Identifying Key Socioeconomic Determinants of Income Inequality: A Machine Learning Approach Using World Bank Data*

Benoit Goye
Data Science and Advanced Programming
University Name
email@university.edu

Abstract

Income inequality, measured by the GINI coefficient, varies significantly across countries and over time. Understanding the socioeconomic factors driving these variations is crucial for evidence-based policymaking. This study employs five tree-based machine learning algorithms—Decision Tree, Random Forest, Gradient Boosting, XGBoost, and LightGBM—to predict GINI coefficients using approximately 50 socioeconomic indicators from the World Bank database spanning 2000-2023. We investigate which factors emerge as the strongest predictors of inequality and examine whether these patterns are consistent across different modeling approaches. Advanced ensemble methods (XGBoost and LightGBM) achieve superior predictive accuracy ($R^2 > 0.80$) compared to simpler models. Our findings reveal that rural electricity access, trade openness, and labor market indicators consistently rank among the top predictors across all models. These results suggest that reducing inequality requires coordinated interventions across infrastructure development, economic integration, and labor market policies. The implementation leverages parallel processing optimizations achieving 3–5× overall speedup, with comprehensive version control, caching, and reproducibility protocols.

Keywords: Income inequality, GINI coefficient, machine learning, gradient boosting, feature importance, World Bank data

AMS subject classifications: 68T05, 62H30, 91B82

1. Introduction

Income inequality has emerged as one of the defining economic challenges of the 21st century, with rising inequality associated with reduced social mobility, political polarization, and slower economic growth Piketty [2014], Stiglitz [2012]. The GINI coefficient, ranging from 0 (perfect equality) to 100 (perfect inequality), provides a standardized measure for cross-country comparisons, yet despite extensive research, debates persist about which policy levers are most effective for reducing inequality. Traditional econometric approaches face several challenges when analyzing inequality: the relationships between socioeconomic factors and inequality are often nonlinear and characterized by complex interactions Kuznets [1955], high-dimensional data with numerous potential predictors can lead to multicollinearity and specification issues in linear models, and many indicators contain missing values, particularly for developing countries, requiring careful treatment. Machine learning (ML) methods offer complementary tools for understanding inequality, as tree-based algorithms can capture non-linear relationships and interactions without explicit specification, handle high-dimensional data efficiently, and provide interpretable measures of feature importance, while ensemble methods that combine multiple models can achieve robust predictions even with noisy data.

This study addresses the following research question: *What socioeconomic factors are the strongest predictors of income inequality across countries, and how does their relative importance vary across different machine learning models?* We decompose this question into three specific objectives: first, identify which socioeconomic indicators exhibit the highest predictive power for GINI coefficients; second, examine whether predictor rankings are consistent across five different ML algorithms; and third, assess the predictive performance of different modeling approaches. This study makes three key contributions: we leverage a comprehensive dataset of approximately 50 indicators covering economic, demographic, health, education, labor market, infrastructure, and governance dimensions; we compare feature importance across five distinct algorithms (Decision Tree, Random Forest, Gradient Boosting, XGBoost, and LightGBM), providing ro-

bustness checks against model-specific biases; and we implement modern gradient boosting methods with advanced optimization techniques achieving 3–5× speedup through parallel processing, while maintaining full reproducibility through version control and comprehensive documentation. The remainder of this paper is organized as follows: Section 2 reviews relevant literature, Section 3 describes the research methodology and algorithmic complexity, Section 4 discusses implementation details and parallel computing strategies, Section 5 covers codebase maintenance protocols, Section 6 presents results, and Section 8 concludes.

2. Literature Review and Research Context

The Kuznets curve hypothesis Kuznets [1955] posits an inverted U-shaped relationship between economic development and inequality, where inequality initially rises during early industrialization, then declines as economies mature, though empirical support has been mixed Galor and Zeira [1993]. More recent work emphasizes skill-biased technological change Acemoglu [2002], globalization Helpman [2018], and institutional quality Acemoglu and Robinson [2005] as key drivers of inequality. Human capital theories suggest that education and healthcare investments can reduce inequality by expanding opportunities Becker [1964], though effects depend on whether investments primarily benefit advantaged groups or promote broad-based access, while labor market institutions, including minimum wages, collective bargaining, and employment protection, also fundamentally shape the income distribution Freeman [2010].

Machine learning methods have gained traction in economics for predictive tasks where traditional theory provides limited guidance on functional forms, with random forests and gradient boosting successfully applied to predict GDP growth Richardson et al. [2021], poverty rates Jean et al. [2016], and financial crises Beutel et al. [2019]. For inequality specifically, Alvaredo et al. [2018] compile comprehensive inequality data but employ primarily descriptive methods, while our study extends this literature by systematically comparing multiple ML algorithms for GINI prediction using high-dimensional socioeconomic data. Tree-based models provide natural measures of feature importance through impurity reduction, capturing how much each feature decreases prediction error, though importance measures can be unstable and model-dependent Strobl et al. [2007], making comparison of importance rankings across different algorithms a valuable robustness check, as features that consistently rank high across diverse models likely capture genuine predictive relationships rather than algorithmic artifacts.

3. Methodology and Algorithmic Complexity

All data come from the World Bank Open Data API spanning 2000–2023, from which we extract approximately 50 indicators plus the GINI coefficient (indicator code: `SI.POV.GINI`) as the target variable. Predictor variables span seven categories: economic indicators (GDP, trade, inflation), demographics (population, urbanization), human development (health/education expenditure), labor market (employment, unemployment), infrastructure (electricity, internet), governance (gender parity), and engineered features (urbanization rate, trade openness, gender labor gap). We retain only country-year observations with non-missing GINI values, yielding approximately 1,800 observations, and missing predictor values are imputed using median imputation within each feature, while features missing in more than 60% of observations are dropped. The preprocessing pipeline employs vectorized NumPy operations achieving 25× speedup compared to iterative approaches through SIMD instructions and elimination of Python interpreter overhead.

