Session 4.4: Ethical Considerations in Artificial Intelligence: Algorithmic Bias, Regulatory Compliance, and AI Governance

THE BUSINESS SITUATION

- A consumer lender approves loan applications automatically when its model's approval score is `0.70` or higher
- Before the model's annual review, the board's risk committee must confirm that it treats applicant groups fairly and that every rejection can be explained to the applicant
- The chief risk officer asks: does our credit model treat applicant groups fairly, can we explain each rejection, and what should the board decide? Today, you audit the model for fairness, explain its decisions, and prepare the facts the board needs

---

Today's flow: 60 min lecture, then 120 min hands-on lab

EXPECTED OUTCOMES

By the end of today, you will be able to:

1. ***"Are the groups treated alike?"***
   Measure approval rates by group and the disparate impact ratio, and test it against the four-fifths rule
2. ***"Why was this applicant rejected?"***
   Explain each decision with feature contributions (how much each feature pushed the score up or down)
3. ***"What should the board decide?"***
   Compare one shared threshold's fairness and risk, write a governance record, and fix a `NameError` caused by a variable that only exists inside a function

Session Map: Follow the Numbers

Today's question: *Does our credit model treat applicant groups fairly, can we explain each rejection, and what should the board decide?*

1 Why it matters
Lecture, part 1

1. **Session Purpose**
   answer today's question
2. **Learning Outcomes**
   3 skills you will show today
3. **Business Context**
   Fairness, Explanations, and Accountability
4. **Data Basics**
   Three Fairness Numbers

→

2 Learn the tools
Lecture, part 2

5. **Lab Dataset**
   1 CSV
6. **Keywords**
   6 terms
7. **Core Foundations**
   5 ideas, one at a time
8. **IDE Setup**
   name files, run in the Terminal
9. **Code Examples**
   3 demos + Quick Checks
10. **References**
    4 official docs

→

3 Practice
Lab

11. **Exercise 1: Guided**
    Fairness Audit Report
12. **Exercise 2: Applied**
    A Reusable Audit Function and a Governance Record
13. **Exercise 3: Advanced**
    Can One Shared Threshold Fix the Gap?

Fairness, Explanations, and Accountability

KEY POINTS

- A model learns patterns from past data.
- Two kinds of unfair treatment:
  - **Disparate treatment:** using a person's group directly in the decision, such as a different threshold for each group. In lending, this is prohibited in many countries, including under the US Equal Credit Opportunity Act (ECOA).
  - **Disparate impact:** a rule that treats everyone the same on paper, but approves one group far less often. It is a warning sign that must be investigated and justified.
- The four-fifths rule: if one group's approval rate is less than 80% (four-fifths) of the most-approved group's rate, the gap needs review.
- Explanations are required: US lenders must give rejected applicants the main reasons (an adverse action notice), and the EU AI Act (European Union Artificial Intelligence Act) treats credit scoring of people as high-risk, with duties for documentation, human oversight, and transparency.
- Governance: the board stays accountable for automated decisions.

Three Fairness Numbers

KEY POINTS

- Approval rate of a group = approved applicants in the group / all applicants in the group.
- Disparate impact ratio = the lower group's approval rate / the higher group's approval rate.
- Feature contribution (for a model that adds up features, like linear regression) = coefficient x (applicant's value - average value).
- Our approval scores come from a model whose formula we were not given.

The Data for Today

FILES

- `data/session_04_04_dataset.csv`

Audited 16-row loan application dataset (fictional teaching data), one row per applicant, 8 in each group:

COLUMNS

`Applicant_ID`	Applicant code (e.g., `APP-6001`).
`Demographic_Group`	An anonymized protected group, `Group_A` or `Group_B`. A protected group is defined by a characteristic such as race, sex, or age that the law forbids using against an applicant. The model does not use this column; auditors use it only to test fairness.
`Credit_Score_Count`	Credit score from a credit bureau (a company that records how people repay debts); higher is better.
`Debt_To_Income_Pct`	Monthly debt payments as a share of monthly income (DTI); lower is safer.
`Algorithmic_Approval_Score`	The model's score from 0 to 1; approved at `0.70` or higher.
`Adverse_Action_Notice_Flag`	`1` if the applicant was rejected and must receive an adverse action notice, else `0`.

