Deep Learning Meets Data Privacy

Presenter: Sharika Loganathan

Data Scientist,

University of Pittsburgh

Agenda

- Why Privacy Matters in the Age of AI
- Limitations of Traditional Anonymization
- From Anonymization to Computable Privacy
- Understanding Differential Privacy
- Rethinking the Machine Learning Pipeline
- Differential Privacy in Practice
- Q&A, Feedback, and References

Why Privacy Matters in the Age of Big Data & AI

Massive Data Collection

AI Models Learn from Us

Risk of Re-identification

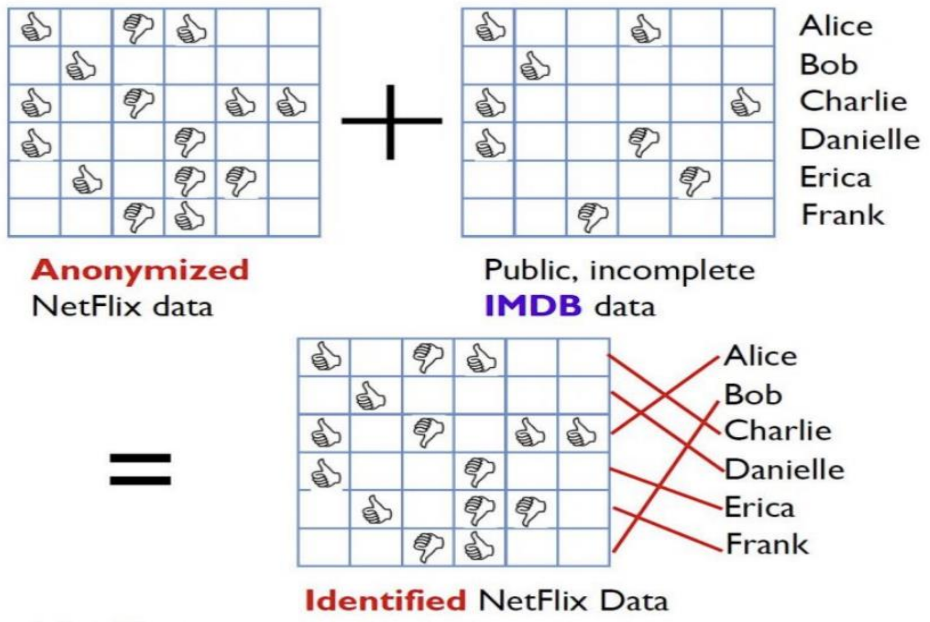
Trust & Transparency

Legal & Ethical Imperatives

Sustainable AI Development

According to Statista, the global data volume was 149 zettabytes in 2024 and is projected to reach 181 zettabytes by the end of 2025

The Problem: Why Traditional Anonymization Fails



Linkage Attacks - Netflix's anonymized movie ratings were de-anonymized using IMDb reviews.

A dataset with **Date of Birth, Sex, and ZIP code** can re-identify most U.S. citizens.

In fact, **87%** of Americans can be uniquely identified using just these three attributes

Traditional anonymization techniques fail against modern data linkage and AI-driven inference attacks — demanding stronger privacy guarantees like **Differential Privacy**.

Need for Computational Privacy

GOAL : Privacy Preserving Data Analysis

- Obtain answers or usefulness in surveys, but at the same time maintain the privacy or "plausible deniability
- Provides **privacy guarantees**
- Protects against **a wide range of attacks** — even unforeseen ones
- Ensures **trustworthy data analysis** in deep learning and AI

What is Differential Privacy

Definition:

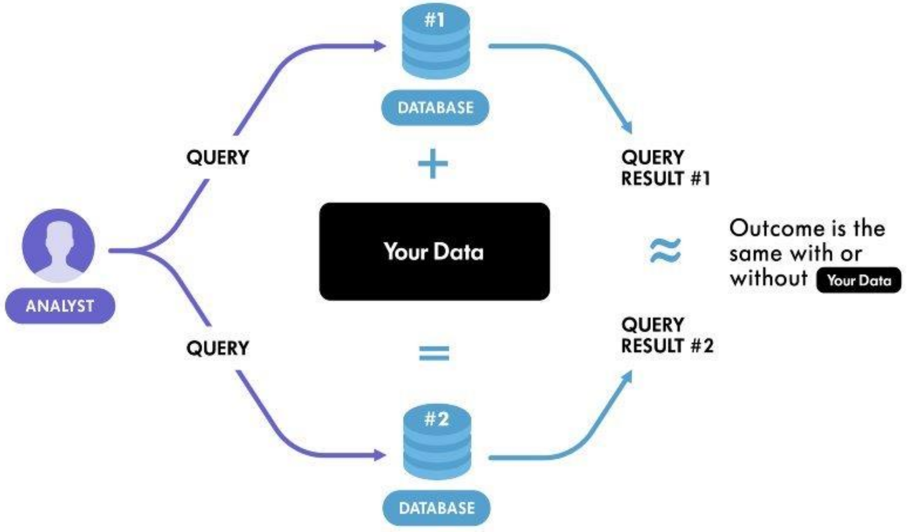
Differential privacy (DP) is a mathematically rigorous framework for releasing statistical information about datasets while protecting the privacy of individual data subjects.

