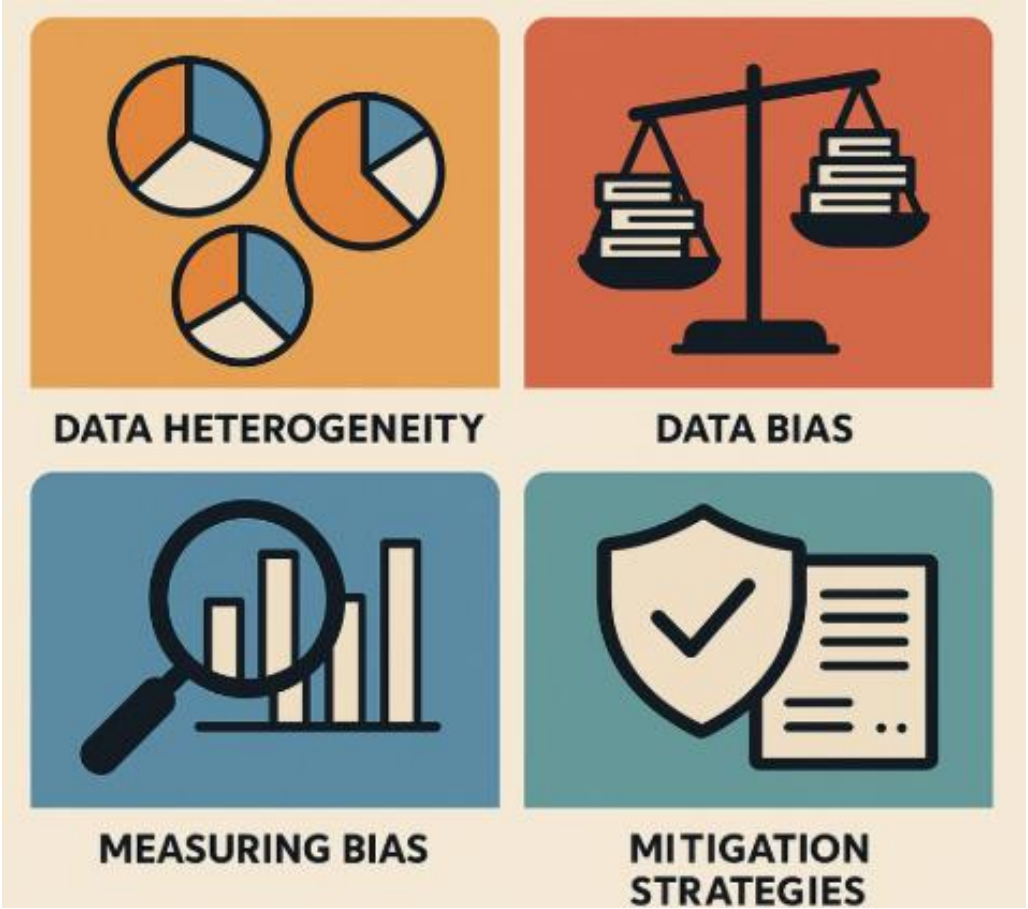
# Data Problems in Federated Learning



# Data Heterogeneity

Example: 2 hospitals collaborate to train a disease diagnosis model using Federated Learning

**The Bias Problem**

- Hospital A (urban): 90% cases of disease X, advanced imaging equipment
- Hospital B (rural): 10% cases of disease X, basic equipment

- Model performs poorly at Hospital B

# Data Bias

Systematic differences in data distribution across federated clients that lead to unfair or suboptimal model behavior.

**Centralized ML**

Data collected centrally, can be balanced and cleaned before training

**Federated Learning**

Data remains distributed, inherently heterogeneous, cannot be directly balanced

**Non-IID Data**

Data across clients is not independently and identically distributed

**Heterogeneity**

Each client has different data characteristics and distributions

- Bias is amplified in FL due to data isolation and heterogeneity

# Why Bias Matters in FL

Models may converge slowly, perform poorly, or exhibit unfair behavior across different client populations

- **Performance disparities:** Model works well for some clients, poorly for others
- **Fairness violations:** Discriminatory outcomes for underrepresented groups
- **Reduced adoption:** Clients drop out when model doesn't serve them
- **Ethical and legal risks:** Regulatory violations (GDPR)
- **Slower convergence:** Biased gradients hinder optimization

