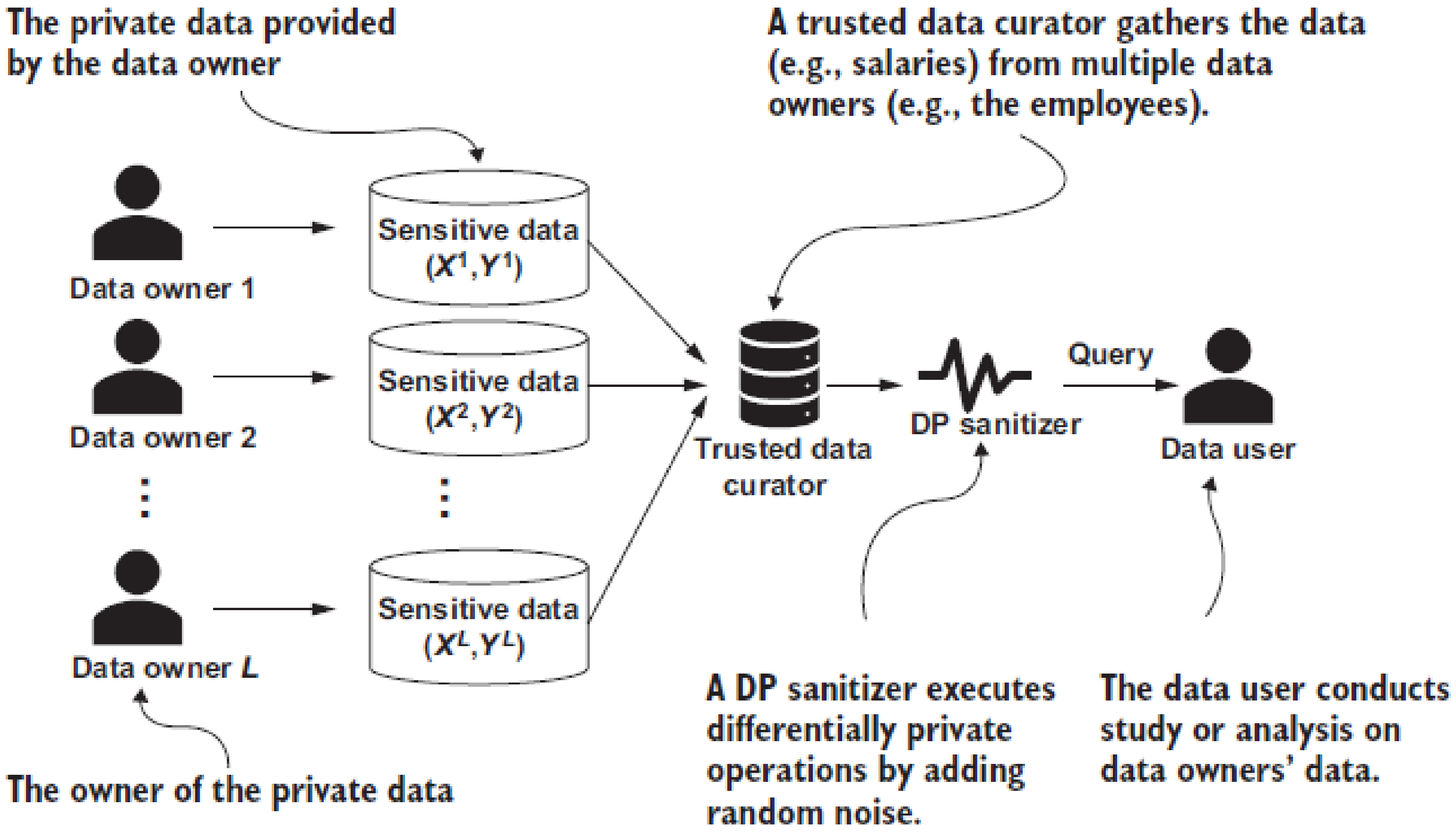
Introduction

- Problem



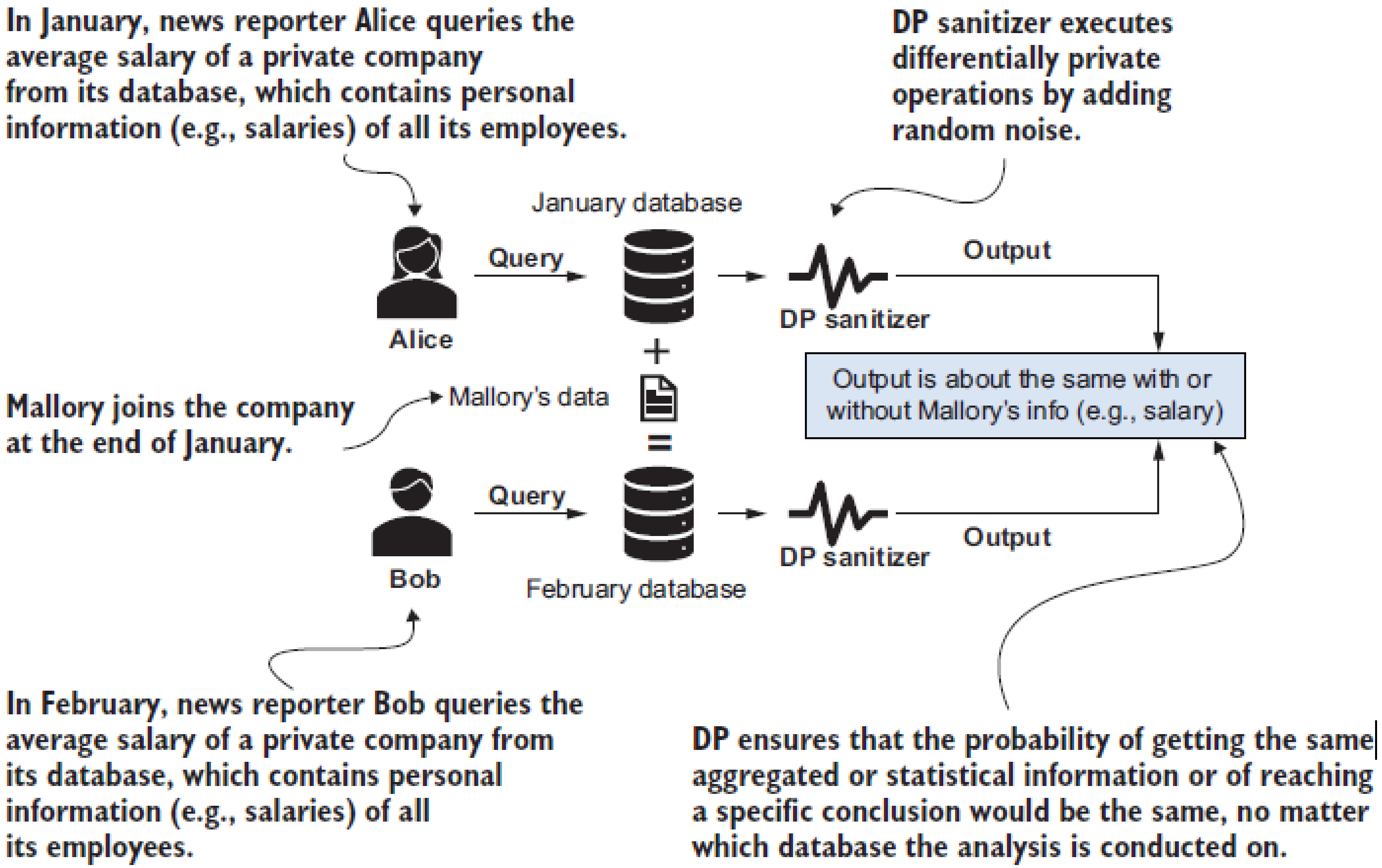
Introduction

- Problem



Introduction

- Objective: Using DP to protect personal data



Introduction

- Example: Mallory – Salary? Random noise $\triangle f/\epsilon$?
  - Query:
    - Count employee
      - 1
    - Average Salary

$$\Delta f = \frac{S_{\max} - S_{\min}}{N}$$

    - Laplace Noise
      - Count Employee

$$\text{Noise}_{\text{count}} \sim \text{Laplace}(0, \frac{1}{\epsilon})$$

      - Average Salary

$$\text{Noise}_{\text{average}} \sim \text{Laplace}\left(0, \frac{S_{\max} - S_{\min}}{\epsilon N}\right)$$

The January salary database

January

Employee (100)	Salary
CEO - Jack	$290,000
CFO - Tim	$250,000
CTO - Mike	$245,000
COO - Peter	$240,000
CMO - Scott	$200,000
95 × Other Employees	$45,000
AVG	$55,000

The average salary of the private company in January

Mallory joins the company for a salary of $156,000.