Key Idea (Cynthia Dwork, 2016):

“The outcome of any analysis is essentially equally likely, independent of whether any individual joins or refrains from joining the dataset.”

Core Principle (Kearns & Roth, 2020):

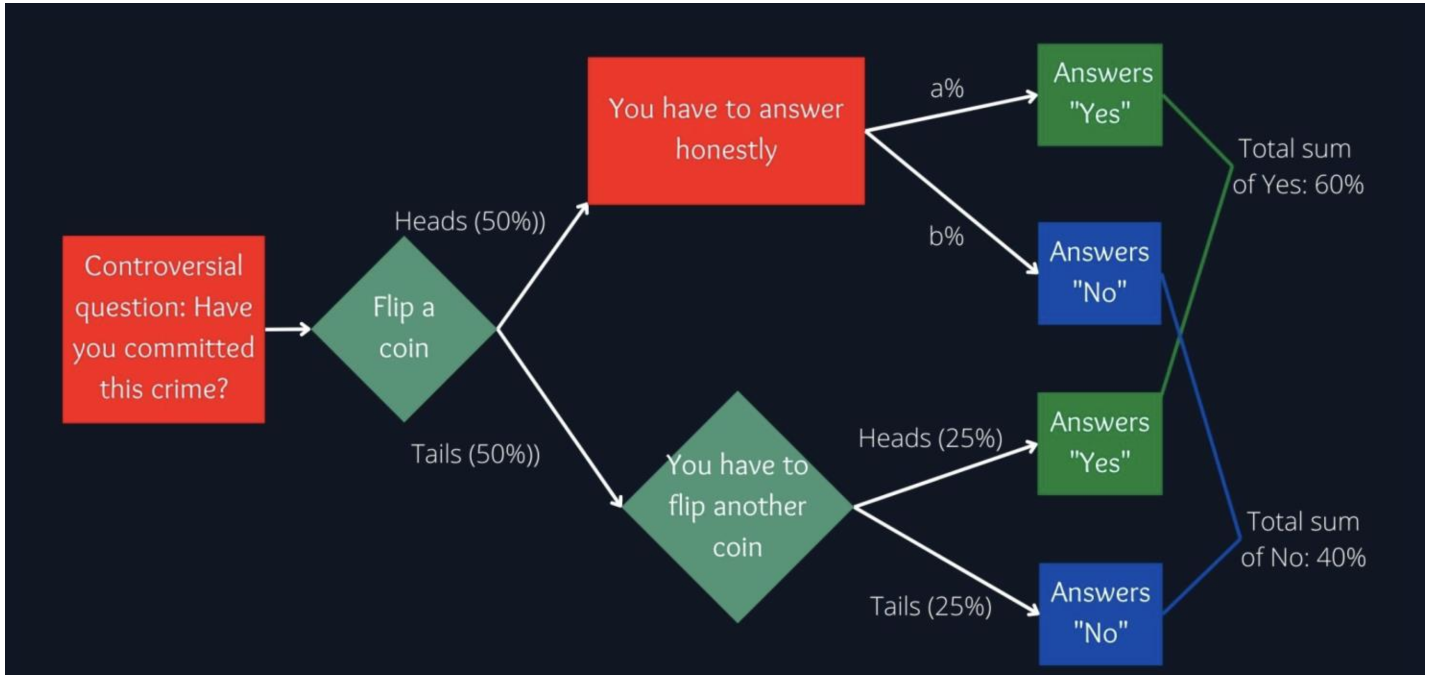
"Adding or removing the data record of a single individual does not change the probability of any result too much".