# Types of Data Heterogeneity

## Feature Distribution Skew

Same labels, different feature distributions

Example: Medical imaging from different hospitals using various equipment

## Label Distribution Skew

Unbalanced class distributions across clients

Example: Smartphone keyboards with user-specific vocabulary

## Quantity Skew

Varying amounts of data per client

Example: Active vs. passive app users

## Temporal Skew

Data collected at different time periods

Example: Sensor data from different time zones

# Feature Distribution Skew

Input features have different distributions across clients, but labels remain consistent

**Example: Image Quality Variation**

- Client 1: High-resolution cameras
- Client 2: Low-resolution cameras
- Client 3: Night mode images (low light)

Same objects, different feature distributions

Same label but very different image characteristics features across hospitals

# Label Distribution Skew

Different clients have different distributions of class labels

**Example: Digit Recognition**

- Client 1: 80% digits 0-4, 20% digits 5-9
- Client 2: 20% digits 0-4, 80% digits 5-9
- Client 3: Only digits 0, 1, 2

- Model learns biased patterns.

# Quantity Skew

**Imbalanced Data Volumes: c**lients have vastly different amounts of local data

**Example: Mobile Keyboard App**

- Power user: 100,000 typing samples
- Average user: 1,000 typing samples
- New user: 50 typing samples

**Large clients dominate aggregation, small clients underrepresented**

# Temporal Skew

## Time-Dependent Distributions

Data distributions change over time at different rates across clients

### Example: E-commerce Recommendations

- Region A: Holiday shopping spike in December
- Region B: Holiday shopping spike in January
- Region C: No seasonal patterns
- Concept drift occurs at different times

# Impact on Model Performance

### Slow Convergence

Conflicting gradients from heterogeneous clients lead to training instability

### Weight Divergence

Local models drift apart, making aggregation less effective

### Poor Generalization

Model performs well on some clients but poorly on others

### Fairness Issues

Minority groups underrepresented in final model

# Mitigation Strategy 1: Smart Client Selection

Select clients representing diverse data distributions in each training round

- **Diversity-Based Selection**: Select clients to maximize diversity in data distributions within each round, ensuring broad coverage of the global data space
  - **Feature Space Coverage**: Select clients covering different regions of feature space
  - **Cluster-Based**: Group similar clients, select one from each cluster

# Mitigation Strategy 1: Smart Client Selection

Select clients representing diverse data distributions in each training round

- **Gradient based selection**: Select clients whose gradients provide maximum information for improving the global model
  - **Gradient Norm Selection**: Select clients with largest gradient magnitudes
    - May select outliers/noisy clients
  - **Gradient Diversity**: Choose clients with diverse gradient directions
    - Computationally expensive
  - **Loss-Based Selection**: Prioritize clients with highest current loss
    - May indicate bad data quality

# Mitigation Strategy 1: Smart Client Selection

Select clients representing diverse data distributions in each training round

- **Fairness-Aware Selection**: Ensure equitable participation and performance across all clients, especially underrepresented or minority groups. Prevent majority domination
  - **Equal Opportunity Selection:** Equal participation probability: ensures minority clients participate equally, prevents majority domination
  - **Performance-Based Fairness:** select struggling clients more often, balances global and local performance

# Mitigation Strategy 2: Weighted Aggregation

**Adjust influence of each client during model aggregation:**

$$w_{t+1} = \Sigma\ \alpha_k \cdot w^k_t \qquad \sum_k \alpha_k = 1$$

- **Data-Size Weighting:** $\alpha_k = n_k / n$
- **Loss-Based Weighting:** Clients with higher local loss get higher weight (to prioritize under-performing clients).
- **Gradient Similarity Weighting:** Weights are adjusted based on how aligned local gradients are with global direction.
- **Fairness-Aware Weighting:** Assigns more weight to disadvantaged or minority clients.
- **Adaptive Weighting:** Weights updated across rounds using performance metrics.