Mallory joins the company

The February salary database

February

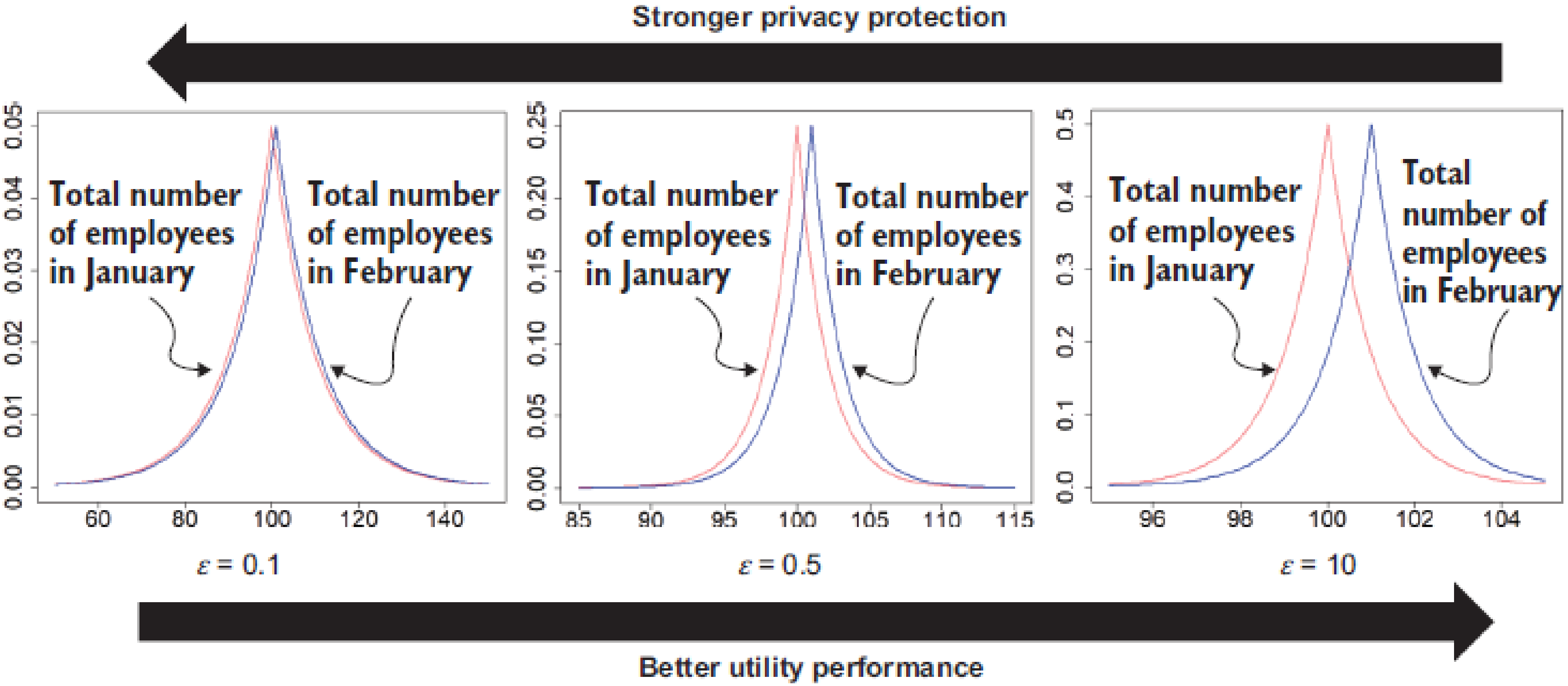
Employee (101)	Salary
CEO - Jack	$290,000
CFO - Tim	$250,000
CTO - Mike	$245,000
COO - Peter	$240,000
CMO - Scott	$200,000
Mallory	$156,000
95 × Other Employees	$45,000
AVG	$56,000

The average salary of the private company in February

Introduction

- Example: Add noise

1. Total Number of Employees:
   - Noisy count $= \mathrm{True\ count} + \mathrm{Noise}_{\mathrm{count}}$
2. Average Salary:
   - Noisy average salary $= \mathrm{True\ average\ salary} + \mathrm{Noise}_{\mathrm{average}}$