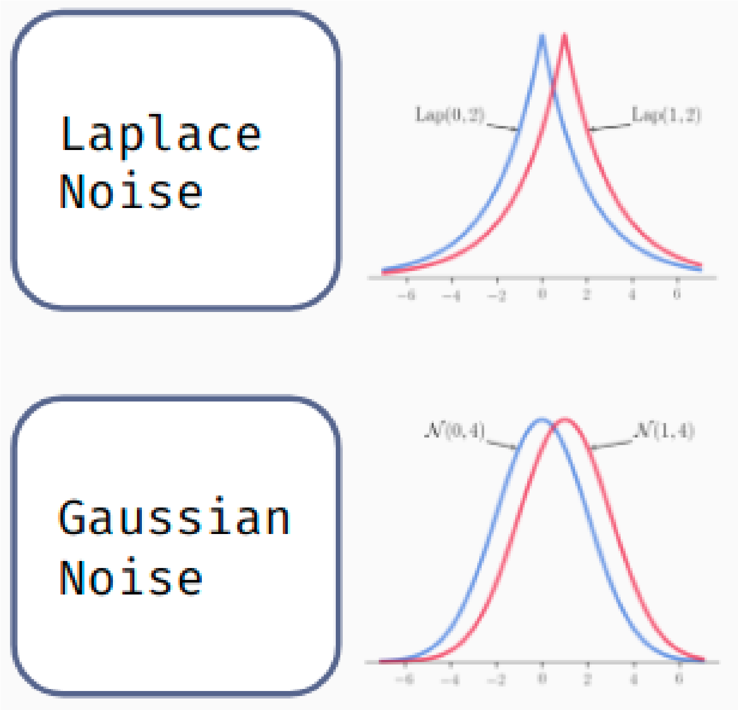


```
Google receives only noisy data and aggregates it...

Browser          True Count       Private Count    Error
--------------------------------------------------------------------------------
Chrome           6,483            6,956            7.3          %
Safari           2,055            2,799            36.2         %
Edge             753              1,622            115.4        %
Firefox          505              1,590            214.9        %
Other            204              1,202            489.2        %

================================================================================
KEY INSIGHTS
================================================================================
✅ Individual Privacy: No one (not even Google) knows your actual browser
✅ Useful Statistics: Aggregate trends are preserved with high accuracy
✅ Mathematical Guarantee: ε-differential privacy ensures plausible deniability

💡 With ε=1.0:
   • Any individual can deny their true browser choice
   • Aggregate statistics have ~1.0% statistical error

================================================================================
PRIVACY GUARANTEE DEMONSTRATION
================================================================================

Example: Alice uses Firefox, but reports 'Chrome' due to randomization

Alice's TRUE browser: firefox
Alice's 10 reports:   ['safari', 'firefox', 'chrome', 'safari', 'other', 'chrome', 'edge', 'chrome', 'firefox', 'safari']

→ Even if someone intercepts Alice's report, they can't be sure
  of her actual browser because randomization provides plausible deniability!
```

```
================================================================================
BONUS: RAPPOR - GOOGLE'S ACTUAL CHROME IMPLEMENTATION
================================================================================

Browser: chrome
5 RAPPOR-encoded reports (each is a bit vector):
  Report 1: [1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0]
  Report 2: [0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1]
  Report 3: [1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1]
  Report 4: [0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1]
  Report 5: [0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1]

Note: RAPPOR provides even stronger privacy by:
  • Using Bloom filters for efficient encoding
  • Adding multiple layers of randomization
  • Enabling longitudinal privacy (multiple reports from same user)
```

Mathematics Behind Differential Privacy





Probability of seeing output O on input D_1

Probability of seeing output O on input D_2

$$\frac{\Pr[\mathcal{M}(D_1) \in O]}{\Pr[\mathcal{M}(D_2) \in O]} \leq e^{\varepsilon}$$

Indistinguishability: bounded ratio of probabilities

🧮 Mathematical Formula:

```
   TY = HY + FY
   where:
      TY = Total Yes (observed)
      HY = Honest Yes = p × True_Yes_Rate
      FY = False Yes = (1-p) × 0.5 = 0.25

   Solving for True_Yes_Rate:
      True_Yes_Rate = (TY - FY) / p
                    = (0.6010 - 0.25) / 0.5
                    = 0.7020
```

Metric	Value
True 'Yes' Rate	70.00%
Observed 'Yes' Rate	60.10%
Estimated 'Yes' Rate	70.20%
Estimation Error	0.20%