# Mitigation Strategy 3: Personalization

**Instead of forcing one global model, allow client-specific adaptation:**

- **Fine-tuning:** Train global model, then locally fine-tune on client data
- Better local performance but requires more computation and storage per client

# Data Bias

1. **Historical Bias:** Existing societal inequalities reflected in data
2. **Representation Bias:** Unbalanced or incomplete data collection
3. **Measurement Bias:** Flawed features or proxies for target concepts
4. **Algorithmic Bias:** Model design choices that favor certain groups
5. **Deployment Bias:** Inappropriate application

# Historical Bias

Data accurately reflects past inequalities and discrimination

**Example: Hiring Algorithms**

Amazon's AI recruiting tool (2014-2017)

- Trained on 10 years of historical resumes
- Most past candidates were male

- System learned to penalize resumes with "women's"

# Representation Bias

Training data fails to represent the target population

**Example: Face Recognition**

Joy Buolamwini's Gender Shades Study (2018)

- Tested IBM, Microsoft on diverse faces
- Error rate for light males: 0.8%
- Error rate for dark females: 34.7%
- Training datasets 75% male, 80% white

Worse performance for underrepresented groups

# Measurement Bias

Choosing features that are poor proxies for the true target or systematically disadvantaged groups

**Example: activity recognition (using accelerometer and gyroscope) using phones from different manufacturers (Samsung, Apple, Xiaomi).**

- Accelerometers and gyroscopes have different sensitivities.
- “Walking” activity looks statistically different across devices.
- The global model misclassified walking patterns from under-represented device types

- Different sensors, devices, or recording conditions
- Distorted global model, reduced fairness and generalization

# Algorithmic Bias

**Aggregation process itself introduces or amplifies unfairness**, even if the data were balanced or representative.

### Example: Federated Credit Scoring System

Banks across different regions collaboratively train a **credit risk prediction model**

| Client (Bank) | Region | Local Data Size | Loan Denial Rate | Economic Context |
|---|---|---|---|---|
| A | Urban | 50,000 | 20% | High income |
| B | Suburban | 15,000 | 30% | Medium income |
| C | Rural | 5,000 | 60% | Low income |

Large clients (Bank A) have **greater influence** on the global model. The **global model** underestimates loan risk for wealthy applicants (urban bias).

**Rural applicants** (low-income) are more often misclassified as *high risk* even when they repay on time.

Bias is **not from data itself**, but from the **aggregation algorithm giving disproportionate influence** to dominant clients.

# Deployment Bias

A **federated model is deployed in an environment different from the one it was trained for**, leading to unfair or unreliable predictions.

**Example: Healthcare:** diabetic retinopathy detection from eye images

| Hospital | Region | Device Type | Population |
|---|---|---|---|
| A | USA | High-end fundus camera | Mostly Caucasian patients |
| B | India | Mid-range fundus camera | Indian patients |
| C | Kenya | Mobile camera (low resolution) | African patients |

- Model achieves high average performance
- The final global model is deployed in rural African clinics
- The local devices have lower-resolution cameras, and lighting conditions differ.
- The local patient population also has different retinal pigmentation patterns.

# How Bias Manifests

### Allocation Harm

Unequal distribution of opportunities or resources

### Quality of Service

System works better for some groups than others

### Stereotyping

Reinforcing or amplifying harmful stereotypes

### Erasure

Making certain groups invisible or unrecognized

# Allocation Harm

Biased AI decisions lead to **unequal distribution of opportunities, resources, or benefits** across groups — e.g., who gets approved, hired, or prioritized.

**Example: Credit Scoring**

- Large banks (urban clients) have mostly wealthy customers.
- Small rural banks have clients with different spending patterns.

- The global FL model learns patterns biased toward urban, high-income behavior.
- The model systematically underestimates rural customers.
- Allocation harm: fewer loans, fewer opportunities for certain populations.

# Quality of Service

Occurs when a model performs worse for certain groups or regions, leading to inconsistent or degraded service quality.

**Example: Healthcare**

- The global model is trained using data from hospitals with high-resolution MRI scanners.

- Deployed in low-resource clinics using older scanners, predictions become less accurate.
- Patients in poorer regions receive **lower diagnostic accuracy**, even though they’re part of the same system.

# Stereotyping

Occurs when the model **reinforces existing stereotypes** or overgeneralizes based on biased correlations in data.
It captures **social or cultural biases** present in the training data and replicates them.

**Example**: recruitment system across companies:

- One client company has mostly male engineers.
- Another has mostly female HR staff.

- The global FL model associates “engineer” with male names and “HR” with female names.
- The model scores women lower for engineering jobs.

# Erasure

Occurs when certain groups or data types are underrepresented or entirely missing from training data — leading the model to “erase” them from recognition or prediction.
The model effectively acts as if those groups don’t exist.

Example: speech recognition

- Clients from major cities contribute diverse, clean audio data.
- Remote regions with poor internet connectivity cannot participate in training - their local languages and dialects are never learned.