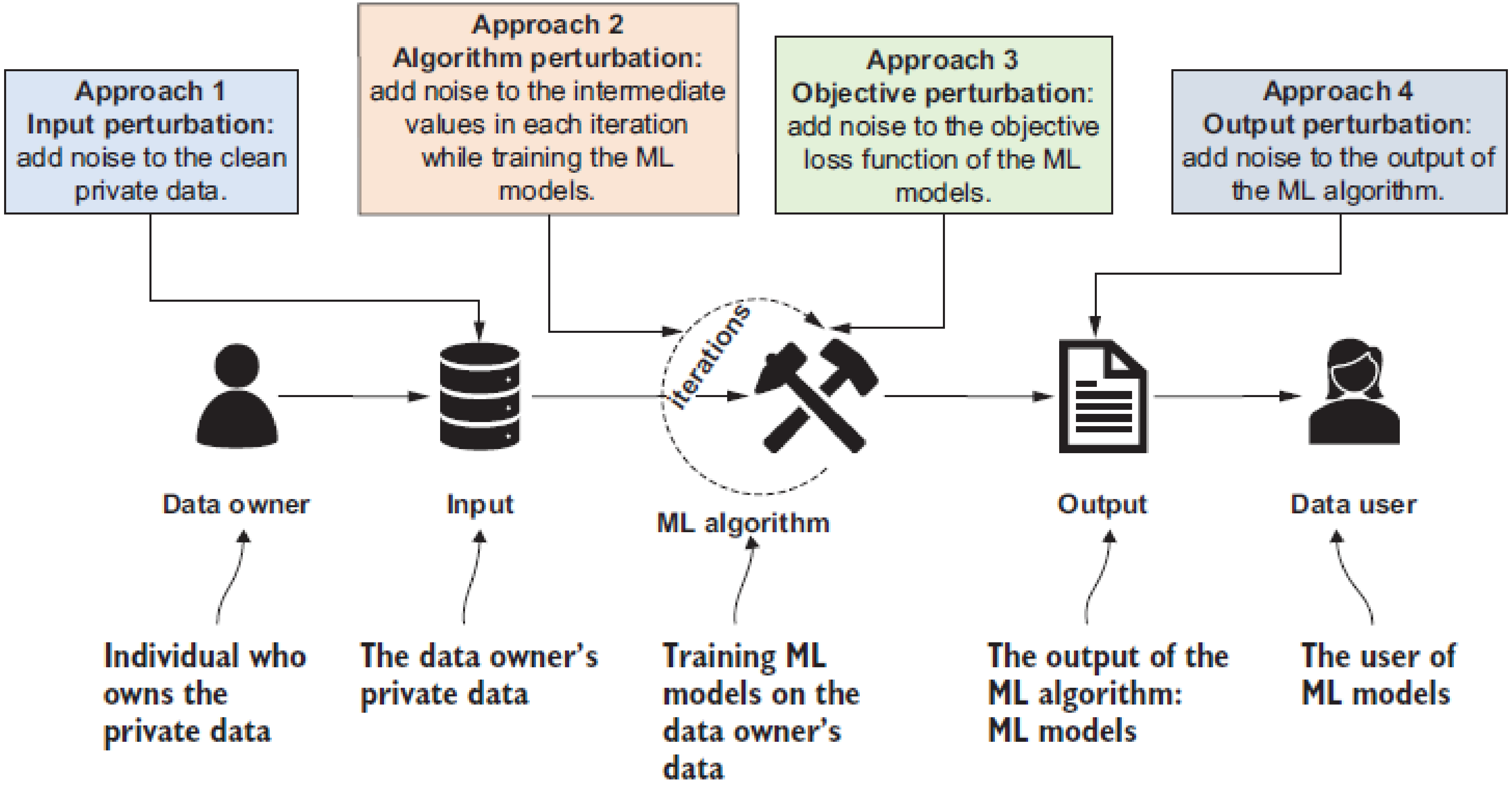


Exercise

- Tính giá trị nhạy cảm cho thuộc tính tuổi ở bảng dữ liệu bên.
- Tính giá trị nhiễu với $\epsilon = 0.1$
- Tính giá trị nhiễu với $\epsilon = 0.01$

