Japan GDP Growth Forecasting with Natural Disasters

Aldina Zeqiri

Abstract

This project builds a reproducible machine learning pipeline to forecast Japan's annual GDP growth using country-year aggregates of natural disasters. The task is predictive rather than causal: the objective is to evaluate whether disaster information observed up to year $t-1$ improves one-year-ahead forecasts of GDP growth in year t. GDP and GDP growth are taken from the World Bank World Development Indicators (WDI), while disaster outcomes (counts, deaths, and damages) are aggregated from EM-DAT. To prevent leakage, all non-GDP covariates are lagged by one year under a strict ex-ante rule. The pipeline compares transparent time-series baselines (mean and rolling averages) against standard supervised learning models (Ridge, Random Forest, Gradient Boosting, and MLP) using a chronological holdout window and time-series cross-validation. Results show that on a long-sample "main" specification, Gradient Boosting improves RMSE relative to strong persistence baselines, indicating incremental predictive content beyond simple rules. In restricted specifications that rely primarily on disaster aggregates, predictive performance deteriorates and R^2 can be near zero, consistent with disasters being noisy at annual aggregation. Adding an oil-related control improves restricted nowcast performance substantially, suggesting that oil shocks provide incremental signal. The report highlights limitations from small samples, structural breaks (e.g., COVID), and feature availability that shortens evaluation windows in some specifications.

1 Introduction

Natural disasters can disrupt production, infrastructure, and supply chains, and Japan provides salient examples such as the Great East Japan Earthquake. A large literature studies macroeconomic consequences of disasters and emphasizes heterogeneity across settings and event types [3, 4, 5]. From a forecasting perspective, an open question is whether disaster databases contain stable predictive patterns at the annual, country-level aggregation used in standard macro data.

This project frames the problem as a machine learning forecasting task. The target is Japan's annual GDP growth in year t, and the horizon is one year ahead. The key design constraint is that forecasts must be based only on information observable up to year $t-1$. This strict ex-ante framing is important because disaster impacts are often realized within-year and GDP outcomes may themselves affect measurement or reporting. A model that uses contemporaneous covariates would therefore risk overstating predictive performance.

The project's contribution is practical and methodological: it delivers a reproducible pipeline, compares multiple models against strong baselines under time-series validation, and documents failure cases explicitly. Rather than claiming that "ML works," the analysis explains when and why models add value, where they fail, and what those failures imply about the underlying signal in annual disaster aggregates.

2 Background

Empirical work on disasters and growth finds that average effects can be difficult to pin down and that impacts depend on exposure, preparedness, and economic structure [4, 5]. Some studies document persistent losses after major catastrophes, while others emphasize adaptation and reconstruction dynamics. For Japan specifically, disasters may operate through complex channels such as supply chain disruptions and sectoral linkages [7]. These mechanisms suggest that disaster information could matter for forecasting, but also that simple annual aggregates may be too coarse to capture the relevant propagation.

Forecasting with annual macro time series is also challenging because samples are small and structural breaks are common. Models with high flexibility can overfit, making transparent baselines and careful validation essential. This motivates a design that (i) prioritizes leakage-free feature construction, (ii) evaluates against rolling-mean baselines, and (iii) reports both holdout performance and time-series cross-validation stability.

3 Design & Architecture

3.1 Repository structure (reproducibility-first)

The repository follows a standard reproducible layout that separates data processing, feature construction, modeling, and artifacts:

- `src/data_loading.py` loads input spreadsheets from `data/` and standardizes columns.
- `src/features.py` constructs a yearly master table, creates lag/rolling features, and lags disaster aggregates.
- `src/models.py` trains models, runs time-series validation, and writes outputs to `results/`.
- `main.py` provides a single CLI entry point for graders.
- `tests/` contains unit/integration tests that enforce data integrity and pipeline stability.
- `dashboard/` contains optional scripts to generate HTML dashboards.

3.2 Pipeline dataflow and run modes (forecast vs nowcast)

Strict vs ex_post. In addition to forecast/nowcast, some diagnostic runs are labeled *ex_post* when they allow contemporaneous proxy information (e.g., shock indicators) that is not available under a strict ex-ante forecasting exercise. Therefore, *ex_post* results are reported as an *upper bound* and are not directly comparable to strict forecasting performance. The system is designed around a single master table indexed by year, with multiple run configurations that differ only in feature availability rules. The forecast mode enforces the strictest timing constraint: to predict gdp_growth_t, the model uses only lagged information up to $t-1$. The nowcast mode is included as a diagnostic benchmark: it allows within-year disaster aggregates and therefore represents an upper bound on what could be achieved if contemporaneous information were available.