- The global model performs poorly on those dialects — effectively erasing them from the system’s linguistic understanding.

# Compounded Bias

- Example: healthcare

| Stage | Type of Bias | Description | Effect |
|---|---|---|---|
| Data Collection | Sampling bias | Dataset mostly includes light-skinned faces | Darker-skinned individuals underrepresented |
| Measurement | Sensor bias | Camera sensors calibrated for lighter tones | Poor feature extraction for dark skin |
| Algorithmic | Training bias | Model minimizes global loss dominated by majority | Model optimizes for lighter faces |
| Deployment | Context bias | Used in airport security in Africa or Asia | System fails to recognize many passengers |

# Compounded Bias

- **Fairness degradation:** model works best for groups already well represented
- **Trust erosion:** users in underrepresented regions lose confidence in system fairness
- **Ethical and legal risks:** amplified inequalities in healthcare

| Stage | Mitigation Strategy |
|---|---|
| **Data** | Ensure diverse, representative, balanced client datasets. |
| **Measurement** | Calibrate sensors/devices; normalize data across clients. |
| **Algorithmic** | Use fairness-aware or reweighted aggregation. |
| **Deployment** | Perform domain adaptation at new sites. |
| **Monitoring** | Continuously audit fairness after deployment. |

# How Bias is Measured?

**Categories of Bias Metrics**

1. **Individual Fairness Metrics:** Similar individuals treated similarly
2. **Group Fairness Metrics:** Statistical parity across demographic groups
3. **Performance Disparity Metrics:** Accuracy differences across clients or groups
4. **Contribution Fairness Metrics:** Fair valuation of client data contributions

# Individual Fairness Metrics

- Similar examples should receive similar predictions

$$if\ d(x_i, xj) is\ small, then\ |\hat{Y}(x_i)\ - \hat{Y}(xj)| should\ also\ be\ small$$

$d(x_i, x_j)$ – similarity distance between two examples (in feature space)

$\hat{Y}(x)$: predicted score / probability

# Individual Fairness Metrics

- Each client computes local fairness scores:

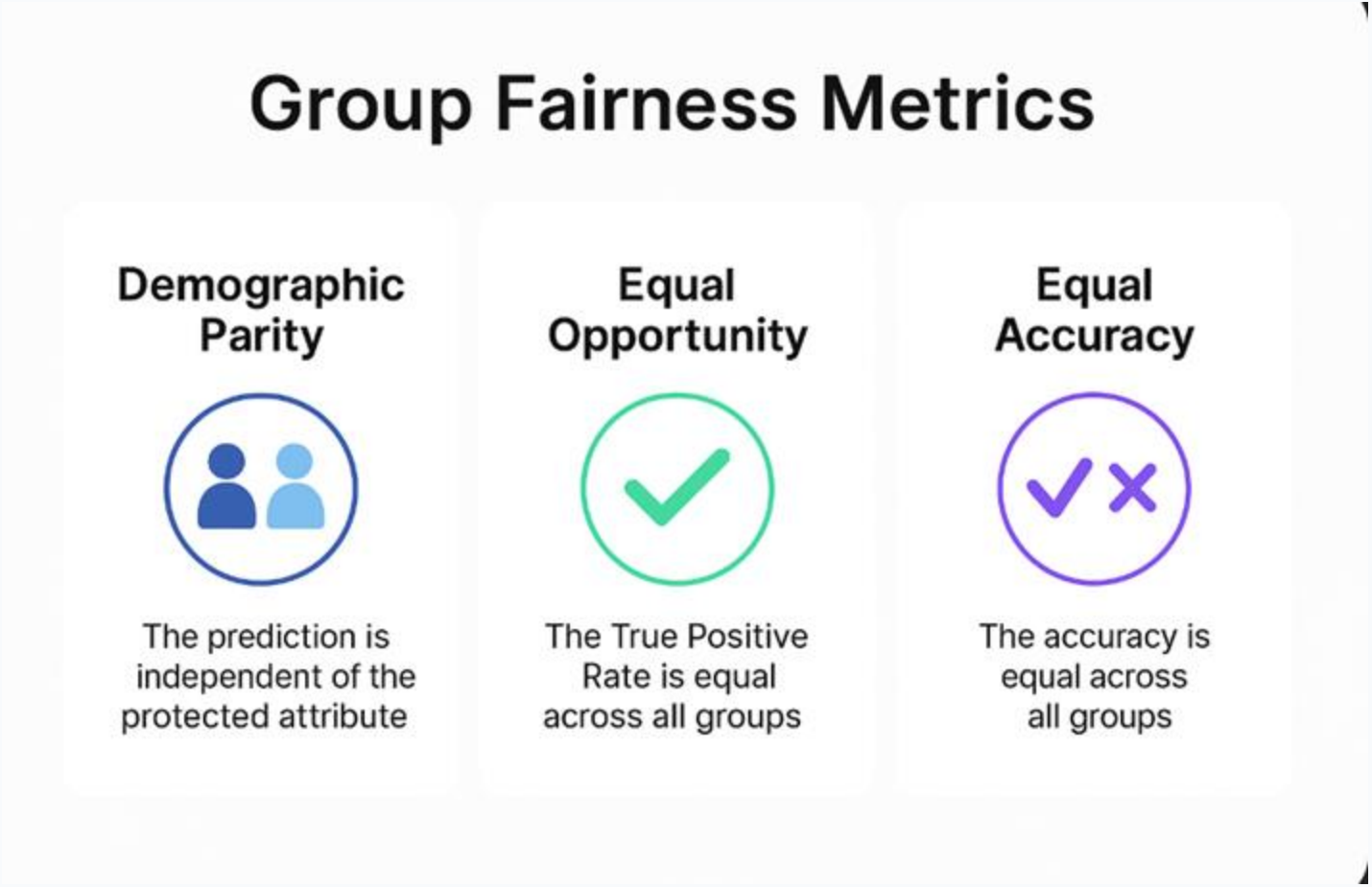
$$\mathcal{L}_{fair} = \sum_{i,j} \max(0, |\hat{Y}_i - \hat{Y}_j| - L \cdot d(x_i, x_j))$$

