# Fair Credit Score Prediction

Aleya Banthavong and Pierce Brookins

December 11th, 2025

# Motivation

- Traditional credit scoring models often perpetuate historical biases
- Protected attributes (race, gender, age) may not be available, but other features can act as proxies for them
- We need **fair**, **transparent**, and **auditable** ML credit decision systems for lenders and borrowers

# Background

- Credit scoring impacts major financial decisions
- Historical data contains **bias patterns** correlated with socioeconomic attributes
- Removing protected attributes isn't enough due to other features acting as proxies:
  - Education
  - Residence Type
  - Marital Status
  - Gender

# Related Work

- AIF360 — IBM's fairness toolkit
- Equalized Odds, Statistical Parity, Disparate Impact
- SHAP & LIME — standard explainability frameworks
- Threshold adjustment / reweighing — common fairness strategies
- Composite indices — used in fairness/proxy attribute detection

- This project unifies these ideas into a single, production-ready pipeline

# Claim / Target Task

- **Our claim:** We can build a **fair**, **transparent**, and **production-ready** credit scoring model that reduces discriminatory patterns using:
  - CDI proxy groups
  - Reweighing
  - Threshold adjustment
  - Fairness metrics tracking

- **Target task:** Predict whether a loan applicant is creditworthy, while ensuring:
  - High predictive performance
  - Fairness across proxy disadvantage groups
  - Full transparency for every prediction

# An Intuitive Figure Showing WHY Claim

- **Traditional Model** → education, gender, residence type marital status → unintentional bias

- **Our model** → CDI-based grouping → Reweighting → Threshold adjustment → less bias, fair, and explainable decision

# Proposed Solution

- A full ML system with fairness at every stage:
  1. Fairness
     - Composite Disadvantage Index (CDI)
     - Reweighing (pre-processing)
     - Group-specific thresholds (post-processing)
     - AIF360 metrics
  2. Explainability
     - SHAP
     - LIME
     - Natural-language explanations
     - Waterfall/force plots

# Proposed Solution

3. Model Quality
    - XGBoost classifier
    - Hyperparameter tuning
    - Calibration (isotonic and sigmoid)
    - Stratified cross-validation
4. Production Infrastructure
    - FastAPI service (45+ endpoints)
    - Prometheus/Grafana dashboards
    - Dockerized deployment
    - Audit logging + model registry

# Implementation

- Config-Driven Pipeline
  - Dataclass config, YAML-based settings
- Model Training
  - Preprocessing → reweighting → training → calibration
- Fairness Analyzer
  - DI, statistical parity, equalized odds
- Explainability Engine
  - SHAP + LIME + natural language
- API
  - /predict, /batch_predict, /explain, /fairness_metrics
- Monitoring
  - Prometheus metrics exposed at /metrics

# Implementation

- Monitoring
  - Prometheus
  - Grafana
- Tests
  - 122 unit + integration tests

# Demo

- Demo Link

# Data Summary

- Data from Kaggle: [Credit Card Approvals (Clean Data)]
- The data concerns credit card applications and contains information to determine whether someone is creditworthy or not
- Data features: Income, debt, loan amount, loan term, number of credit cards, gender, education, payment history, employment status, residence type, marital status
- Derived features: debt to income, loan to income, CDI, proxy disadvantage