Inputs (`data/`)
→ WDI GDP & GDP growth (Japan, annual)
→ EM-DAT events (Japan, event-level)
→ optional macro/oil controls (annual)

Build master table (`src/features.py`)
→ aggregate EM-DAT to country-year: counts, deaths, total damage, average magnitude
→ merge with WDI by year; align calendar years
→ create transformed variables: log(1+damage), damage share of GDP
→ fill disaster missingness with zeros (no recorded disaster)

Feature construction (`src/models.py`)
→ GDP dynamics: lags and rolling moments based on past values only
→ strict ex-ante: shift all disaster/macro/oil features by 1 year (forecast)
→ feature subsets: main vs restricted vs macro vs oil

Modeling & validation (`src/models.py`)
→ chronological holdout test window
→ time-series cross-validation (robustness)
→ compare baselines + ML models
→ save metrics + predictions to `results/`

Artifacts
→ `results/model_metrics_<run>.csv`
→ `results/predictions_<run>_<model>.csv`
→ optional HTML dashboards in `dashboard/`

Figure 1: End-to-end pipeline and dataflow used in the project.

3.3 Key design choices (explained in complete sentences)

All features are constructed under a strict ex-ante rule: predictors for year t must be observable by the end of year $t-1$. This rule is implemented by shifting disaster aggregates and controls by one year. The intent is to ensure that any measured performance corresponds to a feasible forecasting exercise rather than to a contemporaneous explanation of GDP outcomes.

The project uses multiple feature sets to separate genuine predictive content from mechanical improvements due to adding broad macro controls. The main specification includes lagged GDP dynamics and disaster aggregates, while restricted specifications remove GDP lags to test whether disasters alone carry signal. Additional variants include macro controls and an oil-related feature. This modular approach clarifies trade-offs because richer covariate sets can improve performance but also shorten the usable sample and reduce comparability across specifications.

The model set is chosen to balance flexibility and stability. Regularized linear models (Ridge) provide a strong baseline under multicollinearity and small samples [10]. Tree ensembles (Random Forest and Gradient Boosting) capture nonlinearities while remaining relatively robust [8, 9]. MLP is included as a high-capacity benchmark that can reveal overfitting. When flexible models fail, these failure cases are reported explicitly and used to motivate the final model choice.

3.4 Leakage prevention and validation protocol

Leakage prevention is treated as an architectural requirement, not a late-stage fix. The pipeline enforces three concrete safeguards. First, rolling features of GDP growth are computed using shifted

values so that the rolling window ends at $t-1$ rather than at t. Second, all non-GDP covariates are shifted by one year in strict forecast configurations. Third, preprocessing steps that depend on the distribution of features (imputation and scaling) are executed inside a scikit-learn `Pipeline` so they are fit on training folds only and then applied to validation/test folds.

Validation follows time-series logic. A chronological holdout window measures final out-of-sample performance. Time-series cross-validation complements this by repeating the train/validation split across multiple historical folds. Early folds have short training windows, which increases variance and is one reason why conservative hyperparameters are preferred over aggressive tuning.

3.5 Feature-set summary (what changes across runs)

Run configuration	Included feature groups (all timing rules enforced by the run)
Main	Lagged GDP dynamics (lags + rolling mean/std) + lagged disaster aggregates (counts, deaths, damages, magnitude)
Restricted	Disaster aggregates only (no GDP lags), to test whether disasters alone predict growth
Restricted_macro	Restricted + lagged macro controls (when available), to check incremental signal from macro series
Restricted_oil	Restricted + lagged oil-related feature, to capture external macro shocks relevant for Japan

Table 1: Conceptual summary of feature sets used to isolate predictive signal.

3.6 Artifacts and outputs

Each run produces machine-readable outputs that support both grading and interpretation:

- `results/model_metrics_<run>.csv` contains train/test RMSE, MAE, R^2, and subgroup metrics.
- `results/predictions_<run>_<model>.csv` contains year-by-year predictions for plots and diagnostics.
- Optional HTML dashboards are saved in `dashboard/`.