6 Terms You Need Today

TERM 1

`Disparate Impact`

What: a rule that approves one group much less often than another, even though it never uses the group

Why: the most common way bias enters automated decisions

TERM 2

`Disparate Impact Ratio` and the `Four-Fifths Rule`

What: the lower approval rate divided by the higher one, tested against `0.80`

Why: a simple, widely used first test of fairness

TERM 3

`Disparate Treatment`

What: deciding differently because of a person's group

Why: prohibited in lending in many countries, so it can never be the fix

TERM 4

`Explainability` (feature contributions, SHAP)

What: showing how much each feature pushed a decision up or down

Why: applicants, regulators, and managers need reasons, not just scores

TERM 5

`Surrogate Model`

What: a simple model fitted to copy a model whose formula is hidden

Why: makes a hidden model explainable, as long as the copy is accurate

TERM 6

`AI Governance`

What: the rules, records, and people that keep AI decisions accountable

Why: the board, not the model, answers for the decisions

How It Works

7.1

Approval rates by group

Remember: the mean of a 0/1 column is a rate.

7.2

The disparate impact ratio and the four-fifths rule

Remember: lower rate / higher rate; below `0.80`, investigate.

7.3

Where a gap comes from

Remember: a model can be unfair without using the group column; look at the features.

7.4

Explaining one decision

Remember: contribution = coefficient x (value - average); the largest negative one is the main reason.

7.5

Fair process

Remember: never decide by group; fix the cause, add oversight, and document every decision.

7.1 Approval rates by group: On Real Data

SEE IT ON REAL DATA

Group	Applicants	Approved	Approval rate
`Group_A`	`8`	`7`	`0.875`
`Group_B`	`8`	`2`	`0.250`

Remember: the mean of a 0/1 column is a rate.

Approval Rates and the Four-Fifths Rule

YourName_S44_demo_1.py — Input → Process → Output

```python
# 1. Input: the 16 applications
import pandas as pd
df = pd.read_csv("data/session_04_04_dataset.csv")

# 2. Process: decisions at 0.70, approval rate per group, and the ratio
df["Approved"] = (df["Algorithmic_Approval_Score"] >= 0.70).astype(int)
rates = df.groupby("Demographic_Group")["Approved"].mean()
ratio = rates.min() / rates.max()

# 3. Output: rates, ratio, and the four-fifths test
print(rates)
print(f"Disparate impact ratio: {ratio:.3f}")
print("Passes the four-fifths rule:", ratio >= 0.80)
```

HOW IT WORKS

- `(df["Algorithmic_Approval_Score"] >= 0.70).astype(int)` turns each score into `1` (approved) or `0` (rejected).
- `df.groupby("Demographic_Group")["Approved"].mean()` gives each group's approval rate (section 7.1).
- `rates.min() / rates.max()` divides the lower rate by the higher, whichever group it is (section 7.2).

READ THE OUTPUT

```
Demographic_Group
Group_A    0.875
Group_B    0.250
...
```

- 7 of 8 Group A applicants are approved, but only 2 of 8 in Group B.

Quick Check 1: `.map()`

NEW TOOL

`.map()`

`column.map({1: "Approved", 0: "Rejected"})` replaces each value using a dictionary. Why: tables for the board should use words, not codes.

YOUR TASK

Fill in the Blank

Add two lines to part **3. Output**: `df["Decision"] = df["Approved"].___({1: "Approved", 0: "Rejected"})` and `print(pd.crosstab(df["Demographic_Group"], df["Decision"]))`

Check: Group A: `7` approved, `1` rejected; Group B: `2` approved, `6` rejected.

Where Does the Gap Come From?

YourName_S44_demo_2.py — Input → Process → Output

```
# 1. Input: the 16 applications
import pandas as pd
df = pd.read_csv("data/session_04_04_dataset.csv")

# 2. Process: average features and score by group
profile = df.groupby("Demographic_Group")[
    ["Credit_Score_Count", "Debt_To_Income_Pct", "Algorithmic_Approval_Score"]
].mean().round(2)

# 3. Output: the two group profiles
print(profile)
```

HOW IT WORKS

- `df.groupby("Demographic_Group")[[...]].mean()` averages the two features and the score in each group (section 7.3).
- The group column is used here only by the auditor, to compare groups; the model never sees it.