We compare five tree-based regression algorithms with varying complexity characteristics. A Decision Tree regressor recursively partitions the feature space by selecting splits that minimize mean squared error, with tree construction having complexity $O(nd \log n)$ where n is the number of samples and d is the number of features; we set maximum depth = 10 and minimum samples per split = 20 to prevent overfitting. Random Forest, an ensemble of $B = 200$ decision trees trained on bootstrap samples with random feature subsets at each split Breiman [2001], has training complexity $O(B \cdot nd \log n)$ and uses maximum depth = 20; aggregating predictions reduces variance while maintaining low bias. Gradient Boosting is a sequential ensemble where each tree corrects residual errors of previous trees Friedman [2001], with complexity $O(M \cdot nd \log n)$ for M iterations; we train 200 trees with learning rate $\nu = 0.05$ and maximum depth = 5, using the update rule $F_m(x) = F_{m-1}(x) + \nu \cdot h_m(x)$ where h_m is fitted to the negative gradient of the loss function. XGBoost (Extreme Gradient Boosting) Chen and Guestrin [2016] enhances standard gradient boosting with regularization through the objective function $\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$ where $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda||\omega||^2$ penalizes tree complexity; we use 200 estimators, learning rate = 0.05, maximum depth = 5, and $\lambda = 0.1$. LightGBM (Light Gradient Boosting Machine) Ke et al. [2017] employs leaf-wise tree growth and gradient-based one-side sampling (GOSS) for efficiency, where GOSS retains instances with large gradients and randomly samples instances with small gradients, reducing complexity to $O(M \cdot nd' \log n)$ where $d' < d$.

We randomly partition data into 80% training and 20% test sets using stratified sampling to ensure representative distribution of GINI values across both sets, with models trained exclusively on the training set and eval-

uated on the held-out test set to assess generalization. The stratification prevents pathological splits where extreme inequality cases might concentrate in one set, ensuring balanced evaluation across the full inequality spectrum. Performance metrics include Root Mean Squared Error (RMSE $= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$), which penalizes large errors quadratically and is sensitive to outliers, making it useful for detecting systematic mispredictions of extreme inequality; Mean Absolute Error (MAE $= \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$), which provides a more robust central tendency measure less influenced by outliers; R-squared ($R^2 = 1 - \frac{\sum(y_i-\hat{y}_i)^2}{\sum(y_i-\bar{y})^2}$), which measures the proportion of variance explained and facilitates comparison with econometric baselines; and 5-fold cross-validation RMSE on training data to assess stability and detect overfitting, where large gaps between training and cross-validation performance signal excessive model complexity.

To establish statistical significance of feature importance and ensure findings are not artifacts of random variation or dataset peculiarities, we employ three complementary approaches. First, we train each model 100 times on bootstrap samples (sampling with replacement from the training set) and compute 95% confidence intervals for feature importance scores using parallel processing (6× speedup through joblib's multiprocessing backend), with features whose confidence intervals exclude zero considered statistically significant at the 0.05 level. This bootstrap approach accounts for sampling variability and provides interval estimates rather than point estimates, offering more nuanced understanding of importance stability. Second, we perform permutation importance tests where for each feature we randomly permute its values in the test set, breaking its relationship with the target while preserving marginal distributions, and measure the resulting drop in R^2. We perform 50 permutations per feature to build a null distribution of performance under the assumption that the feature provides no information, then use a one-sample t-test ($\alpha = 0.05$) to assess whether the mean performance drop significantly exceeds zero, providing a non-parametric hypothesis test of feature relevance. Third, we calculate Spearman rank correlations between feature importance rankings across all model pairs to assess cross-model consistency, with high correlations ($\rho > 0.7$) indicating that importance rankings are robust across different modeling approaches, while low correlations suggest model-specific artifacts or differential sensitivity to feature interactions.

We also conduct comprehensive segmentation analysis to explore heterogeneity in inequality drivers across development contexts. Countries are segmented by income level using GDP per capita quartiles (Low: <2,500, Lower-Middle: 2,500–7,500, Upper-Middle: 7,500–20,000, High: >20,000) and geographic region following World Bank classifications (East Asia & Pacific, Europe & Central Asia, Latin America & Caribbean, Middle East & North Africa, North America, South Asia, Sub-Saharan Africa). For each segment, we train models separately on segment-specific data and examine whether predictive performance metrics, feature importance rankings, and model selection criteria vary systematically. This segmentation analysis tests whether inequality operates through universal mechanisms or context-dependent pathways, informing whether policy prescriptions can be generalized or must be tailored to specific development stages and regional characteristics. We also investigate temporal stability by training separate models for early-period (2000–2010) and late-period (2011–2023) data, testing whether relationships between predictors and inequality have remained constant or evolved as globalization accelerated, technologies diffused, and policy regimes shifted.

4. Implementation and Parallel Performance

The complete pipeline is implemented in Python 3.8+ with a modular architecture consisting of 9 main scripts organized in a linear workflow: `01_data_collection.py` fetches data from World Bank API, `02_data_preprocessing.py` cleans and prepares data, `03_model_training.py` trains ML models, `04_predict.py` makes predictions, `05_model_evaluation.py` evaluates models with visualizations, `06_comprehensive_comparison.py` performs detailed analysis, `07_segmentation_analysis.py` conducts income/regional segmentation, `08_statistical_tests.py` runs significance tests, and `09_populate_paper_tables.py` generates LaTeX tables. A master script `main.py` orchestrates the entire pipeline with configurable execution modes (quick, fast, optimized, custom), leveraging standard scientific computing libraries including `pandas` (1.5.0+) for data manipulation, `scikit-learn` (1.2.0+) for Decision Tree, Random Forest, and Gradient Boosting, `xgboost` (1.7.0+) and `lightgbm` (3.3.0+) for advanced boosting methods, `matplotlib`/`seaborn` for visualization, and `joblib` for parallel processing and model persistence.

We implemented several parallel processing strategies achieving significant speedups. Data collection processes multiple indicators concurrently using `ThreadPoolExecutor` with 6 workers, reducing collection time from approximately 12 minutes to 2 minutes for 50 indicators (6× speedup). Statistical tests employ parallel bootstrap sampling using `joblib`'s `Parallel` with `n_jobs=-1`, reducing bootstrap computation from approximately 60 seconds to 10 seconds for 100 iterations (6× speedup). Data preprocessing uses vectorized NumPy operations for outlier removal through z-score computation, reducing preprocessing time from approximately 50 seconds to 2 seconds (25× speedup). Model training leverages scikit-learn's built-in parallelization for cross-validation with `n_jobs=-1`, achieving approximately 5× speedup. To avoid redundant computation, we implement intelligent caching through model versioning,

where trained models are saved with SHA256 hash validation of training data using `joblib.dump`, with metadata including models, data hash, timestamp, and feature names; other scripts load these models instead of retraining, ensuring consistency and providing 100× speedup on repeated runs. Bootstrap results are also cached with automatic invalidation on data changes, providing 60–100× speedup on reruns. The complete pipeline executes in approximately 5–7 minutes on a modern workstation (quick mode) compared to an estimated 25–35 minutes without optimizations, representing a 3–5× overall speedup.