4 Implementation

4.1 Data sources and preprocessing

GDP and GDP growth are taken from the World Bank's World Development Indicators (WDI) [1]. Disaster aggregates are constructed from EM-DAT [2] at the country-year level. Inputs are stored as spreadsheets in `data/`, and the pipeline transforms them into a single yearly master table. The preprocessing steps align GDP years, compute or import growth, aggregate disaster outcomes (counts, deaths, damages), construct GDP lag and rolling features, and then shift all disaster/control features by one year to satisfy the strict ex-ante rule.

Two implementation details matter for correctness. First, disaster missingness is not treated as "unknown" but as "no recorded disaster," and the pipeline fills disaster aggregate columns with zeros after the merge. Second, transformations are chosen to reduce skewness and prevent extreme

values from dominating: the pipeline uses log(1+damage) and damage share of GDP rather than raw damages, because EM-DAT damages can be heavy-tailed and sometimes missing.

4.2 Feature engineering (how each feature is computed)

GDP dynamics are represented with lagged values and rolling statistics. Concretely, the dataset includes gdp_growth_{t-1} and gdp_growth_{t-2}, and rolling mean/std features computed from past values only. The rolling features are implemented using shifted series so that the rolling window ends at $t-1$, which prevents information from year t entering predictors for year t.

Disaster aggregates are computed at the country-year level and then lagged under strict forecasting. The lagging step is explicit: the columns for event count, deaths, damages, and magnitude are shifted by one year, and the contemporaneous columns are only used in nowcast runs. This makes the timing rule auditable in code and keeps the meaning of each configuration clear.

4.3 Missing values, scaling, and leakage-proof preprocessing

Feature availability differs across specifications because some macro series are not available for the full historical window. Rather than dropping large parts of the data manually, the pipeline keeps the merged table and uses a consistent strategy: the target and essential GDP lag features define the usable rows, while remaining missing values in predictors are handled within the model pipeline.

For models that require numeric stability (Ridge and MLP), preprocessing is implemented as `SimpleImputer` followed by `StandardScaler` inside a scikit-learn `Pipeline`. This is important because it ensures imputation and scaling are fit on training data only in each fold. Tree models do not require scaling, but they still receive imputed features for consistency.

4.4 Model training and validation

The training code compares baselines and ML models using a chronological holdout test window. Time-series cross-validation is implemented as a robustness check to verify that conclusions do not depend on a single split. The implementation saves both summary metrics and year-by-year predictions to support reproducible plots and diagnostic analysis.

Baselines are treated as first-class models because annual macro series are often hard to beat with complex estimators. The mean baseline and rolling baselines serve as realistic reference points: any ML model that cannot outperform them is interpreted as lacking incremental predictive value for the given feature set.

4.5 Hyperparameter policy

Hyperparameters are kept conservative because annual time series provide a limited number of observations. Aggressive tuning can overfit the validation procedure itself, especially when early cross-validation folds contain very short training windows. The report therefore emphasizes transparent comparisons across a small set of standard models with reasonable defaults, and prioritizes out-of-sample performance and stability over marginal gains from extensive search.

4.6 Testing and contracts (what the tests enforce)

The test suite is designed to catch silent errors that would invalidate forecasting claims. Unit tests check that the master table has the required columns and that key transformations behave as intended. Integration tests run a small end-to-end pipeline to ensure that outputs are generated deterministically and that the core artifacts (metrics and prediction files) are produced. This matters because time-series forecasting projects can fail in subtle ways, such as accidentally shuffling years, using contemporaneous covariates, or changing feature definitions without updating downstream scripts.

4.7 Implementation issues encountered

Several practical issues affected the empirical results and are important for interpretation. Adding certain controls (macro variables or oil-related features) reduced the usable period because some series are not available for the full historical window. This mechanically shortens the holdout test window and makes performance across specifications less directly comparable.

Model stability also differs across feature sets. Flexible estimators such as MLP can fit the training data almost perfectly on short samples but generalize poorly, which is consistent with high variance and overfitting. This reinforces the role of regularization and strong baselines in small-sample time-series forecasting.