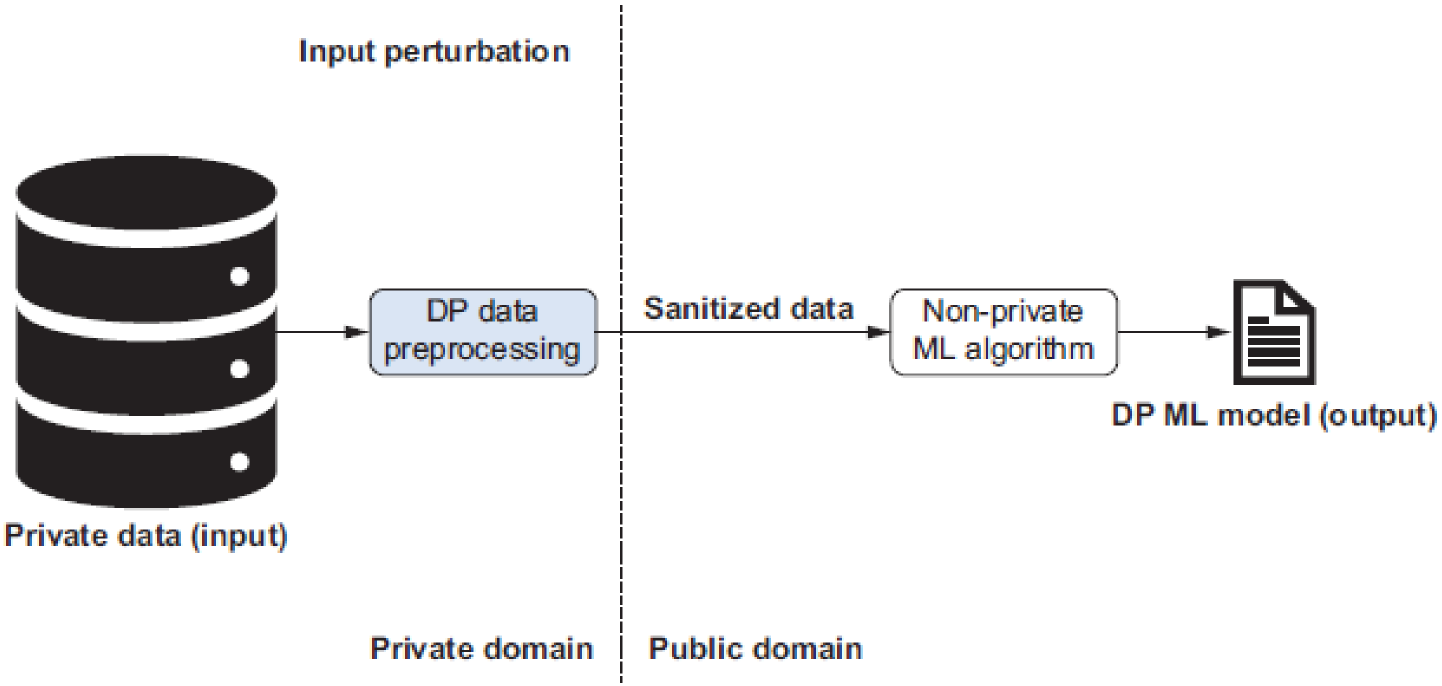
SSNumber	Age	ZipCode	Condition
1234-12-1234	21	23058	heart disease
2345-23-2345	24	23059	heart disease
3456-34-3456	26	23060	viral infection
4567-45-4567	27	23061	viral infection
9012-90-9012	32	23058	kidney stone
0123-12-0123	34	23059	kidney stone
4321-43-4321	35	23060	aids
5432-54-5432	38	23061	aids
5678-56-5678	43	23058	kidney stone
6789-67-6789	43	23059	heart disease
7890-78-7890	47	23060	viral infection
8901-89-8901	49	23061	viral infection