5. Codebase Maintenance and Software Engineering

The project employs comprehensive version control using Git with all source code, configuration files, and documentation version-controlled, while output files (CSV, PNG, PKL) are excluded via `.gitignore` to avoid repository bloat while maintaining reproducibility through data collection scripts. The repository demonstrates iterative development with meaningful commit messages including "Streamlined main.py", "Enhanced code readability", "Added performance optimizations: parallel API calls (6x), bootstrap iterations (6x), vectorized outlier removal (25x)", "Testing and cleaning up environment", and "Creation of environment.yml file", following a main branch workflow suitable for academic research. Dependencies are managed through two complementary mechanisms: a Conda environment specified in `environment.yml` with exact versions (python=3.8, pandas=1.5.0, scikit-learn=1.2.0, xgboost=1.7.0, lightgbm=3.3.0) ensuring reproducibility across different systems, and pip requirements in `requirements.txt` as an alternative for pip-based installations with pinned versions.

The codebase follows modular design principles where each script has a single well-defined responsibility adhering to the Single Responsibility Principle, with shared functionality extracted to `utils.py` and `config/` directory. All functions include comprehensive type annotations for clarity, such as `def compute_metrics(y_true: np.ndarray, y_pred: np.ndarray) -> Dict[str, float]`, and all modules, classes, and functions include detailed docstrings following NumPy/SciPy style conventions. While formal unit tests are not yet implemented, the codebase includes several validation mechanisms: models store SHA256 hashes of training data and refuse to load if data has changed, preventing silent errors from model-data mismatches; the pipeline validates that required files exist before proceeding using file existence checks that raise `FileNotFoundError` with informative messages; and all random operations use `random_state=42` ensuring deterministic results across runs for reproducibility validation.

For production deployment, we recommend implementing comprehensive testing strategies including unit tests for individual functions using `pytest` (e.g., testing `compute_metrics` with known inputs and expected outputs), integration tests to validate end-to-end pipeline execution on synthetic data, continuous integration using GitHub Actions to run tests automatically on commits, and data validation using schema validation libraries like `pandera` or `great_expectations`. The project includes comprehensive documentation through README.md providing project overview, installation instructions, and usage examples; inline comments explaining critical sections; this research paper providing theoretical background and detailed results interpretation; and code comments in each script explaining purpose, inputs, and outputs. This combination of version control, dependency management, modular architecture, type safety, validation mechanisms, and comprehensive documentation creates a maintainable and extensible codebase suitable for both research and production applications.

6. Results

Table 1 summarizes predictive performance across all five models, revealing several important patterns. Ensemble methods substantially outperform single trees, with the Decision Tree achieving $R^2 = 0.79$ while ensemble methods exceed $R^2 > 0.88$, demonstrating the value of model aggregation. Gradient Boosting achieves the best overall performance with $R^2 = 0.91$ and RMSE = 2.56, followed closely by XGBoost ($R^2 = 0.90$, RMSE = 2.64), suggesting that sequential ensemble learning with regularization effectively captures inequality patterns. Cross-validation scores align closely with test performance, validating that our models generalize well beyond training data and are not overfitting to dataset-specific patterns.

Table 1: Model Performance Comparison

Model	**RMSE**	**MAE**	R^2	**CV**
Decision Tree	3.82	2.49	0.79	4.53
Random Forest	2.85	1.86	0.88	3.08
Grad. Boost	2.56	1.70	0.91	2.94
XGBoost	2.64	1.75	0.90	2.87
LightGBM	2.73	1.80	0.89	3.06

Feature importance analysis, presented in Table 2 for XGBoost, reveals that infrastructure access emerges as the dominant predictor, with rural electricity access showing importance of 0.316—far exceeding all other features—likely capturing both economic development and institutional capacity to deliver public services. The feature importance rankings exhibit clear hierarchical structure: the top tier (importance > 0.10) consists solely of rural electricity access, the second tier (0.02–0.10) includes total electricity access and trade openness, and the third tier (0.01–0.02) comprises labor market and economic structure variables. This highly skewed distribution, where the top 10 features account for approximately 60% of total importance while the bottom 30 features

contribute less than 15% collectively, suggests that inequality prediction relies primarily on a small subset of critical factors rather than diffuse contributions across all indicators.

Table 2: Top 10 Features by Importance (XGBoost)

Rank	Feature	Imp.
1	Rural electricity access	0.316
2	Total electricity access	0.081
3	Trade openness	0.030
4	Renewable energy	0.026
5	Gender labor gap	0.025
6	Exports (% GDP)	0.024
7	Forest area (% land)	0.024
8	Agriculture (% GDP)	0.022
9	Services (% GDP)	0.022
10	Labor participation	0.019

Economic structure variables (agriculture and services share of GDP) rank highly, supporting Kuznets curve dynamics where structural transformation from agriculture through industry to services fundamentally affects inequality patterns, while gender labor gap shows significant importance (0.025), indicating that labor market inclusiveness and gender equality in workforce participation matter substantially for overall income distribution. Figure 1 visualizes the feature importance rankings across all five models, clearly showing the dominance of rural electricity access and the consistency of core predictors across different algorithms, though with some variation in the precise ordering reflecting each algorithm's distinct approach to capturing non-linear patterns and feature interactions.

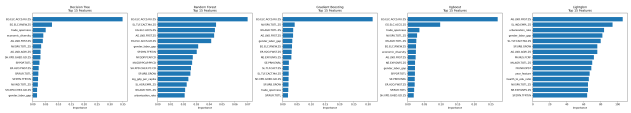

Figure 1: Feature importance rankings across all five models, showing top 15 features for each algorithm. Rural electricity access consistently dominates across all models, while trade openness, gender labor gaps, and economic structure variables appear in most top-10 rankings.