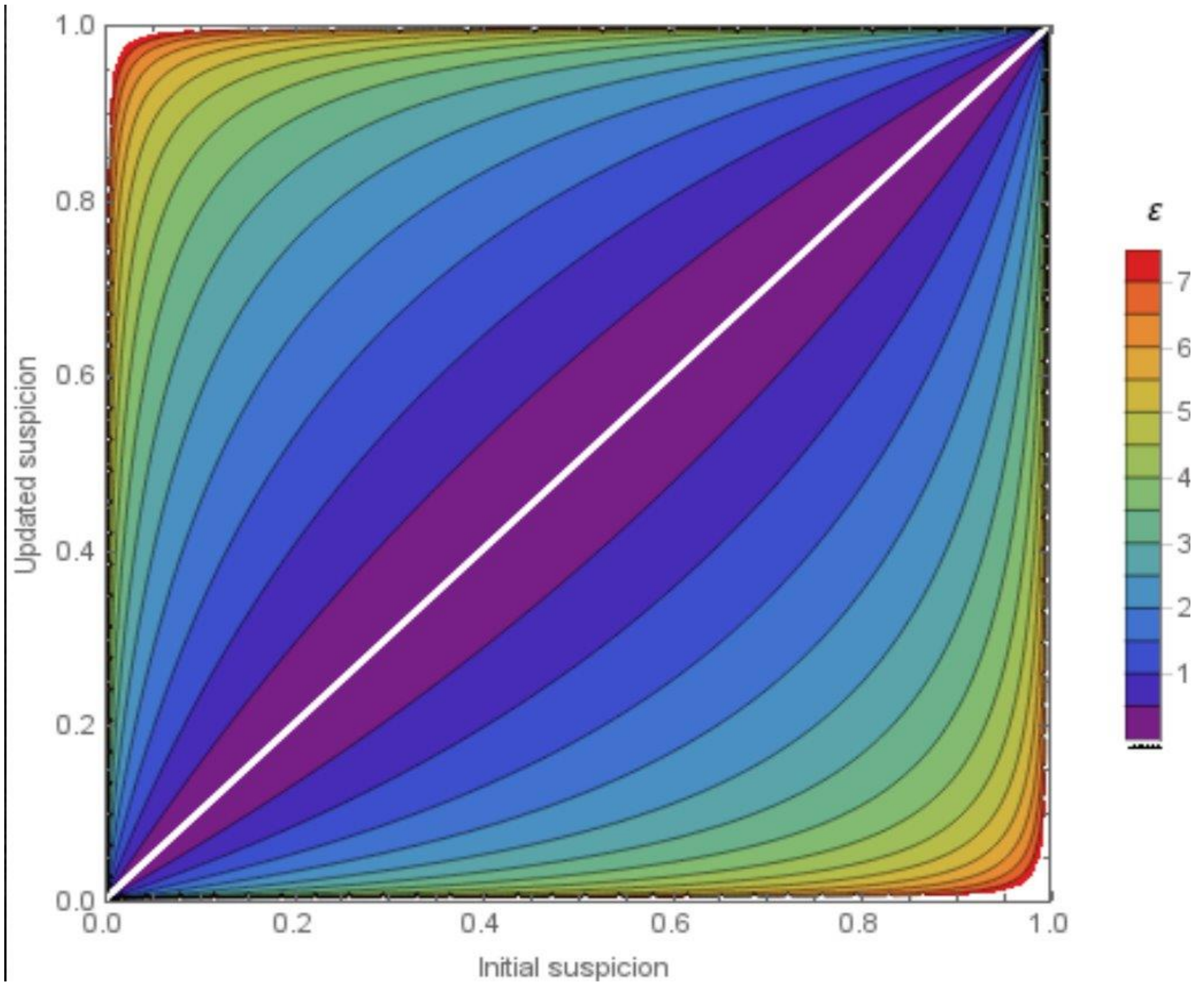
```
   Press ENTER to verify privacy guarantees...

================================================================================
                      DIFFERENTIAL PRIVACY VERIFICATION
================================================================================
```

🔬 Testing the mathematical guarantee:

```
   Pr[K(D₁) ∈ S] ≤ exp(ε) × Pr[K(D₂) ∈ S] + δ

   Dataset D₁: 600 'Yes' → Pr = 0.6000
   Dataset D₂: 601 'Yes' → Pr = 0.6010

   Privacy Loss: ln(0.6010/0.6000) = 0.0017
   Required Bound: ε + δ = 1.0000 + 1e-05 = 1.0000
```