An additional unexpected pattern appears in some restricted runs where Gradient Boosting matches the mean baseline almost exactly. This suggests that under the restricted feature set, either the effective predictors contain too little variation, missingness reduces useful splits, or conservative settings lead to no meaningful tree improvements. Rather than hiding this behavior, the report treats it as a diagnostic result: model choice cannot compensate for a feature set that provides little predictive signal.

5 Evaluation

5.1 Headline results with baselines

Table 2 reports two representative scenarios and compares the best non-baseline model against a strong baseline. Numbers are taken from the saved `model_metrics` outputs.

Setting	Best model	RMSE	R^2	MAE	Best baseline	Baseline RMSE
Forecast (strict, main)	Gradient Boosting	1.503	0.128	1.086	Roll-3 mean	2.016
Nowcast (ex_post, restricted_oil)	Ridge	1.592	0.401	1.416	Mean (train)	2.102

Table 2: Headline test-set results with baseline comparison.

On the main long-sample specification, Gradient Boosting improves RMSE by roughly 25% relative to a rolling-mean baseline, indicating incremental predictive content beyond persistence. On the restricted_oil specification, Ridge achieves a substantially higher R^2 on a shorter evaluation window, consistent with oil-related information capturing relevant macro shocks.

5.2 Restricted nowcast results (with and without oil)

Table 3 highlights how adding the oil feature changes generalization performance in the restricted setting. The results show that disasters-only restricted features provide limited predictive power, while the oil augmentation yields a sizable improvement and better fit in severe-disaster years as well.

Setting	Model	RMSE test	R^2 test	RMSE severe	R^2 severe
Nowcast (ex_post, restricted)	Ridge	2.012	0.043	1.892	0.250
Nowcast (ex_post, restricted_oil)	Ridge	1.592	0.401	1.734	0.371

Table 3: Restricted nowcast performance for Ridge with and without oil features.

5.3 Prediction diagnostics and model behavior (strict forecast main)

The strict forecast prediction file indicates that a small number of extreme years dominate the test error. In particular, the COVID recession year (2020) is under-predicted in magnitude, which is consistent with the difficulty of forecasting rare structural breaks using only lagged aggregates.

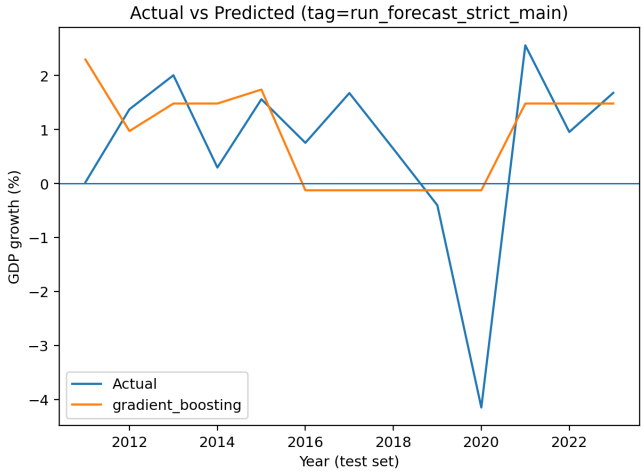

Figure 2: Strict forecast (main): Predicted vs actual GDP growth for Gradient Boosting.
Commentary. The model tracks moderate fluctuations but underestimates the magnitude of rare breaks, which is typical when training on a small annual sample with few crisis observations.

5.4 Exploratory data analysis (context for the forecasting difficulty)

EDA figures provide context for why the task is hard. GDP growth has a small number of extreme years, and disaster damage variables are heavy-tailed. These properties make models sensitive to a few observations, amplify overfitting risk, and motivate conservative model selection and strong baselines.