- L: constant ensuring fairness.

- Combine with standard loss:

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \lambda \cdot \mathcal{L}_{fair}$$

# Group Fairness Metrics



# Group Fairness Metrics

**1. Demographic Parity (Statistical Parity):** A model satisfies Demographic Parity if its predictions are *independent* of a sensitive attribute (e.g., gender, race, region).

- The probability of getting a positive prediction (e.g., “approved loan,” “disease detected,” “admitted to program”) should be **the same across all demographic groups**, regardless of their actual labels.
- Equal positive prediction rate across groups

$$P(\hat{Y} = 1|A = a) = P(\hat{Y} = 1|A = b)$$

Demographic parity

$$\text{DP Difference} = |P(\hat{Y} = 1|A = a) - P(\hat{Y} = 1|A = b)|$$

Bias is measured using DP difference or DP ratio

$$\text{DP Ratio} = \frac{P(\hat{Y} = 1|A = a)}{P(\hat{Y} = 1|A = b)}$$

$$0.8 \leq \text{DP Ratio} \leq 1.25$$

$\hat{Y}$= model prediction (1 = positive outcome)
A = sensitive attribute (e.g., gender)
a,b = two demographic groups (e.g., male, female)

# Group Fairness Metrics

**1. Demographic Parity (Statistical Parity):**

It is not possible to compute global

$$P(\hat{Y}|A)$$

1. Each client computes local probability
2. Secure aggregation: clients send *aggregated counts*
3. Global estimation by the server

$$P(\hat{Y}=1|A=a)=\frac{\sum_i n_{a,i}^{+}}{\sum_i n_{a,i}}$$

$$\text{DP difference} = |P(\hat{Y}=1|A=a)-P(\hat{Y}=1|A=b)|$$

$$\text{DP ratio} = \frac{P(\hat{Y}=1|A=a)}{P(\hat{Y}=1|A=b)}$$

# Group Fairness Metrics

**2. Equal Opportunity:** checks whether a model is equally accurate for the “positive” class across all demographic groups.

It ensures that individuals who *deserve* a positive outcome (true label = 1) have an equal chance of being correctly predicted as positive, regardless of their group.

$$P(\hat{Y} = 1 \mid Y = 1, A = a) = P(\hat{Y} = 1 \mid Y = 1, A = b)$$

$\hat{Y}$: predicted label

Y: true label

A: sensitive attribute (e.g., gender, ethnicity, region)

# Group Fairness Metrics

**2. Equal Opportunity:**

Example: medical diagnosis model

| Group | True Positive Rate (TPR) | Meaning |
|---|---|---|
| Men | 0.90 | 90% of sick men correctly diagnosed |
| Women | 0.70 | 70% of sick women correctly diagnosed |

- Equal Opportunity focuses specifically on **true positive rate** for all groups.