✅ SATISFIES (ε,δ)-Differential Privacy

Privacy Budget Epsilon

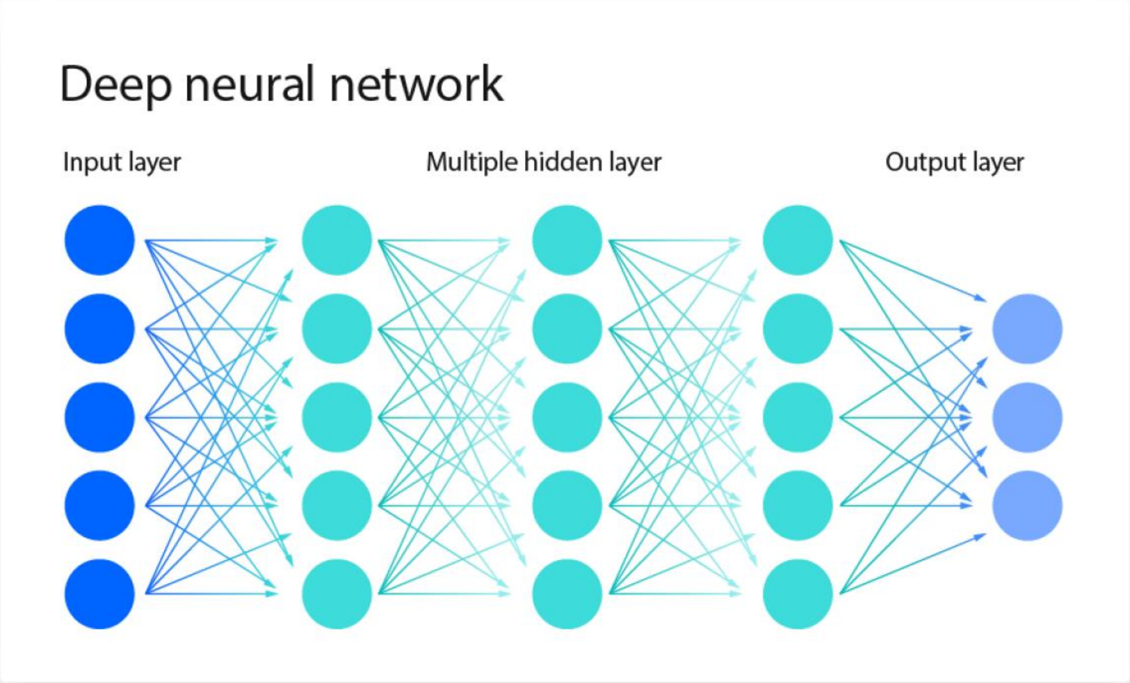


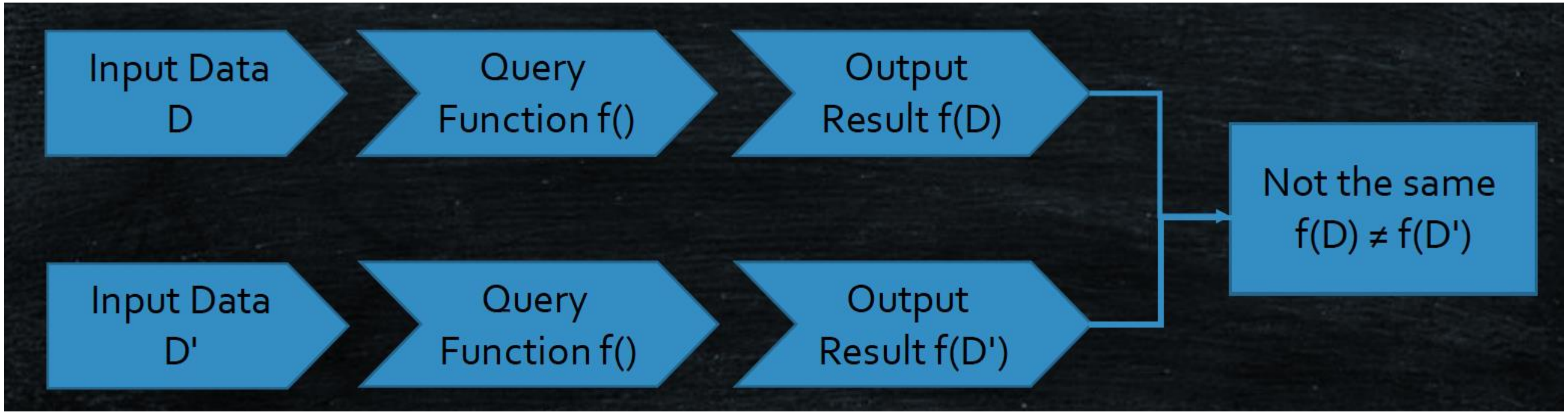
Initial suspicion was 50%,

ϵ=1.1 epsilon

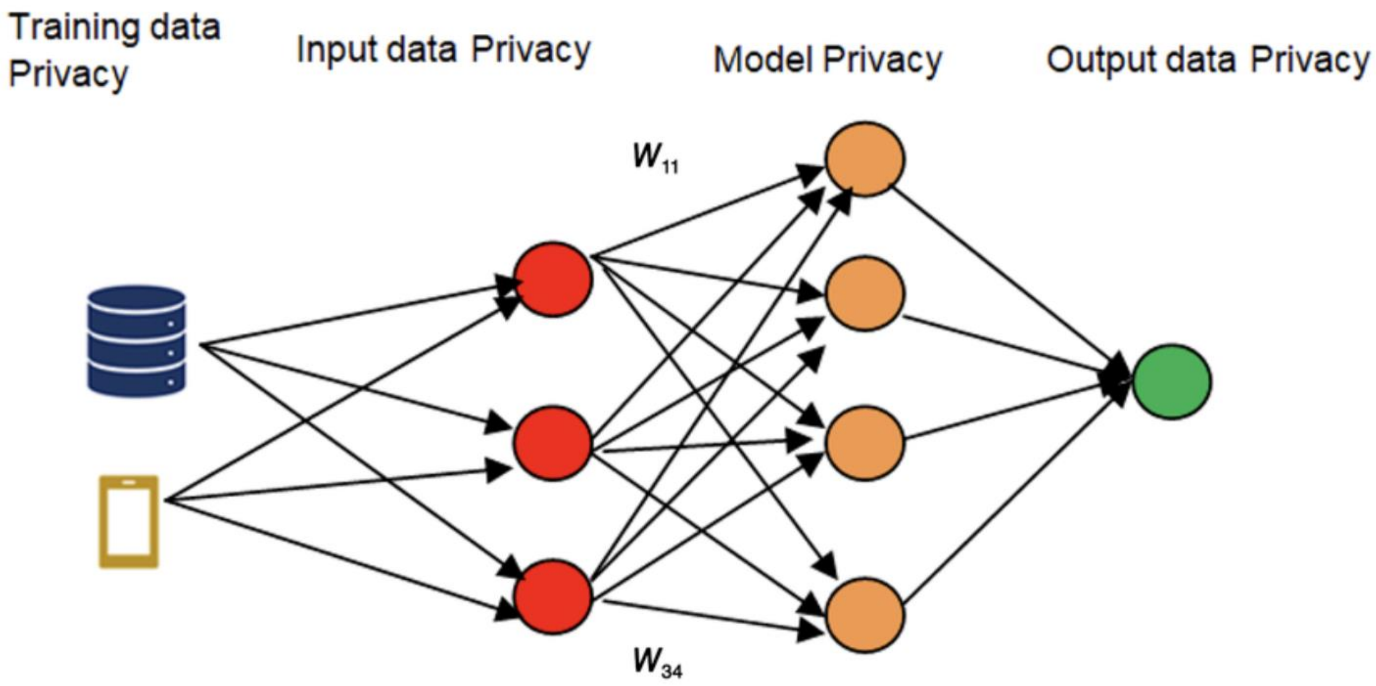
Updated suspicion = range of 25% to 75%.