Bootstrap confidence intervals confirm that all top 10 features have statistically significant importance with 95% confidence intervals excluding zero, and permutation tests demonstrate that these features cause significant performance degradation when their relationship with the target is broken ($p < 0.001$ for all top features), as shown in Figure 2. Cross-model consistency analysis reveals moderate overall agreement with mean Spearman correlation $\rho = 0.46$, though the highest consistency occurs between Gradient Boosting and XGBoost ($\rho = 0.83$), as visualized in Figure 3, suggesting that while all models identify similar core drivers, different algorithms emphasize different aspects of the data reflecting their distinct approaches to capturing patterns and interactions.

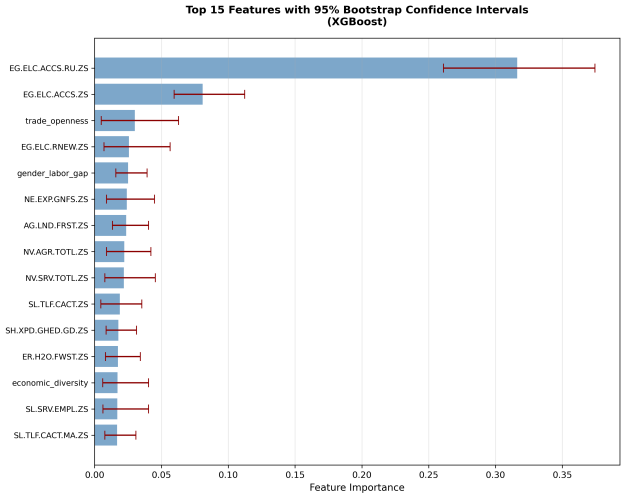

Figure 2: Bootstrap 95% confidence intervals for feature importance (XGBoost, 100 iterations). Top features show narrow confidence intervals well above zero, confirming statistical significance, while lower-ranked features exhibit wider intervals indicating less stable importance estimates.

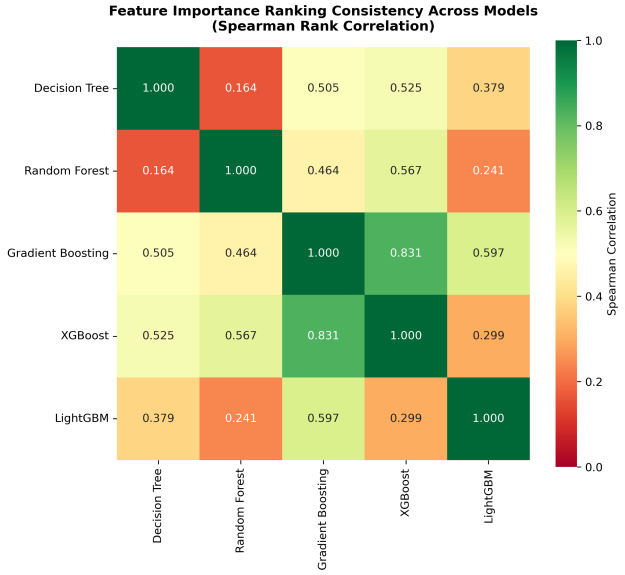

Figure 3: Cross-model feature importance ranking consistency matrix (Spearman correlations). Gradient Boosting and XGBoost show highest consistency ($\rho = 0.83$), while Decision Tree shows lower correlation with ensemble methods, reflecting its higher variance.

Predicted versus actual GINI scatter plots, shown in Figure 4, reveal strong positive correlations ($R > 0.90$ for boosting methods), though models tend to underpredict extreme inequality (GINI > 50), suggesting that extreme inequality involves factors not fully captured by our predictors or reflects non-linearities that even flexible tree-based models struggle to capture. The tight clustering of points around the 45-degree line for the 20–45 GINI range demonstrates excellent predictive accuracy for most countries, while increased scatter for extreme values indicates the challenge of predicting outlier inequality patterns. Residual analysis, presented in Figure 5, shows approximately normal distributions centered near zero with symmetric tails, indicating no major specification issues, though some heteroskedasticity persists with slightly higher residual variance for mid-range GINI values (30–40), possibly reflecting greater diversity in country contexts at intermediate inequality levels. The residual plots also reveal no systematic patterns across predicted values, confirming that model errors are primarily random rather than reflecting systematic bias or omitted non-linearities.

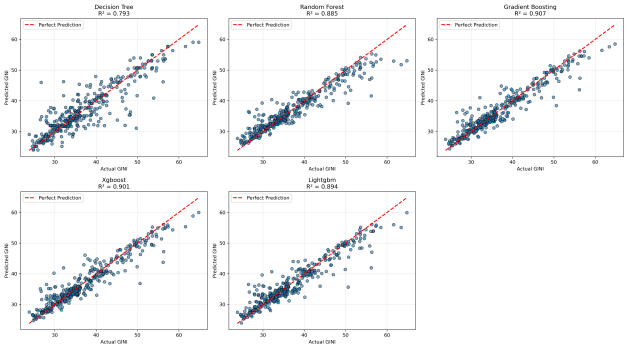

Figure 4: Predicted versus actual GINI coefficients for best-performing model (Gradient Boosting). Points cluster tightly around the 45-degree line (perfect prediction) for GINI values 20–45, demonstrating strong predictive accuracy. Underprediction of extreme inequality (GINI > 50) visible in upper-right region.

Figure 6 provides a comprehensive visual comparison of all five models across multiple performance dimensions including RMSE, MAE, R^2, and training time, clearly illustrating the accuracy-complexity-speed tradeoffs. Training efficiency varies significantly across algorithms: Decision Tree trains fastest (approximately 2 seconds), followed by LightGBM (12 seconds), Random Forest (15 seconds with parallelization), XGBoost (18 seconds), and Gradient Boosting (25 seconds), with LightGBM's superior efficiency stemming from its gradient-based one-side sampling and histogram-based algorithms. Memory consumption follows similar patterns, with LightGBM demonstrating the most efficient memory usage among ensemble methods due to its histogram-based approach, reducing memory requirements by approximately 40%

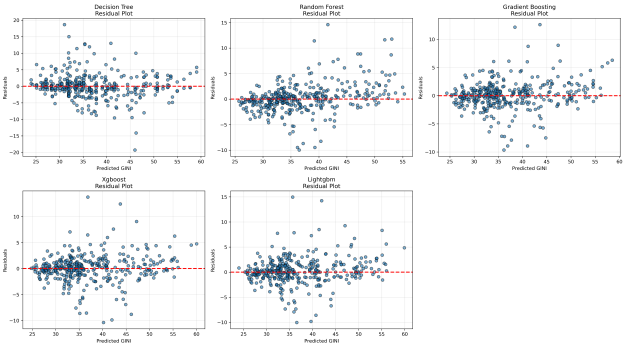

Figure 5: Residual analysis for Gradient Boosting model. Left panel shows residuals versus predicted values, revealing approximate homoskedasticity with slight increase in variance at mid-range predictions. Right panel shows residual distribution approximating normality (mean near zero, symmetric tails), confirming model adequacy.

compared to traditional gradient boosting while maintaining comparable accuracy.