# Group Fairness Metrics

**2. Equal Opportunity:**

$$TPR_a = P(\hat{Y} = 1 | Y = 1, A = a)$$
$$TPR_b = P(\hat{Y} = 1 | Y = 1, A = b).$$

**Equal Opportunity Difference**

$$|TPR_a - TPR_b|$$

Equal Opportunity Ratio

$$\frac{TPR_a}{TPR_b}$$

A model is fair

$$0.8 \leq \frac{TPR_a}{TPR_b} \leq 1.25$$

# Group Fairness Metrics

**2. Equal Opportunity:**

Each client (e.g., hospital, bank, device) computes counts locally:

$$n_{a,i}^{(1,1)} = |\{x \in D_i : Y = 1, \hat{Y} = 1, A = a\}|$$

$$n_{a,i}^{(1)} = |\{x \in D_i : Y = 1, A = a\}|$$

i index of the client

$n_{a,i}^{(1,1)}$: number of *true positives* for group a on client i

$n_{a,i}^{(1)}$: number of *actual positives* for group a on client i

Secure global aggregation

computes the **global True Positive Rate per group**

$$N_a^{(1,1)} = \sum_i n_{a,i}^{(1,1)}$$

$$N_a^{(1)} = \sum_i n_{a,i}^{(1)}$$

$$TPR_a = \frac{N_a^{(1,1)}}{N_a^{(1)}}$$

$$TPR_b = \frac{N_b^{(1,1)}}{N_b^{(1)}}$$

# Group Fairness Metrics

**2. Equal Opportunity:**

Equal Opportunity Fairness Metric EOD

$$EOD = |TPR_a - TPR_b|$$

$$EOR = \frac{TPR_a}{TPR_b}$$

A model is considered **fair** if:

$$0.8 \leq EOR \leq 1.25$$

# Group Fairness Metrics

**3. Equalized Odds:** a **stronger fairness criterion** than Demographic Parity or Equal Opportunity.

$$P(\hat{Y} = 1 \mid Y = y, A = a) = P(\hat{Y} = 1 \mid Y = y, A = b) \quad \text{for both } y \in \{0, 1\}$$

- For **Y = 1 (positive cases)** → equal chance of being correctly predicted
- For **Y = 0 (negative cases)** → equal chance of being *not* wrongly predicted

# Group Fairness Metrics

**3. Equalized Odds:**

For group A=a:

$$TPR_a = P(\hat{Y} = 1 \mid Y = 1, A = a)$$

$$FPR_a = P(\hat{Y} = 1 \mid Y = 0, A = a)$$

Equalized Odd:

$$TPR_a = TPR_b \quad \text{and} \quad FPR_a = FPR_b$$

**Differences** or **ratios** between groups can measure the fairness gap (**EOdds_Diff close to 0 for a fair model**)

$$EOdds\_Diff = |TPR_a - TPR_b| + |FPR_a - FPR_b|$$

Normalized version:

$$EOdds\_Ratio = \frac{TPR_a / FPR_a}{TPR_b / FPR_b}$$

# Performance Disparity Metrics

**Measure how well the model performs for different clients or groups**

- **Accuracy gap:** Difference between best and worst performing clients
- **Standard Deviation of Performance**: spread of client performance around the mean
- **Variance in Performance:** Spread of accuracies across all clients

# Accuracy Gap

Measures the **difference between best and worst performance**:

- Small gap: uniform performance
- Large gap: significant disparity

| Hospital | Location | Data Size | X-ray Device | Population Type |
|---|---|---|---|---|
| A | Urban | 50,000 | High-end scanner | Adults |
| B | Suburban | 20,000 | Mid-range scanner | Adults & Elderly |
| C | Rural | 8,000 | Low-end portable scanner | Children & Elderly |

| Client (Hospital) | Accuracy (%) |
|---|---|
| A (Urban) | 92% |
| B (Suburban) | 85% |
| C (Rural) | 71% |

$$\text{Accuracy Gap}_{\text{max-min}} = 92\% - 71\% = 21\%$$

# Standard Deviation of Performance

Standard Deviation of Performance: spread of client performance around the mean

$$\sigma_M = \sqrt{\frac{1}{N}\sum_i (M_i - \bar{M})^2}$$

- N: number of clients
- $\overline{M}$: average performance across clients
- Smaller σ → fairer model
- Larger σ → uneven performance

# Variance in Performance

- Measures **how much the model's performance differs across clients or groups**.