Machine Learning Pipeline

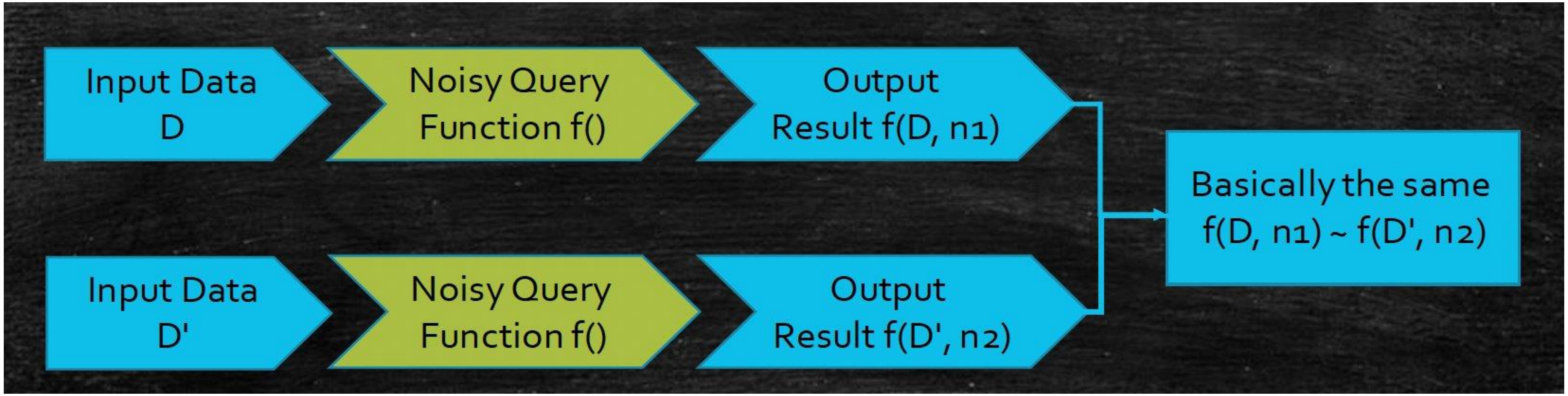




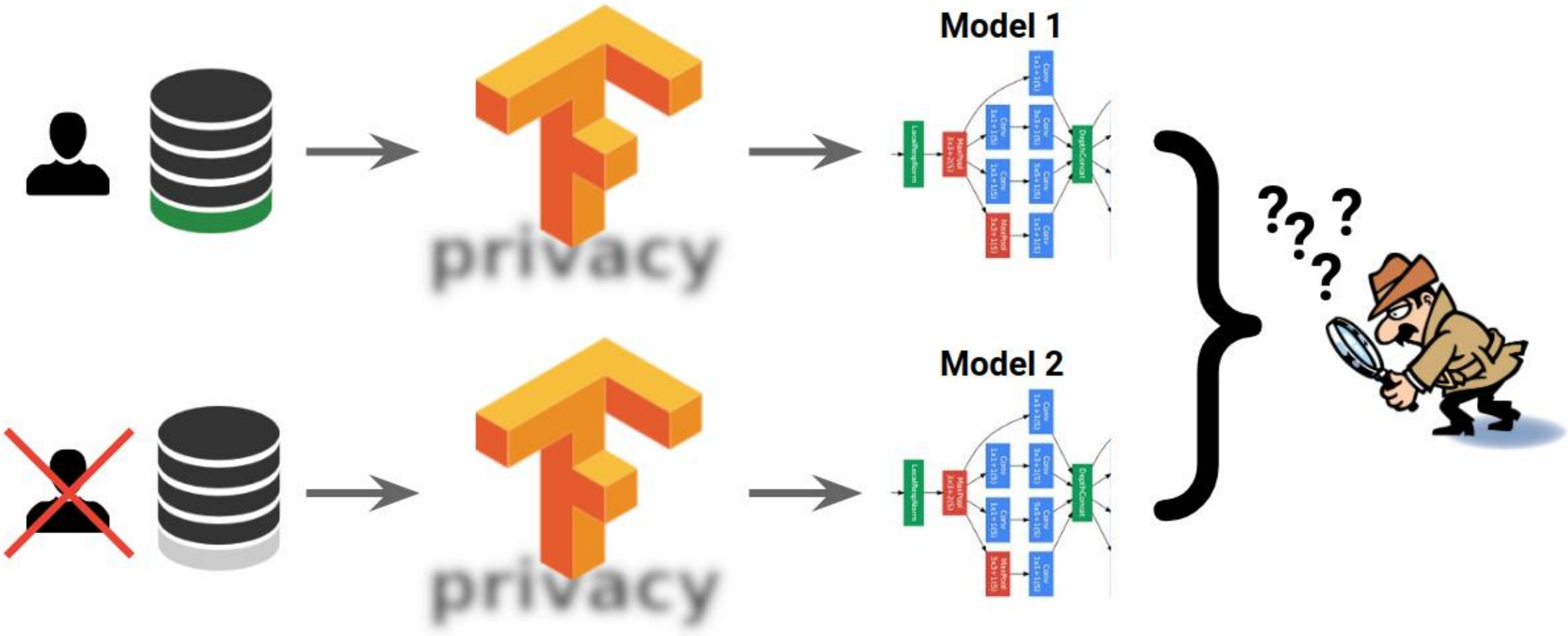
Differential Privacy Pipeline



Data privacy at different stages in deep learning network



Differential Privacy in Practice



Government, Healthcare and Technology

United States® Census Bureau

amazon

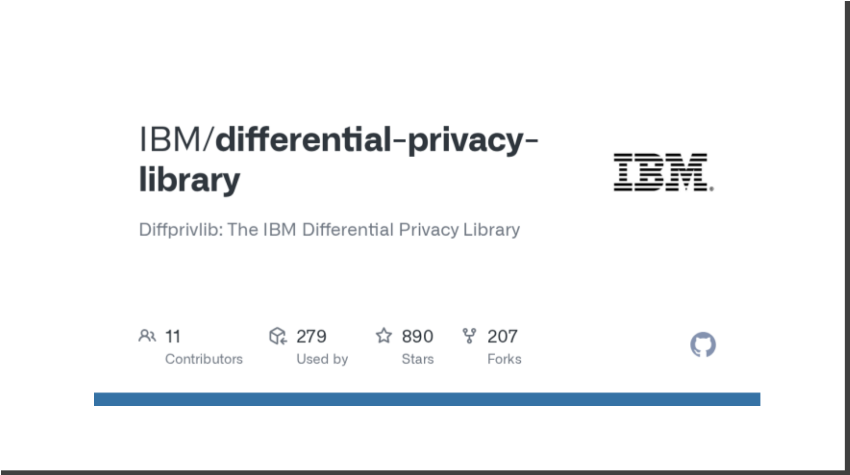
Google

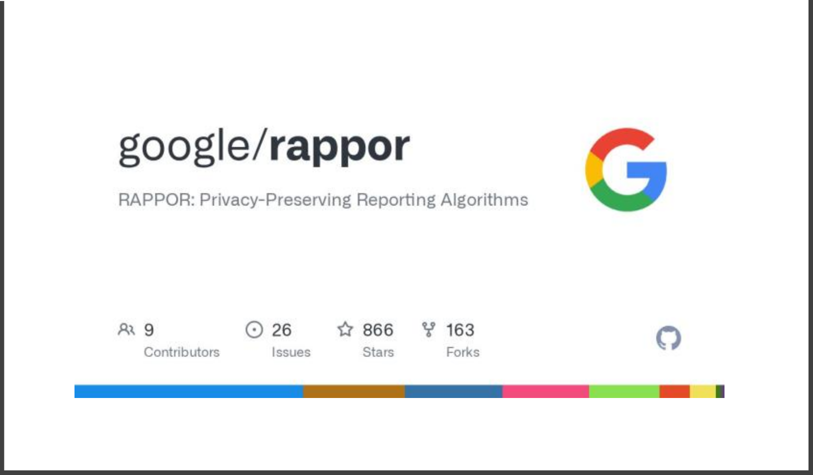
Uber

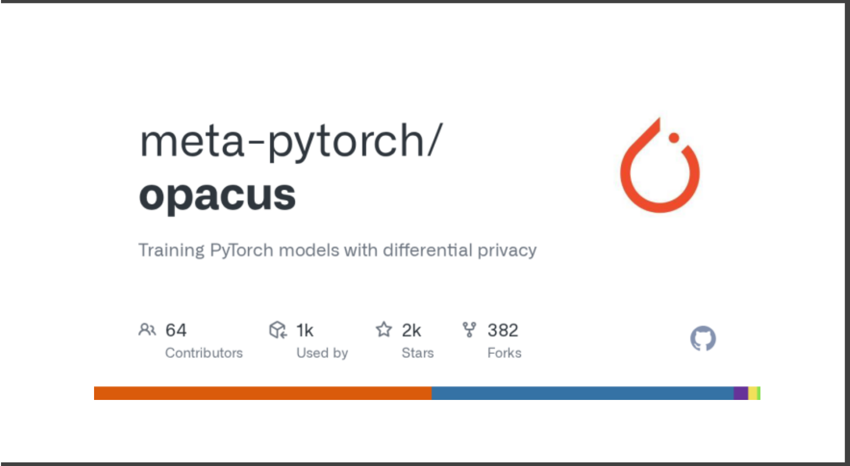
	Local Differential Privacy (LDP)	Central Differential Privacy (CDP)	Differentially Private SGD (DP-SGD)
Definition	Each user's data is randomized before it is sent to the server. The server never sees the raw data.	Raw data is collected and stored by a trusted server, and noise is added when releasing aggregate results.	A training algorithm where noise is added to gradients during model training to ensure privacy.
