Why can a fair algorithm produce unfair outcomes?

The fairness paradox:

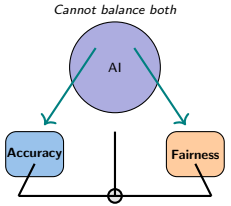
- Algorithms promise objectivity and consistency
- They treat inputs identically every time
- They do not hold personal prejudice
- Yet outcomes systematically disadvantage certain groups

Where fairness breaks down:

- Training data reflects historical discrimination
- Features act as proxies for protected attributes
- Optimization for accuracy ignores disparate impact
- Consistent application of biased patterns is still bias

The core tension:

- Accuracy pushes toward patterns in data
- Fairness pushes toward equal treatment across groups
- Mathematical impossibility: both cannot be maximized
- Choosing fairness definition is a value judgment



Core tension: Optimizing for accuracy often conflicts with fairness. Choosing where to land on the tradeoff curve is a human decision, not a technical one.

Insight

Fair algorithms are not those that treat everyone the same. They are those that account for historical inequality and correct for it.

Have you ever felt a system treated you differently – and could not explain why?

Think about your experiences with automated decisions:

- Have you been rejected for a loan or credit card without clear explanation?
- Have you noticed different prices or offers than your friends for the same product?
- Have you been flagged by a fraud detection system and struggled to understand why?
- Have you felt that a decision was based on factors beyond your control?

What makes algorithmic decisions feel unfair:

- Lack of transparency: the system cannot explain itself
- Lack of recourse: no human to appeal to
- Perceived arbitrariness: slight changes in inputs yield wildly different outputs
- Suspicion of bias: outcomes correlate with demographic groups

The Explainability Gap

When systems cannot explain their decisions, users assume the worst.

Reflection

What are the competing definitions of algorithmic fairness?

Three fundamental fairness metrics:

1. Demographic Parity

- Equal approval rates across all groups
- Outcome-focused: same percentage approved regardless of group
- Ignores differences in qualifications
- Aligns with affirmative action principles
- Formula: approval rate Group A equals approval rate Group B

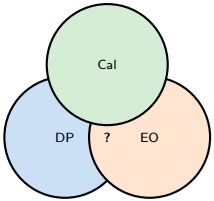
2. Equalized Odds

- Equal error rates across groups given true outcome
- Accuracy-focused: same false positive and false negative rates
- Conditions on actual qualification, not just outcome
- Aligns with meritocratic principles
- Formula: error rates match across groups for qualified and unqualified

3. Calibration

- Predicted probabilities mean the same thing for all groups
- Score-focused: same interpretation of risk scores
- Ensures consistent meaning of predictions
- Aligns with actuarial fairness
- Formula: predicted probability matches observed rate within each group

Metric	Focus
Demographic Parity	Outcome
Equalized Odds	Accuracy
Calibration	Score meaning



The three metrics overlap rarely. Satisfying all three simultaneously is mathematically impossible when base rates differ across groups.

How does a lending algorithm that optimizes for accuracy end up discriminating?

The discrimination pipeline:

Step 1: Historical data encodes bias

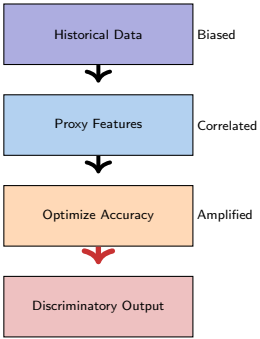
- Model trains on twenty years of past loan decisions
- Past lenders discriminated against certain groups
- Data shows lower approval rates for those groups
- Model learns historical discrimination as ground truth

Step 2: Proxy variables carry the signal

- Protected attributes like race are excluded from model
- But zip code, employer, and university correlate with race
- Model uses proxies to achieve same discriminatory outcome
- Excluding race does not eliminate racial bias

Step 3: Optimization amplifies disparity

- Model optimizes for overall accuracy
- Majority group has more training data
- Accuracy improves more for majority than minority
- Disparity in error rates grows during training



Each step in the pipeline introduces or amplifies bias. Removing one step is insufficient; all three must be addressed.