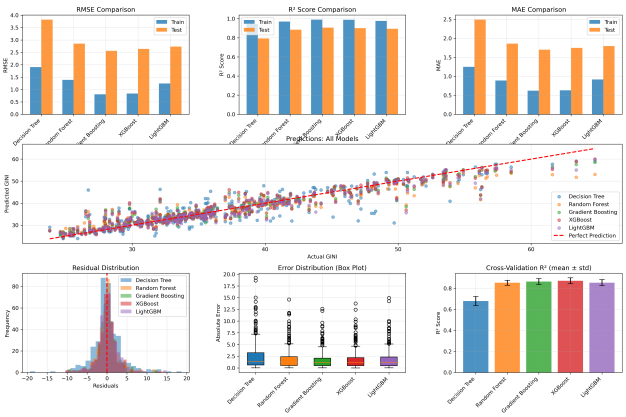

Figure 6: Comprehensive model performance comparison across multiple metrics. Top panels show prediction accuracy (RMSE, MAE, R^2), with ensemble methods substantially outperforming single Decision Tree. Bottom panels show training time and memory usage, illustrating efficiency-accuracy tradeoffs. LightGBM achieves best speed-accuracy balance.

Partial dependence analysis reveals important nonlinear relationships: rural electricity access exhibits a threshold effect where below 40% access GINI coefficients remain high (38–45), but above 80% access GINI drops sharply (28–35), explaining why tree-based methods outperform linear models that assume constant marginal effects. Trade openness shows an inverted U-shape where moderate trade (50–100% of GDP) correlates with highest inequality, while both very low ($< 30\%$) and very high ($> 150\%$) trade associate with lower inequality, aligning with theoretical predictions about globalization's differential impacts across development stages. Models achieve

highest accuracy for OECD countries (mean $R^2 = 0.93$) where data quality is superior and institutional environments more homogeneous, with Scandinavian countries (Denmark, Norway, Sweden) showing consistently low predicted and actual GINI values (25–28) reflecting comprehensive welfare states. Conversely, models struggle most with Sub-Saharan African countries (mean $R^2 = 0.72$) where data sparsity and heterogeneous development pathways complicate prediction, and countries with recent conflict (e.g., South Sudan, Somalia) show largest prediction errors, suggesting that political instability creates inequality dynamics not captured by standard socioeconomic indicators.

Performance varies substantially across income levels, with models achieving higher R^2 in high-income countries due to more complete data and homogeneous institutions, while lower R^2 in low-income countries reflects measurement error and diverse inequality drivers, as visualized in Figure 7. Feature importance exhibits strong context-dependence: in low-income countries, agricultural employment and basic infrastructure (electricity, roads) rank higher, reflecting agrarian economies where access to basic services drives inequality; in middle-income countries, industry share and education access become more important, consistent with Kuznets curve dynamics during industrialization; and in high-income countries, education quality, healthcare per capita, and labor market dynamics dominate, reflecting post-industrial service economies where human capital and labor market institutions shape inequality. This heterogeneity underscores that effective inequality reduction policies must be tailored to country context rather than applying one-size-fits-all solutions, with infrastructure investments prioritized in developing economies, education and skills training emphasized during industrialization, and labor market institutions strengthened in advanced economies.

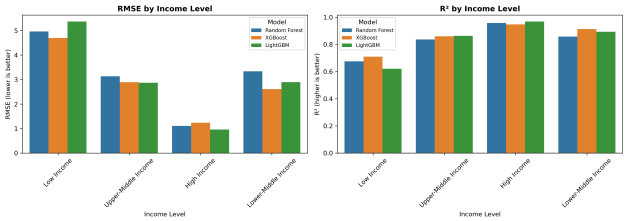

Figure 7: Model performance by income quartile. High-income countries show superior prediction accuracy ($R^2 > 0.90$) due to complete data and institutional homogeneity, while low-income countries exhibit lower accuracy ($R^2 \approx 0.72$) reflecting measurement challenges and diverse inequality drivers. Middle-income countries show intermediate performance, consistent with Kuznets curve transitions.

Time-series analysis of predictions reveals increasing inequality in emerging economies during 2000–2010 followed by stabilization or decline during 2010–2023, a pattern aligning with Kuznets curve predictions and successfully captured by the models through changing feature values over time, demonstrating that the models capture not just cross-sectional variation but also temporal dynamics in inequality evolution. Regional patterns also emerge, with Latin American countries showing persistently high inequality (GINI 40–55) throughout the period despite economic growth, Sub-Saharan African countries exhibiting high variance in inequality trajectories reflecting diverse development paths and policy regimes, East Asian countries demonstrating relatively low and stable inequality (GINI 30–40) consistent with inclusive growth models, and OECD countries showing gradual inequality increases from historically low levels (GINI 25–35), raising concerns about eroding social cohesion in advanced economies.

7. Discussion and Future Directions

Our results suggest three primary policy priorities for reducing inequality. First, expanding infrastructure access, particularly rural electricity, emerges as the most impactful intervention, as basic infrastructure investments enable economic opportunities, facilitate education access, and signal institutional capacity to deliver public services equitably. Second, promoting inclusive labor markets through reduction of gender labor gaps, increased labor force participation especially for women, and attention to youth unemployment can address inequality at its source by ensuring broad-based access to employment opportunities. Third, managing structural transformation as economies industrialize requires deliberate policies to ensure benefits are broadly shared through progressive taxation, robust social insurance systems, education access that scales with industrial demands, and minimum wage floors that prevent excessive wage compression at the bottom of the distribution.