Noise	At the client/user side (before data collection).	At the server/output side (after data collection).	During model training, to the gradients and clipped updates.
Privacy Parameter (ε)	Typically larger ε (less accuracy) because each user adds noise individually.	Smaller ε possible with the same privacy level — better accuracy since aggregation reduces noise.	ε controlled via gradient clipping and noise multiplier; balance between model utility and privacy.

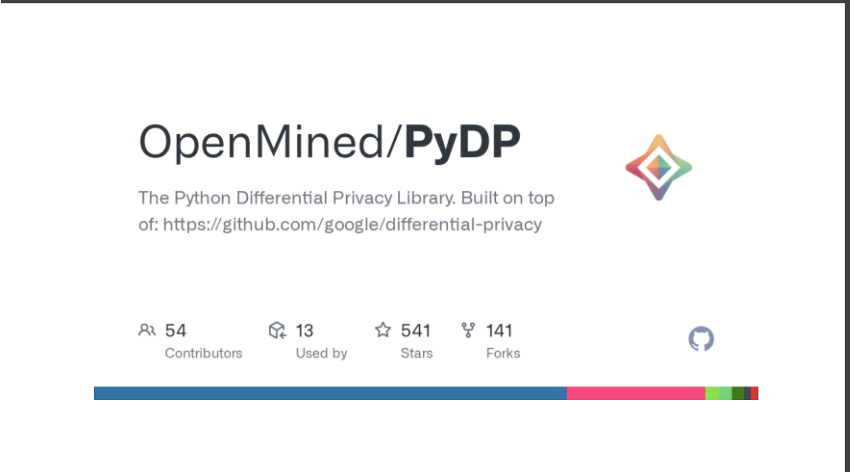
	Local Differential Privacy (LDP)	Central Differential Privacy (CDP)	Differentially Private SGD (DP-SGD)
Key Trade-off	Strongest privacy, weakest accuracy	Balanced privacy and accuracy	Moderate privacy, good accuracy for ML
Example Use Cases	- Google Chrome's RAPPOR for user telemetry - Apple's iOS keyboard data - Privacy-preserving analytics without raw data sharing	- Census data release (e.g., US Census Bureau) - Aggregate statistics and queries on sensitive datasets	- Training privacy-preserving ML models - Used in Google's and OpenAI's research for differentially private training
Frameworks / Tools	OpenDP / Google's RAPPOR / Apple LDP framework	TensorFlow Privacy (for central DP queries) / OpenDP	TensorFlow Privacy, PyTorch Opacus, Diffprivlib

Differential Privacy Framework and Tools









```
================================================================================
📚 LIBRARIES FOR PRODUCTION USE
================================================================================

1. Google's PipelineDP:
   pip install pipeline-dp
   → Production-ready, used internally at Google

2. OpenMined's PyDP:
   pip install python-dp
   → Python bindings for Google's differential-privacy library

3. IBM's diffprivlib:
   pip install diffprivlib
   → Scikit-learn compatible, easy to use
```

References

Google

ChatGPT
Gemini
perplexity
Claude

NotebookLM

https://github.com/sharikalog7/Differential-Privacy