Insight

Discrimination is not a bug introduced by careless modeling. It is the natural outcome of optimizing for accuracy on biased historical data using correlated features. Fairness requires intervention at every stage.

How do fairness-aware and standard ML pipelines differ in structure?

Standard ML pipeline:

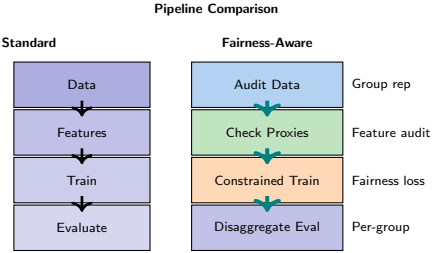
- Collect data and split into train and test sets
- Engineer features without checking correlations
- Train model to maximize overall accuracy
- Evaluate using aggregate metrics only
- Deploy without disaggregated monitoring
- No audit for disparate impact

Fairness-aware ML pipeline:

- Define protected groups before data collection
- Audit training data for representation gaps
- Check features for proxy variable correlations
- Apply fairness constraints during training
- Evaluate accuracy and fairness metrics per group
- Monitor disparate impact in production
- Conduct ongoing third-party fairness audits

Critical differences:

- Standard pipeline measures only accuracy
- Fairness pipeline measures accuracy and fairness together
- Standard pipeline treats all users as one population



Fairness-aware pipelines add auditing and constraints at every stage. Each addition reduces risk of disparate impact.

What happens when a fairness fix for one group harms another?

The fairness tradeoff dilemma:

Phase 1: Detect disparity

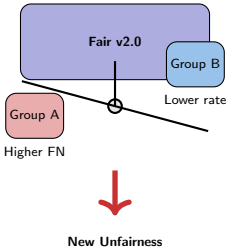
- Model shows higher false positive rate for Group A
- Group A members are incorrectly denied more often
- Fairness audit flags this as unacceptable
- Team commits to fixing the disparity

Phase 2: Apply correction

- Lower decision threshold for Group A to reduce false positives
- False positive rate for Group A drops to match Group B
- But lowering threshold increases false negatives
- More unqualified Group A applicants are now approved

Phase 3: New disparity emerges

- Default rate for Group A rises due to false negatives
- Group B sees their approval rate drop to maintain balance
- Fixing one metric broke another
- No solution satisfies all stakeholders



The impossibility: Fixing false positive disparity creates false negative disparity. No adjustment satisfies all fairness criteria simultaneously.

Insight

Fairness is not a single dial to adjust. It is a set of conflicting constraints. Improving fairness on one dimension often degrades it on another. The impossibility theorem guarantees tradeoffs.

Where do fairness gaps appear most in financial algorithms?

Fairness disparity by application area:

The chart shows disparate impact ratios across different financial domains. Lower values indicate larger fairness gaps.

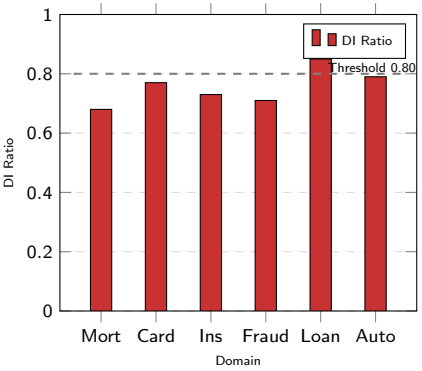
Patterns in the data:

- Mortgage lending shows largest fairness gaps
- Credit card approval shows moderate gaps
- Insurance pricing shows significant demographic disparities
- Fraud detection flags minority groups at higher rates
- Loan servicing varies less across groups

Why disparities persist:

- Historical redlining encoded in zip code data
- Income and wealth gaps from past discrimination
- Thin credit files disproportionately affect minorities
- Proxy variables hard to identify and remove
- Business incentives favor accuracy over fairness

Disparate Impact Ratio by Domain



Domains below the 0.80 threshold (mortgage, insurance, fraud) fail the four-fifths rule and require intervention. Loan servicing and auto perform better.