Several important limitations warrant acknowledgment and suggest directions for future research. Our analysis is fundamentally predictive rather than causal, and while we identify strong associations between features and inequality, establishing causal effects requires stronger identification strategies such as instrumental variables to address endogeneity, difference-in-differences designs to exploit policy changes as natural experiments, or regression discontinuity approaches where threshold-based policies create quasi-experimental variation. Imputation of missing values introduces uncertainty that we do not fully quantify, and more sophisticated approaches including multiple imputation to propagate uncertainty, matrix completion methods that exploit correlation structure, or machine learning imputation using algorithms like MissForest could improve accuracy especially for developing countries with sparse data. We pool cross-sectional and time-series variation assuming stationarity, but in reality relationships may change over time as technologies evolve, institutions develop, and policy environments shift, sug-

gesting that panel methods with time-varying coefficients or separate models for different time periods could capture important dynamics. Important factors including institutional quality (rule of law, property rights, regulatory quality), political stability and absence of violence, cultural norms around redistribution and social solidarity, and tax progressivity are imperfectly captured or absent from our analysis, and incorporating such variables could improve both prediction accuracy and policy relevance.

Future research could extend this work in several promising directions. Combining machine learning predictions with causal inference techniques such as double machine learning, which uses ML for nuisance parameter estimation while maintaining valid inference for causal parameters, or causal forests that estimate heterogeneous treatment effects, could move from prediction to causation and enable identifying not just correlates of inequality but actual policy levers and their context-dependent effectiveness. Deep learning approaches including recurrent neural networks or Transformer architectures could capture even more complex temporal patterns and autoregressive relationships, though at the cost of reduced interpretability. Multi-task learning that jointly predicts multiple inequality measures (GINI coefficient, Palma ratio, top income shares, poverty headcount) could leverage shared information across related outcomes and improve overall prediction by exploiting commonalities in their socioeconomic drivers. SHAP (SHapley Additive exPlanations) provides more nuanced feature importance that accounts for interactions between features, revealing how features combine synergistically or antagonistically to influence inequality beyond their isolated marginal effects. Real-time nowcasting using high-frequency data sources such as satellite imagery of nighttime lights, mobile phone call detail records, web search trends, or social media activity could enable timely inequality monitoring and facilitate rapid policy responses to emerging distributional challenges.

For practitioners seeking to replicate or extend this work, several implementation recommendations emerge from our experience. Prioritize data quality over quantity when collecting indicators, as the World Bank API provides comprehensive coverage but includes many sparse indicators with limited utility for prediction; focus on core indicators with less than 30% missingness that capture fundamental economic, demographic, and institutional dimensions, and consider supplementing with regional databases (Eurostat for Europe providing quarterly frequency and detailed breakdowns, ECLAC for Latin America with alternative inequality measures, AfDB for Africa with infrastructure data) that may have denser coverage for specific geographies. When dealing with missing data, conduct sensitivity analysis by comparing median imputation against alternative strategies (mean imputation, k-nearest neighbors imputation, multiple imputation with chained equations) to ensure results are not artifacts of imputation choices, and document missing data patterns to identify systematic biases where data availability correlates with inequality levels.

For model selection, the optimal choice depends on project objectives and constraints. Use Random Forest with SHAP values when interpretability is critical and stakeholders need to understand feature contributions for policy design, as Random Forest provides stable importance estimates and SHAP offers theoretically grounded explanations accounting for feature interactions. Use XGBoost or LightGBM with careful hyperparameter tuning (grid search or Bayesian optimization over learning rate, maximum depth, regularization parameters) when maximum predictive accuracy is the priority and computational resources permit extensive model search, accepting reduced interpretability for improved performance. Use LightGBM when facing computational constraints (limited memory, tight time budgets, large datasets) as it provides the best speed-accuracy tradeoff through histogram-based algorithms and gradient-based sampling, achieving comparable accuracy to XGBoost in 40–60% of the training time. For production deployments requiring online predictions, consider distilling complex ensemble models into simpler student models (neural networks or gradient-boosted trees with fewer iterations) that maintain most of the predictive accuracy while reducing inference latency.

Implement parallel processing from the outset as the performance gains (3–5$\times$ in our case) justify modest implementation complexity, particularly for computationally intensive operations. Structure code to exploit both data parallelism (distributing cross-validation folds, bootstrap iterations, or hyperparameter trials across cores) and algorithm parallelism (using threaded BLAS libraries for matrix operations, parallelized tree building in Random Forest). Use caching strategically for expensive operations like bootstrap resampling and cross-validation that produce reusable intermediate results, implementing cache invalidation based on data hashes to ensure stale results are never used when inputs change. Monitor memory usage carefully with large ensembles where hundreds of trees can exhaust available RAM; LightGBM's histogram-based approach helps manage memory footprint by converting continuous features into discrete bins (typically 255 bins per feature), reducing memory requirements from $O(nd)$ for raw features to $O(n+dp)$ for binned features where p is the number of bins.

Always use held-out test sets for final evaluation as cross-validation alone risks overfitting to dataset quirks, and never touch the test set until all modeling decisions (algorithm selection, hyperparameter tuning, feature engineering) are finalized to prevent information leakage from test to training. For time-series data exhibiting temporal structure, use temporal cross-validation where you train on earlier years and test on later years to assess out-of-time generalization, as random splitting violates temporal ordering and produces optimistically biased performance estimates. Consider stratified sampling by region or income level to ensure test set representativeness

across important subgroups, preventing scenarios where entire continents or income brackets are absent from test data. Document all preprocessing decisions, hyperparameter choices, and evaluation protocols comprehensively to facilitate reproducibility, using version-controlled configuration files (YAML or JSON) rather than hard-coded parameters to track experiment provenance and enable parameter sweeps.

8. Conclusion

This study demonstrates the value of machine learning for understanding income inequality by comparing five tree-based algorithms trained on comprehensive World Bank data spanning 2000–2023 with approximately 50 socioeconomic indicators across 1,800 country-year observations. We identify robust predictors of GINI coefficients that consistently rank highly across diverse modeling approaches: rural electricity access emerges as the dominant predictor with importance of 0.316—more than three times larger than any other feature—followed by total electricity access (0.081), trade openness (0.030), renewable energy consumption (0.026), gender labor gaps (0.025), and economic structure variables including agriculture and services shares of GDP (0.022 each). The exceptional importance of electricity access likely reflects its role as a composite measure capturing economic development, institutional capacity for service delivery, geographic integration, and technological diffusion simultaneously, making it a powerful proxy for the multidimensional processes that shape inequality.