It captures the **spread or inconsistency** in how well the global model performs for different participants.

- It's the **statistical variance** of a performance metric (e.g., accuracy, F1, loss) across clients.

$$\mathrm{Var}(M) = \frac{1}{N}\sum_{i=1}^{N}(M_i - \bar{M})^2$$

- N: number of clients (or groups)
- $M_i$: performance metric (e.g., accuracy, F1) for client i
- $\bar{M}$: mean performance across clients
- Low variance → the model performs similarly for all clients (fair, consistent)
- High variance → performance differs greatly (unfair, unbalanced)

# Contribution Fairness Metrics

Each client's **influence** in federated learning should match its **true contribution** to the global model's performance:

1. **Shapley Value:** Fair allocation based on marginal contributions
2. **Gradient similarity-based**: compare update direction
3. **Performance Gain-based**: Measure delta in accuracy/loss

# Shapley Value

| Symbol | Meaning |
|---|---|
| $M_{all}$ | The **global model's performance metric** (e.g., accuracy, F1, loss) when **all clients** participate in training |
| $M_{-i}$ | The same performance metric, but measured when **client ( i ) is excluded** (i.e., the model is trained or aggregated *without* client ( i )'s data) |
| $Contrib_i$ | The **contribution score** of client ( i ): how much performance decreases when it's removed |

$$\text{Contrib}_i = M_{all} - M_{-i}$$

- If removing client i, performance drop: $M_{-i} << M_{all}$: client i is valuable
- If removing client i barely changes performance: $M_{-i} \sim M_{all}$: client has little impact

# Gradient-Based Contribution Metrics

- How each client's update aligns with the global learning direction
- Cosine similarity

$$\text{Contrib}_i = \cos(g_i, g_{\text{global}}) = \frac{g_i \cdot g_{\text{global}}}{\|g_i\| \, \|g_{\text{global}}\|}$$

- Gradient norm contribution

$$\text{Contrib}_i = \frac{\|g_i\|}{\sum_j \|g_j\|}$$

# Bias Mitigation

**Bias mitigation can occur at different stages of the ML pipeline**

## Pre-Processing

**When:** Before training

**Target:** Training data

## In-Processing

**When:** During training

**Target:** Learning algorithm

## Post-Processing

**When:** After training

**Target:** Model predictions

# Pre-Processing Methods

**Transform data to remove bias before model training**

1. **Reweighting:** Assign different weights to training samples
2. **Resampling:** Oversample minority groups or undersample majority
3. **Augmentation & Synthetic Data:** Data augmentation & generate synthetic samples to balance dataset
4. **Fairness-Aware Feature Engineering:** Remove or transform biased features

# Reweighting

- Assign **different weights** to samples to **rebalance** their influence during training.
- Underrepresented or disadvantaged groups should “count more” when optimizing the loss.

$w_i$ - weights set based on group frequencies

A - attribute (e.g. gender)

If 70% are male, 30% female:

$$w_{male} = \frac{1}{0.7} = 1.43, \quad w_{female} = \frac{1}{0.3} = 3.33$$

# Resampling

Modify dataset composition by equalize representation of sensitive groups or classes.

1. **Random Oversampling:** Duplicate minority samples
2. **Random Undersampling:** Remove majority samples
3. **Stratified Sampling:** Preserve class ratios across groups
4. **Hybrid Sampling:** Combine both to avoid overfitting or data loss

# Augmentation & Data Generation

- Augment data
- Generate synthetic samples for underrepresented groups

# Fairness-Aware Feature Engineering

Modify or remove **biased features**

**1. Remove Protected Attributes**

- Drop sensitive features
- The bias can “leak” through correlated variables - correlation information analysis.