Differential privacy for machine learning



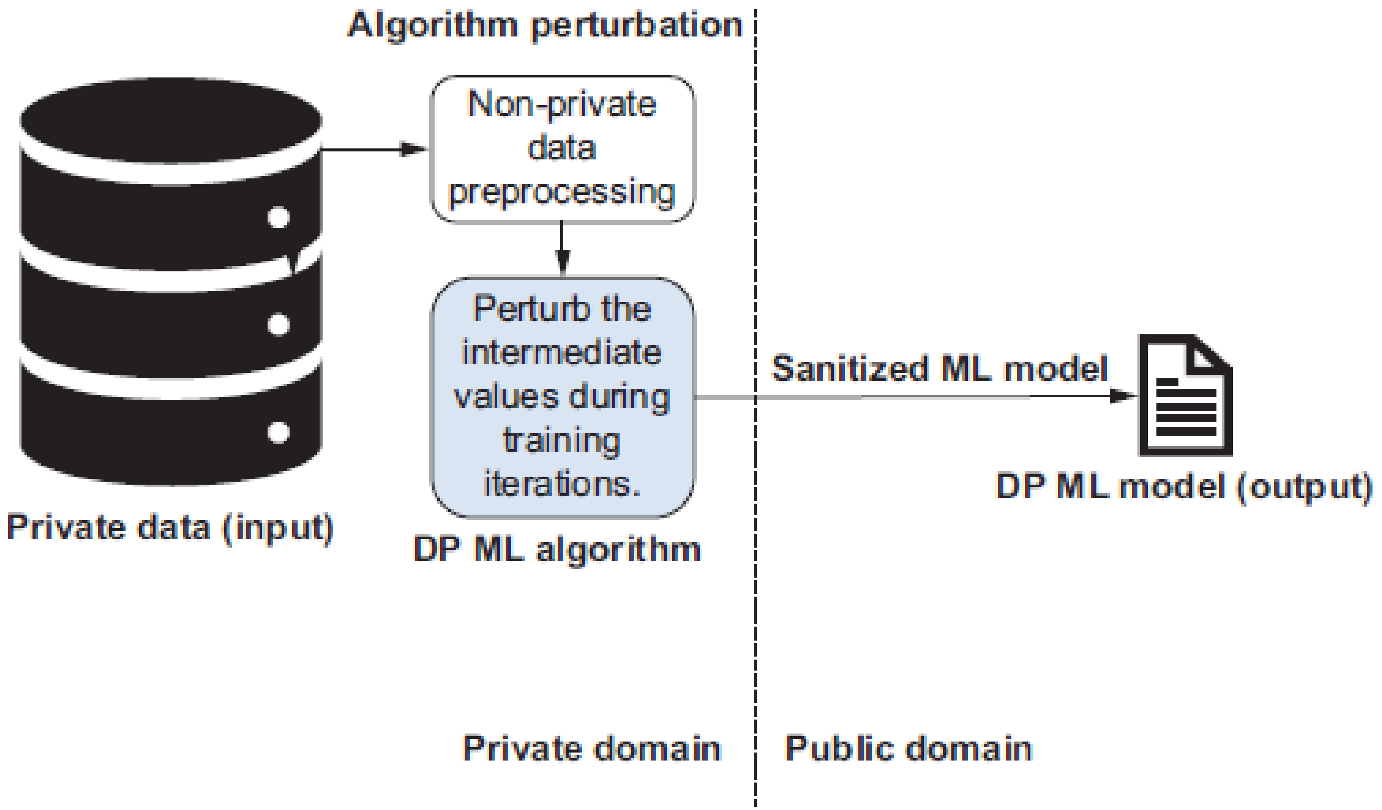
Differential privacy for machine learning

- Input perturbation



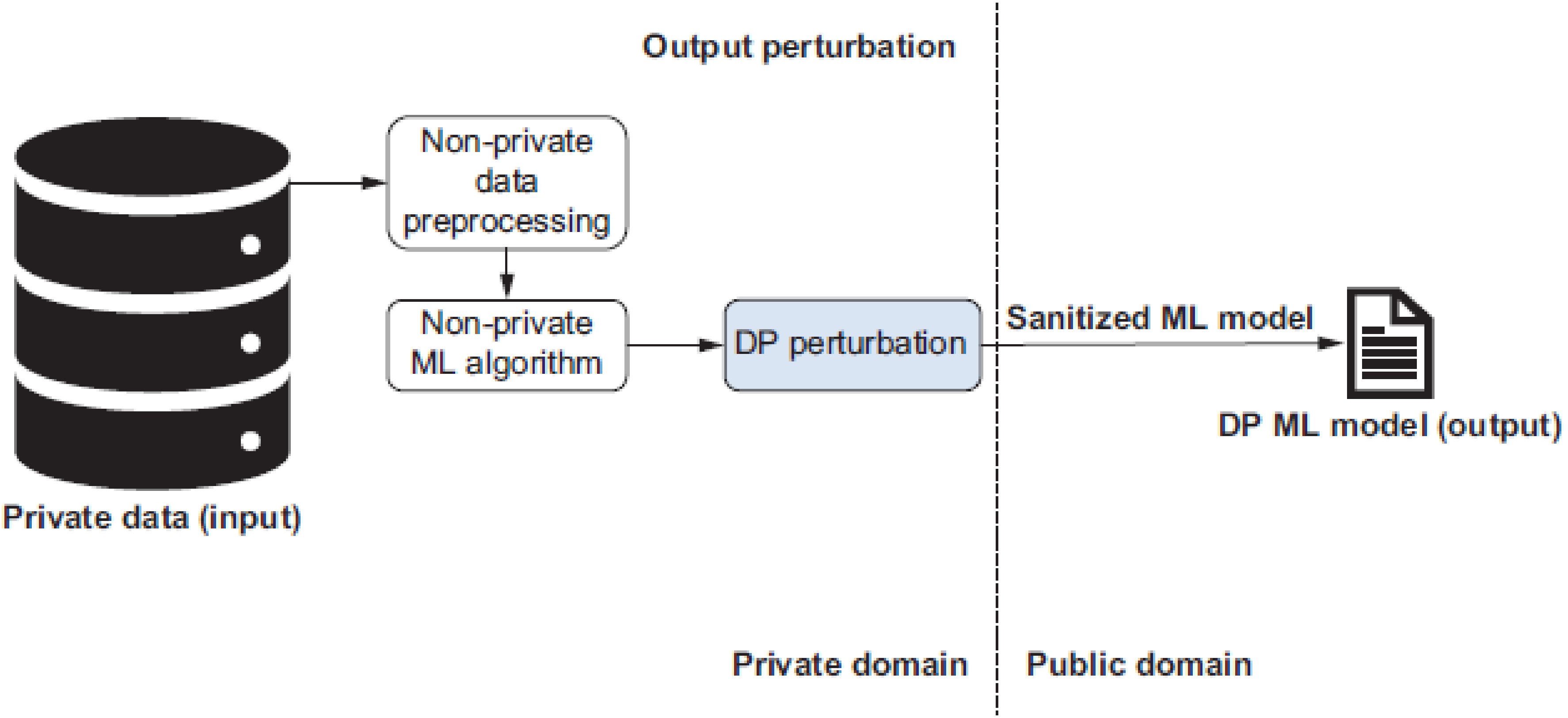
Differential privacy for machine learning