Insight

Fairness gaps are not uniform across financial products. Mortgage lending suffers most due to historical redlining. Fraud detection suffers from class imbalance. Each domain requires tailored mitigation strategies.

Who decides what fair means – and who is left out of that decision?

Who currently decides:

Technical teams

- Choose which fairness metric to optimize
- Select features and training data
- Set decision thresholds per group
- Often lack diversity and domain expertise

Business stakeholders

- Prioritize profitability over fairness when conflicts arise
- Approve or reject fairness interventions based on cost
- Define acceptable accuracy-fairness tradeoff
- Rarely accountable for discriminatory outcomes

Regulators

- Set minimum fairness standards after harm occurs
- Enforce disparate impact rules retrospectively
- Lack technical capacity to audit complex models
- Operate on slow timelines relative to deployment speed

Who is excluded:

Affected communities

- No voice in defining fairness for systems that judge them
- Discover bias only after experiencing harm
- Lack technical literacy to challenge decisions
- Face barriers to legal recourse

Advocates and ethicists

- Consulted rarely and late in development
- Recommendations overridden by business needs
- Limited enforcement power

Stakeholder	Power
Technical teams	High
Business	Highest
Regulators	Medium
Affected communities	Lowest
Ethicists	Low

Power imbalance means fairness definitions reflect business priorities, not community needs.

Insight

Three tests to evaluate whether a financial algorithm is fair enough to deploy

The Fairness Deployment Checklist

Test 1: Which definition of fairness are you using, and why?

- Have you explicitly chosen demographic parity, equalized odds, or calibration?
- Can you justify why that metric aligns with stakeholder values?
- Have affected communities been consulted on the choice?
- Is the chosen metric documented and auditable?

Test 2: Have you measured disparate impact across protected groups?

- Have you disaggregated accuracy metrics by race, gender, age?
- Does your system pass the four-fifths rule for all groups?
- Have you identified and removed proxy variables?
- Are error rates monitored separately for each protected group?

Test 3: Is there a human appeal process?

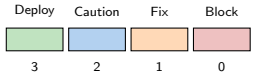
- Can users challenge algorithmic decisions?
- Is there a human reviewer with authority to override?
- Are explanations provided for adverse decisions?
- Is the appeal process accessible and timely?

Deployment Readiness Score

Test	Pass?
Fairness metric chosen	Yes/No
Disparate impact measured	Yes/No
Human appeal exists	Yes/No
Total	**0–3**

Interpretation:

- 3/3: Ready for deployment with monitoring
- 2/3: Deploy with elevated oversight
- 1/3: Return to development for fixes
- 0/3: Do not deploy; high risk of harm



Insight

Fairness is not binary. It is a spectrum from harmful to responsible. The three-test checklist provides a structured way to evaluate

Your Challenge

You have a loan approval model. Group A has a higher approval rate than Group B. Propose three interventions. For each, explain what improves and what might get worse.

Scenario:

- Your model approves seventy-five percent of Group A applicants
- Your model approves sixty percent of Group B applicants
- Disparate impact ratio is 0.80, just at the legal threshold
- Business wants to maintain current approval rates to control risk
- Regulators are scrutinizing your fairness metrics

Your task: Propose three interventions and analyze tradeoffs

Intervention 1: Lower threshold for Group B

- **What improves:** Group B approval rate increases, disparate impact ratio rises above 0.80
- **What worsens:** False positive rate for Group B increases, more defaults expected, profitability drops
- **Who wins:** Qualified Group B applicants who were previously denied
- **Who loses:** Business takes higher losses, unqualified Group B applicants take on debt they cannot repay

Intervention 2: Retrain model with fairness constraint

- **What improves:** Model learns to weight features differently, reducing reliance on proxy variables
- **What worsens:** Overall accuracy drops slightly, development cost increases
- **Who wins:** Both groups see more balanced error rates
- **Who loses:** Business accepts lower profit in exchange for compliance

Intervention 3: Remove proxy features like zip code

- **What improves:** Direct correlation with protected attributes eliminated