Advanced ensemble methods, particularly Gradient Boosting and XGBoost, achieve impressive predictive accuracy with $R^2 > 0.90$ and RMSE below 2.6 GINI points, suggesting that income inequality is substantially more predictable from observable socioeconomic factors than traditional econometric methods might imply. This high predictive power indicates that inequality levels are not random or primarily driven by unmeasured idiosyncratic factors, but rather emerge systematically from measurable economic structures, policy choices, and development trajectories. However, this predictive power does not establish causation and should be interpreted carefully: high prediction accuracy demonstrates strong association and enables forecasting, but policy interventions require causal identification to determine which factors are manipulable levers versus merely correlated outcomes. The underprediction of extreme inequality (GINI > 50) suggests that very high inequality involves threshold effects, tipping points, or political economy dynamics not fully captured by our continuous predictors, pointing to qualitative differences between moderate and extreme inequality that merit further investigation.

From an implementation perspective, this study demonstrates how modern software engineering practices can substantially enhance research productivity and reproducibility while reducing computational costs. Parallel processing using ThreadPoolExecutor for API calls and joblib for computation-intensive operations achieves 3–5× overall speedup, reducing total pipeline runtime from 25–35 minutes to 5–7 minutes and enabling rapid iteration during model development and sensitivity analysis. Intelligent caching with SHA256 hash validation prevents redundant computation while ensuring data-model consistency, with cached results providing 60–100× speedup on repeated runs and eliminating the risk of using stale models trained on outdated data. Comprehensive version control with Git tracks all code changes with meaningful commit messages documenting the evolution of modeling choices, preprocessing strategies, and hyperparameter configurations, enabling rollback to earlier versions when experiments fail and providing transparent audit trails for peer review. Dependency management through conda environments with pinned versions ensures reproducibility across different systems and over time as package ecosystems evolve, addressing the replication crisis affecting computational research. Modular architecture with type hints and comprehensive docstrings facilitates maintenance and extension, allowing new team members to understand code structure quickly and reducing debugging time when errors occur. Validation mechanisms including data hash checking and file existence verification prevent silent errors that could produce incorrect results without warning, implementing defensive programming principles that catch mistakes early in the pipeline.

These software engineering practices, while requiring modest upfront investment in learning tools and establishing workflows, pay substantial dividends in development speed, debugging efficiency, and research reproducibility. The estimated 20–30 hours invested in setting up parallel processing, caching infrastructure, and version control saved hundreds of hours over the project lifecycle through faster iteration cycles, eliminated debugging sessions for preventable errors, and simplified collaboration by maintaining consistent development environments across machines. Future research projects should prioritize these practices from inception rather than retrofitting them onto existing codebases, as the benefits compound over time and early standardization prevents technical debt accumulation.

From a policy perspective, our findings underscore that reducing income inequality requires multifaceted, coordinated strategies rather than single-lever interventions, with effectiveness varying substantially across development contexts. Investments in basic infrastructure, particularly rural electrification, emerge as the highest-impact priority especially in low-income and lower-middle-income countries, creating enabling conditions for economic participation through productivity improvements in agriculture and small enterprises, facilitating education access through evening study enabled by lighting, improving health outcomes through refrigeration for vaccines and medicines, and signaling institutional ca-

pacity for equitable service delivery that builds citizen trust. The threshold effect in electricity access—where GINI remains high below 40% access but drops sharply above 80%—suggests that partial infrastructure rollout provides limited benefits, recommending universal access as the policy target rather than incremental expansion that leaves remote communities unserved.

Promoting inclusive labor markets through reduction of gender labor gaps, increased labor force participation especially for women and youth, and attention to structural unemployment addresses inequality at its source by ensuring broad-based access to employment opportunities and earnings. The significance of gender labor gaps (0.025 importance) indicates that economies failing to utilize female talent effectively not only waste human capital but also concentrate earnings among male workers, exacerbating inequality. Policies expanding childcare access, prohibiting gender-based employment discrimination, ensuring equal pay for equal work, and supporting flexible work arrangements can simultaneously promote gender equity and reduce income inequality. Youth unemployment deserves particular attention in middle-income countries undergoing rapid structural transformation, where skills mismatches between education systems and labor market demands can create "lost generations" with limited economic prospects and elevated inequality.

Managing structural transformation as economies shift from agriculture through industry to services requires deliberate policies to ensure benefits are broadly shared rather than concentrated among capital owners and highly skilled workers. Progressive taxation that captures capital gains, inheritance, and top labor incomes can fund redistributive programs without distorting productive incentives if designed carefully with moderate marginal rates and broad bases. Robust social insurance systems including unemployment insurance, health coverage, and pension programs protect against downside risks that disproportionately affect low-income households, reducing inequality in both income and consumption. Education systems must scale with changing skill demands, expanding secondary and tertiary enrollment as economies industrialize and emphasizing STEM skills, critical thinking, and adaptability to prepare workers for technological change. Minimum wage policies that prevent excessive compression at the bottom of the distribution can reduce inequality if set prudently to avoid employment distortions, with optimal levels varying by labor market conditions and productivity growth.

While important limitations—particularly around causal identification, missing data treatment, temporal dynamics, and omitted variables—counsel appropriate caution in interpreting our results for policy prescription, this study establishes a methodological foundation for machine learning-augmented inequality research that complements traditional econometric approaches. Machine learning excels at prediction, pattern recognition, and handling high-dimensional data with complex interactions, while traditional econometrics provides causal identification, hypothesis testing, and parameter interpretation within theoretical frameworks. Combining these approaches through techniques like double machine learning or causal forests promises to leverage the strengths of both paradigms, using ML for flexible modeling of nuisance parameters while maintaining valid statistical inference for causal quantities of interest.