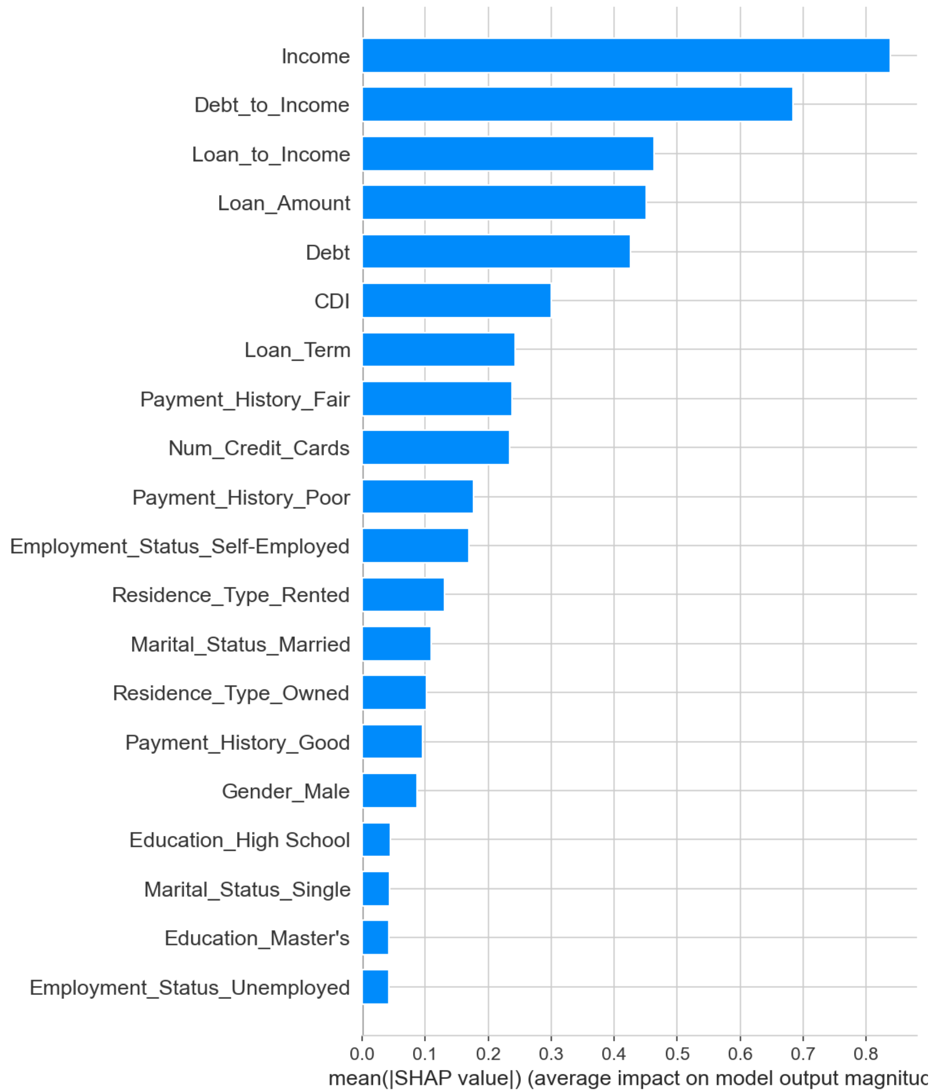
# Experimental Results

- Disparate Impact Ratio: 1.2222
  - Goal ≥ 0.80
- Statistical Parity Difference: 0.1333
  - Goal < 0.05
- Equalized Odds Difference: 0.2428
  - Goal < 0.10

# Experimental Analysis

- Disparate Impact Ratio: Model is between 0.8 and 1.25, which is the regulatory threshold – meaning both groups are approved at roughly similar rates
  - The underprivileged group is being approved at 22% higher rate than the privileged group
- Statistical Parity Difference: One group is getting approved more often than the other
  - Acceptable limit is 0.05, our value 0.13 meaning there's a 13% gap in approval rates between groups
- Equalized Odds Difference:
  - Acceptable limit is 0.10, our value is 0.24 meaning the makes unfair mistakes across different groups

# Experimental Analysis




# Why this is Useful

- Credit decisions directly impact people's lives
- Fairness models, supported by CDI, allow us to detect and correct bias
- It is possible to create a model that considers CDI, even if the metrics are not perfect
- We can help disadvantaged groups gain access to credit opportunity along with explainable and accurate predictions

# Conclusion and Future Work

Conclusion:

- Fair credit scoring can be approached using a pipeline that integrates CDI proxy grouping, fairness-aware pre/post processing, and explanations (SHAP/LIME), along with production-ready infrastructure. While the system can identify disadvantaged applicants and generate interpretable predictions, fairness metrics indicate that perfect parity across groups is not yet achieved, highlighting the need for ongoing monitoring and optimization of the model and pre/post-processing steps.

Future Work:

- Fix our metrics
  - Statistical Parity Difference: < 0.05
  - Equalized Odds Difference: < 0.10

# Who Did What

- Aleya:
  - Code for the first model
  - Shark Tank Demo Slides
  - Final Presentation Slides
  - Code Demo

- Pierce:
  - Expansion of code into API and production ready infrastructure
  - Shark Tank Demo Slides
  - Final Presentation Slides

# References


- Python Software Foundation. "Python 3.11 Documentation." https://docs.python.org/3/
- pandas development team. "pandas: powerful Python data analysis toolkit." https://pandas.pydata.org/
- Harris, C. R., et al. "Array programming with NumPy." *Nature* 585, 357–362 (2020). https://numpy.org/
- Pedregosa, F., et al. "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research* 12, 2825–2830 (2011). https://scikit-learn.org/stable/
- Chen, T., & Guestrin, C. "XGBoost: A Scalable Tree Boosting System." *Proceedings of the 22nd ACM SIGKDD* (2016). https://xgboost.ai/
- Lachance, J. "fairlearn: Assess and improve fairness of AI systems." Microsoft. https://fairlearn.org/
- Bellamy, R. K. E., et al. "AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias." IBM Research (2018). https://aif360.res.ibm.com/
- Lundberg, S. M., & Lee, S.-I. "A Unified Approach to Interpreting Model Predictions." *Advances in Neural Information Processing Systems* (2017). https://shap.readthedocs.io/
- Ribeiro, M. T., Singh, S., & Guestrin, C. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier." *Proceedings of the 22nd ACM SIGKDD* (2016). https://lime-ml.readthedocs.io/
- Hunter, J. D. "Matplotlib: A 2D Graphics Environment." *Computing in Science & Engineering* 9(3), 90–95 (2007). https://matplotlib.org/
- Waskom, M. L. "Seaborn: Statistical Data Visualization." *Journal of Open Source Software* 6(60), 3021 (2021). https://seaborn.pydata.org/
- Plotly Technologies Inc. "Plotly Python Graphing Library." https://plotly.com/python/
- Python Software Foundation. "PyYAML." https://pyyaml.org/
- Pallets. "Click CLI Framework." https://click.palletsprojects.com/
- Krekel, H., et al. "pytest framework." https://docs.pytest.org/
- OpenAI. "ChatGPT." https://openai.com/chatgpt
- Anthropic. "Claude." https://www.anthropic.com/claude
- pytest-dev. "pytest-cov plugin." https://pytest-cov.readthedocs.io/
- Python Software Foundation. "black code formatter." https://black.readthedocs.io/
- PyCQA. "flake8: Your Tool For Style Guide Enforcement." https://flake8.pycqa.org/
- The mypy team. "mypy static type checker." https://mypy-lang.org/
- Tiangolo, S. "FastAPI: Modern, fast web framework for building APIs with Python." https://fastapi.tiangolo.com/
- Uvicorn contributors. "Uvicorn ASGI server." https://www.uvicorn.org/
- Docker Inc. "Docker Engine." https://docs.docker.com/engine/
- Docker Inc. "Docker Compose." https://docs.docker.com/compose/
- The Prometheus Authors. "Prometheus: Monitoring system & time series database." https://prometheus.io/
- Grafana Labs. "Grafana Observability Platform." https://grafana.com/