READ THE OUTPUT

- The `...` hides the debt-to-income column: Group A `26.58`%, Group B `36.05`%.

Quick Check 2: `.T`

NEW TOOL

`.T`

`table.T` flips a table so that rows become columns. Why: with many measures and few groups, the flipped table is easier to read in a report.

YOUR TASK

Fill in the Blank

In part **3. Output**, change the line to `print(profile._)`

Check: the measures are now rows and `Group_A` and `Group_B` are the two columns.

Explain Each Decision

YourName_S44_demo_3.py — Input → Process → Output

```
# 1. Input: the 16 applications and the two features
import pandas as pd
from sklearn.linear_model import LinearRegression
df = pd.read_csv("data/session_04_04_dataset.csv")
features = ["Credit_Score_Count", "Debt_To_Income_Pct"]

# 2. Process: copy the score with a linear model, then contributions
surrogate = LinearRegression().fit(df[features], df["Algorithmic_Approval_Score"])
contributions = (df[features] - df[features].mean()) * surrogate.coef_
contributions.columns = ["Credit_Score_Effect", "Debt_Ratio_Effect"]
report = pd.concat([df[["Applicant_ID", "Algorithmic_Approval_Score"]],
                    contributions.round(3)], axis=1)

# 3. Output: how well the copy fits, then the explanations
print("Surrogate R-squared:", round(surrogate.score(df[features], df["Algorithmic_Approval_Score"]), 3)
print(report[df["Algorithmic_Approval_Score"] < 0.70])
```

HOW IT WORKS

- `LinearRegression().fit(df[features], df["Algorithmic_Approval_Score"])` fits a simple model that copies the scores from the two features: the surrogate (section 7.4).
- `(df[features] - df[features].mean()) * surrogate.coef_` computes every applicant's contribution for both features at once: each column's average is subtracted, and each column is multiplied by its own coefficient.
- `contributions.columns = [...]` gives the two results readable names.

READ THE OUTPUT

```
Surrogate R-squared: 0.975
   Applicant_ID  ...  Debt_Ratio_Effect
1      APP-6001  ...             -0.021
...
```

- The surrogate explains `97.5` % of the score differences, so its explanations can be trusted.

Quick Check 3: `.idxmin(axis=1)`

NEW TOOL

`.idxmin(axis=1)`

`contributions.idxmin(axis=1)` returns, for each row, the column with the most negative value. Why: that column is the main reason to print on the adverse action notice.

YOUR TASK

Fill in the Blank

Add one line at the end of part **2. Process**: `report["Main_Reason"] = contributions._____(axis=1)`

Check: the rejected applicants all show `Credit_Score_Effect` as the main reason.

Where to Look Things Up

REFERENCE

European Commission: Regulatory Framework for Artificial Intelligence (EU AI Act)

https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai

REFERENCE

US Consumer Financial Protection Bureau: Equal Credit Opportunity Act (Regulation B)

https://www.consumerfinance.gov/rules-policy/regulations/1002/

REFERENCE

Fairlearn: A Python Toolkit for Assessing and Improving Fairness in AI

https://fairlearn.org/

REFERENCE

Barocas, Hardt, Narayanan: Fairness and Machine Learning: Limitations and Opportunities

https://fairmlbook.org/

Three Exercises

EXERCISE 1: GUIDED

Fairness Audit Report

Goal: Audit approval rates at the current threshold

File:

`YourName_S44_governance.py`

Check: ratio `0.286`: below the four-fifths rule

EXERCISE 2: APPLIED

A Reusable Audit Function and a Governance Record

Goal: Turn the audit into a function and save a board record

File:

`YourName_S44_practice.py`

Check: `Status`: `Board review required`

EXERCISE 3: ADVANCED

Can One Shared Threshold Fix the Gap?

Goal: Can one shared threshold fix the gap?

File:

`YourName_S44_challenge.py`

Check: only `0.45` and `0.50` reach `0.8` or more

Full steps for all three exercises: sections 11-13 of the course page.