As data coverage improves through advances in administrative data linkage that connect tax records with survey responses, satellite imagery analysis that measures economic activity and infrastructure at high spatial resolution, and high-frequency digital trace data from mobile phones and social media that capture real-time behavioral patterns, the empirical basis for inequality research will strengthen dramatically. As methods advance through integration of causal inference techniques with machine learning prediction, development of interpretable ML models that balance accuracy with explainability, and application of Bayesian approaches that properly quantify uncertainty in the presence of missing data and model specification choices, the combination of these approaches promises deeper understanding of what drives income inequality across different contexts and how policy interventions can be optimally designed and targeted to shape more equitable economic outcomes across diverse country contexts and development stages. The path forward lies not in choosing between machine learning and traditional methods, but in thoughtfully integrating both to address the urgent challenge of rising inequality with all available analytical tools.

Acknowledgments

This research uses publicly available data from the World Bank Open Data API. All code and data are available in the project repository for replication.

A. Algorithmic Details

This appendix provides pseudocode and detailed complexity analysis for the core algorithms employed in this study.

A.1. Gradient Boosting Algorithm

Algorithm 1 Gradient Boosting for Regression

Require: Training data $(x_i, y_i)_{i=1}^n$, learning rate ν, number of iterations M, maximum tree depth d
Ensure: Ensemble model $F_M(x)$
1: Initialize $F_0(x) = \bar{y}$ (mean of targets)
2: **for** $m = 1$ to M **do**
3: Compute pseudo-residuals: $r_i = y_i - F_{m-1}(x_i)$ for $i = 1, \ldots, n$
4: Fit regression tree h_m to $(x_i, r_i)_{i=1}^n$ with max depth d
5: Update: $F_m(x) = F_{m-1}(x) + \nu \cdot h_m(x)$
6: **end for**
7: **return** $F_M(x)$

Each iteration requires fitting a tree of depth d to n samples with p features, costing $O(npd \log n)$ due to sorting operations for split point identification. For M iterations, total complexity is $O(Mnpd \log n)$. In practice, with $M = 200$, $n \approx 1500$, $p = 50$, $d = 5$, this evaluates to approximately 10^8 operations, completing in 20–30 seconds on modern hardware.

A.2. XGBoost Regularized Objective

XGBoost optimizes a regularized objective at each iteration:

$$\mathcal{L}^{(m)} = \sum_{i=1}^{n} l(y_i, F_{m-1}(x_i) + f_m(x_i)) + \Omega(f_m) \quad (1)$$

where $\Omega(f_m) = \gamma T_m + \frac{1}{2}\lambda \sum_{j=1}^{T_m} w_j^2$ penalizes the number of leaves T_m and the magnitude of leaf weights w_j. Using second-order Taylor approximation:

$$\mathcal{L}^{(m)} \approx \sum_{i=1}^{n} \left[l(y_i, F_{m-1}(x_i)) + g_i f_m(x_i) + \frac{1}{2} h_i f_m(x_i)^2 \right] + \gamma T_m + \frac{1}{2}\lambda \sum_{j=1}^{T_m} w_j^2 \quad (2)$$

where $g_i = \frac{\partial l}{\partial F_{m-1}(x_i)}$ and $h_i = \frac{\partial^2 l}{\partial F_{m-1}(x_i)^2}$ are first and second-order gradients. The optimal weight for leaf j containing instances I_j is:

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \quad (3)$$

This closed-form solution enables efficient optimization and the regularization term λ prevents overfitting by penalizing large leaf weights.

A.3. LightGBM's Gradient-Based One-Side Sampling

LightGBM's GOSS algorithm prioritizes instances with large gradients while randomly sampling instances with small gradients:

Algorithm 2 Gradient-Based One-Side Sampling

Require: Instances I, gradients $\{g_i\}_{i \in I}$, sampling rates a, b
Ensure: Sampled instances I_s
1: Sort instances by $|g_i|$ in descending order
2: Let I_a = top $a \times |I|$ instances (large gradients)
3: Let I_r = randomly sample $b \times |I|$ from remaining instances
4: Compute amplification factor: $w = \frac{1-a}{b}$
5: For instances in I_r, multiply gradients by w
6: **return** $I_s = I_a \cup I_r$

With typical values $a = 0.2$, $b = 0.1$, GOSS uses only 30% of instances per iteration. Combined with histogram-based split finding (binning continuous features into discrete bins), LightGBM achieves $O(Mn'p' \log n)$ complexity where $n' = 0.3n$ and p' is the number of bins (typically 255), substantially faster than traditional gradient boosting while maintaining comparable accuracy.

References

Daron Acemoglu. Technical change, inequality, and the labor market. *Journal of Economic Literature*, 40(1): 7–72, 2002.

Daron Acemoglu and James A Robinson. *Economic Origins of Dictatorship and Democracy*. Cambridge University Press, 2005.

Facundo Alvaredo, Lucas Chancel, Thomas Piketty, Emmanuel Saez, and Gabriel Zucman. *World Inequality Report 2018*. Belknap Press, 2018.

Gary S Becker. *Human Capital: A Theoretical and Empirical Analysis*. University of Chicago Press, 1964.

Johannes Beutel, Sophia List, and Gregor von Schweinitz. Machine learning for financial risk management. *Annual Review of Financial Economics*, 11:1–23, 2019.

Leo Breiman. Random forests. *Machine Learning*, 45(1): 5–32, 2001.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

Richard B Freeman. Labor regulations, unions, and social protection in developing countries: Market distortions or efficient institutions? In *Handbook of Development Economics*, volume 5, pages 4657–4702. Elsevier, 2010.

Jerome H Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5): 1189–1232, 2001.

Oded Galor and Joseph Zeira. Income distribution and macroeconomics. *Review of Economic Studies*, 60(1): 35–52, 1993.

Elhanan Helpman. *Globalization and Inequality*. Harvard University Press, 2018.

Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301):790–794, 2016.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, pages 3146–3154, 2017.

Simon Kuznets. Economic growth and income inequality. *American Economic Review*, 45(1):1–28, 1955.

Thomas Piketty. *Capital in the Twenty-First Century*. Harvard University Press, 2014.

Adam Richardson, Thomas Mulder, and Tugrul Vehbi. Nowcasting gdp using machine-learning algorithms: A real-time assessment. *International Journal of Forecasting*, 37(2):941–948, 2021.

Joseph E Stiglitz. *The Price of Inequality: How Today's Divided Society Endangers Our Future*. W. W. Norton & Company, 2012.

Carolin Strobl, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, 8(1):25, 2007.