- Algorithm perturbation



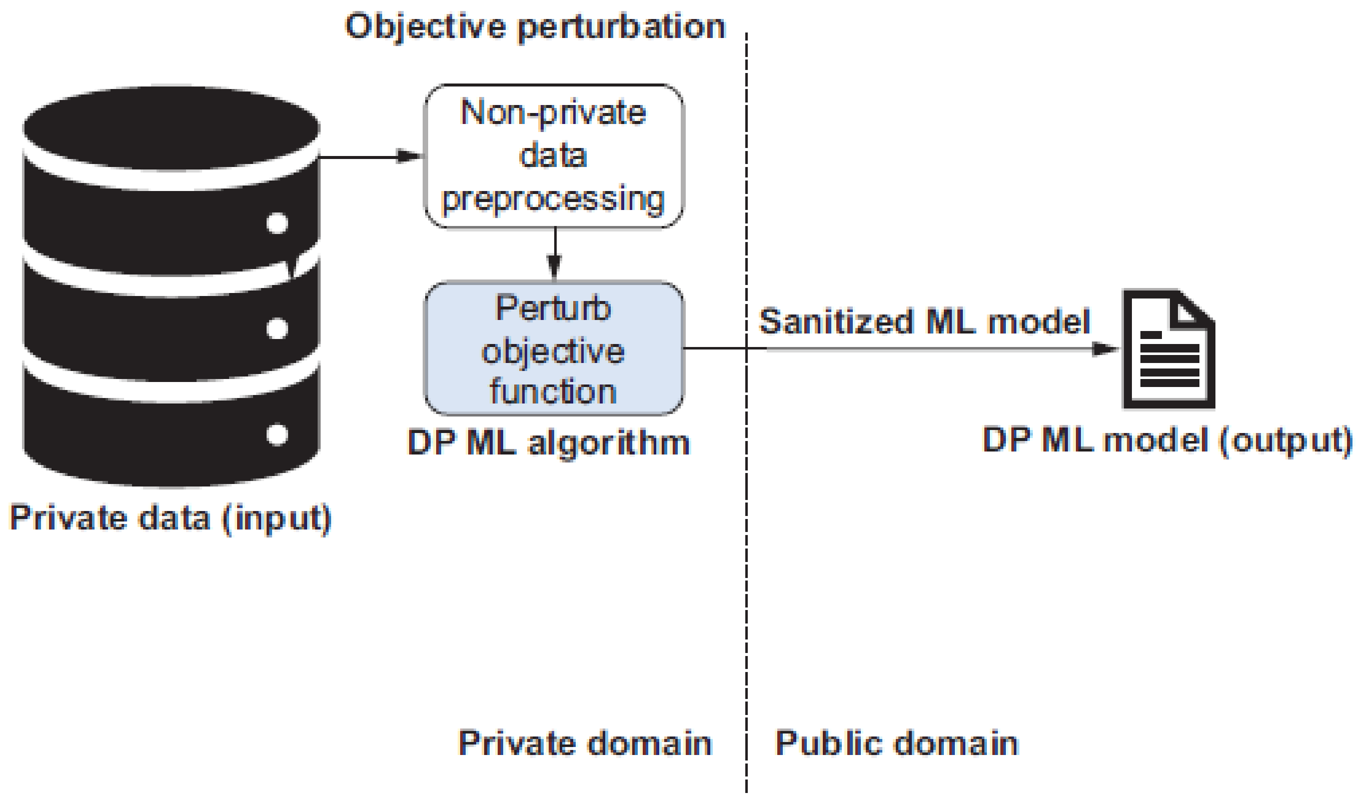
Differential privacy for machine learning