**2. Transform Correlated Features**

- Use feature transformations to remove correlations with protected attributes

# In-Processing Methods

**Modify the learning algorithm to incorporate fairness constraints: Fairness**

**Constraints:** Add fairness metrics as constraints to optimization

**Constrained Optimization**

- Standard model training: optimize loss function

$$\min_w L(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(f_w(x_i), y_i)$$

- **Add fairness constraints** to satisfy fairness conditions (e.g., equal opportunity, demographic parity).

$$\min_w L(w) \quad \text{s.t.} \quad \text{FairnessMetric}(w) \leq \delta$$

# Fairness Constraints

## Constrained Optimization

- **Hard Constraint Approach:** Directly enforces fairness conditions during optimization

$$\min_{w} L(w) \quad \text{s.t. fairness metric} \leq \delta$$

- **Soft Constraint Approach:** Adds fairness term as a penalty to the loss

$$\min_{w} L(w) + \lambda \cdot F(w)$$

# Post-Processing Methods

**Adjust model predictions to satisfy fairness criteria:**

1. **Threshold Optimization:** Use different decision thresholds per group
2. **Calibrated Equalized Odds:** Adjust predictions to equalize TPR and FPR
3. **Reject Option Classification:** Flip predictions in uncertain region for fairness

# Threshold Optimization

- Improves equity in model decisions after training, without retraining the model itself.
- A model typically outputs scores $p = f_w(x)$
- Then it is applied a decision threshold $\hat{y} = 1$ if $p > 0.5$ to decide between classes

Standard decision

$$\hat{y} = \begin{cases} 1, & \text{if } f_w(x) > \tau \\ 0, & \text{otherwise} \end{cases}$$

Fair threshold optimization for each group

$$\hat{y}_g = \begin{cases} 1, & \text{if } f_w(x) > \tau_g \\ 0, & \text{otherwise} \end{cases}$$

such that some fairness criterion hold

$$P(\hat{Y} = 1 | A = 0) \approx P(\hat{Y} = 1 | A = 1)$$

# Threshold Optimization

| Level | Description | Example |
|---|---|---|
| **Client-Level Thresholds** | Each client chooses its own local threshold for fairness within its population. | Hospital A (urban) sets $\tau$=0.55, Hospital B (rural) $\tau$=0.45. |
| **Group-Level Thresholds** | Each demographic group gets a threshold shared across clients. | $\tau$_male, $\tau$_female, $\tau$_elderly, etc. |
| **Global Post-Aggregation Thresholds** | Server applies thresholds on global validation data (if available). | Global $\tau$_g tuned to equalize fairness metrics. |

# Calibrated Equalized Odds

Adjust predictions to produce positives at the same rate across groups.

1. Train the model
2. Group calibration
3. Fair adjustment**: randomizes predictions near the decision boundary to equalize both rates smoothly:** flips some predictions probabilistically, to balance **true positive rate** and **false positive rate** across groups.

# Calibrated Equalized Odds

- Global FL model predicts the probability of disease.
- Two groups: younger patients (A=0) and older patients (A=1).

| Group | TPR | FPR | Accuracy |
|---|---|---|---|
| Young | 0.92 | 0.08 | 0.89 |
| Old | 0.70 | 0.03 | 0.86 |

- Bias: The model is less sensitive for older patients
- Adjust predicted probabilities for each group

| Group | TPR (after CEO) | FPR (after CEO) | Accuracy |
|---|---|---|---|
| Young | 0.86 | 0.06 | 0.88 |
| Old | 0.85 | 0.05 | 0.87 |

- Implementation: client level / server level

# Reject Option Classification

- Adjusts predictions only for samples near the decision boundary — the model's zone of uncertainty
- Reassign those uncertain predictions in favor of the disadvantaged group to improve fairness.

# Reject Option Classification

- Define a **confidence interval** around the decision threshold τ (δ is a small margin - uncertain region): [τ−δ,τ+δ]
- For samples **outside of this region**, keep the original prediction
- For samples **inside this uncertain region**, **flip predictions**:
  - If the sample belongs to a **disadvantaged group**, flip 0 → 1 to increase fairness.
  - If it belongs to an **advantaged group**, flip 1 → 0 to decrease unfair advantage.

$$\hat{Y}' = \begin{cases} 1, & \text{if } f_w(x) > \tau + \delta \\ 0, & \text{if } f_w(x) < \tau - \delta \\ 1, & \text{if } A \in \text{disadvantaged group and } f_w(x) \in [\tau - \delta, \tau + \delta] \\ 0, & \text{if } A \in \text{advantaged group and } f_w(x) \in [\tau - \delta, \tau + \delta] \end{cases}$$

A: protected attribute (e.g., gender, ethnicity)

$f_w(x)$: model score

$\hat{Y}=1$ [$f_w(x)>\tau$]: base prediction