- Output perturbation



Differential privacy for machine learning

- Objective perturbation



Differential privacy for machine learning

- Differentially private naive Bayes classification - Output perturbation
  - NAIVE BAYES CLASSIFICATION

$$P(C_j|X) = \frac{P(X|C_j) \times P(C_j)}{P(X)}$$

Since $P(X)$ is the same for all classes, it is sufficient to find the class with the maximum $P(X| \ C_j) \cdot P(C_j)$. Assuming the independence of features, that class is equal to $P(C_j) \cdot \prod_{i=1}^{n} P(F_i = x_i|C_j)$. Hence, the probability of assigning C_j to the given instance X is proportional to $P(C_1) \cdot \prod_{i=1}^{3} P(F_i = x_i|C_1)$.

Differential privacy for machine learning

- Differentially private naive Bayes classification
  - Dataset
  - X = (Age = Young, Income = Medium, Gender = Female).

Number	Age	Income	Gender	Missed payment (yes or no)
1	Young	Low	Male	Yes
2	Young	High	Female	Yes
3	Medium	High	Male	No
4	Old	Medium	Male	No

Differential privacy for machine learning

- Differentially private naive Bayes classification

Input: the user-specified privacy budget ϵ, the training data X (n-dimensional vector), the set of n feature names F, and the set k of classes C

Input: privacy budget ϵ; $X = [x_1, x_2, \ldots, x_n]$; $F = [F_1, F_2, \ldots, F_n]$; $C = [C_1, C_2, \ldots, C_k]$.
Output: perturbed $P(F_i = x_i|C_j)$ and $P(C_j)$, $i = 1, 2, \ldots, n$ $j = 1, 2, \ldots, k$.

for *each feature* F_i **do**
  if F_i *is discrete* **then**
    sensitivity $s \leftarrow 1$; — Calculate the sensitivity for discrete features.
    $n'_{ij} = n_{ij} + Lap(0, \frac{s}{\epsilon})$; — Perturb the aggregated data.
    Use n'_{ij} to calculate $P(F_i = x_i|C_j)$; — Calculate the differentially private conditional probability.
  else
    compute sensitivity of mean and variance, s_μ and s_{σ^2}; — Calculate the sensitivity for numerical features.
    $\mu'_i = \mu_i + Lap(0, \frac{s_\mu}{\epsilon})$; — Perturb the aggregated data.
    ${\sigma^2}'_i = \sigma^2_i + Lap(0, \frac{s_{\sigma^2}}{\epsilon})$;
    Use μ'_i and ${\sigma^2}'_i$ to calculate $P(F_i = x_i|C_j)$; — Calculate the differentially private conditional probability.
for *each class* C_j **do**
  sensitivity $s \leftarrow 1$; — Calculate the sensitivity for count.
  count the perturbed total number of samples of class C_j: $n'_j \leftarrow n_j + Lap(0, \frac{s}{\epsilon})$; — Perturb the count.
  Use n'_j to calculate the prior $P(C_j)$; — Calculate the differentially private prior probability.
return perturbed $P(F_i = x_i|C_j)$ and $P(C_j)$, $i = 1, 2, \ldots, n$ $j = 1, 2, \ldots, k$.

Output: the differentially private conditional probability $P(F_i = x_i|C_j)$ and class prior probability $P(C_j)$

Differential privacy for machine learning

- Differentially private naive Bayes classification
  - Implement
    - Load dataset
    - Naive Bayes with no privacy
    - Install IBM Differential Privacy Library, diffprivlib
    - Train a naive Bayes classifier while satisfying DP

Differential privacy for machine learning

- Differentially private k-means clustering - Algorithm perturbation
  - Each sample of the k-means clustering is a d-dimensional point, and assume the k-means algorithm has a predetermined number of running iterations, denoted as t. In each iteration of the k-means algorithm, two values are calculated:
  - The total number of samples of each cluster C_i, denoted as n_i (i.e., the count queries)
  - The sum of the samples of each cluster C_i (to recalculate the centroids), denoted as s_i (i.e., the sum queries)

Differential privacy for machine learning

- Differentially private k-means clustering