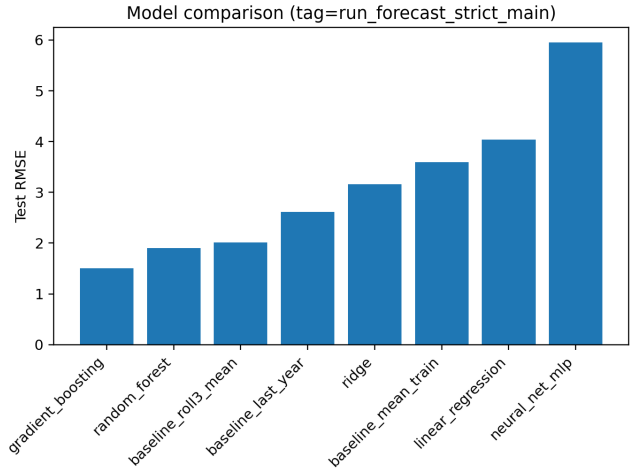

Figure 3: Model comparison (strict forecast, main): test RMSE across baselines and ML models.
Commentary. Gradient Boosting outperforms strong persistence baselines, indicating that the feature set provides additional predictive content beyond simple time-series rules.

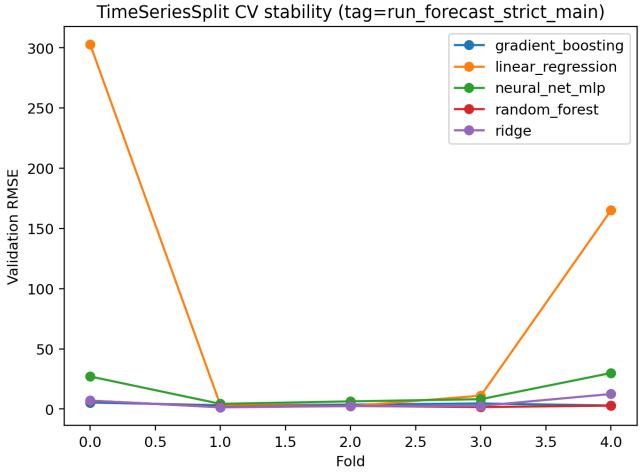

Figure 4: Time-series cross-validation stability (strict forecast, main).
Commentary. Early folds have short training windows, which makes validation error volatile; the plot helps assess whether the selected model remains competitive across folds rather than only in one split.

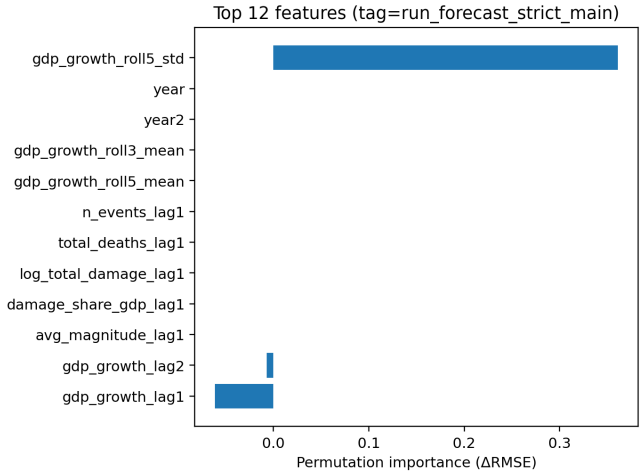

Figure 5: Feature importance for the selected model (strict forecast, main).
Commentary. Lagged GDP dynamics dominate importance, while disaster aggregates contribute *minimal* marginal signal at annual frequency, which is consistent with disasters being noisy when aggregated to country-year level.

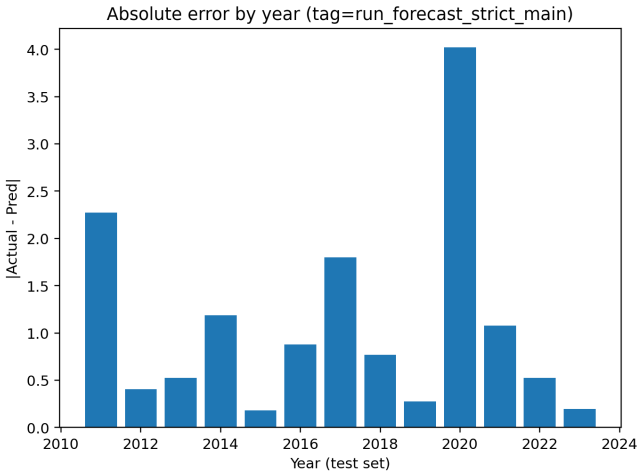

Figure 6: Errors by year (strict forecast, main): large-error years highlight structural breaks and rare shocks.
Commentary. The largest errors concentrate in a few extreme years, supporting the interpretation that structural breaks drive most out-of-sample loss.

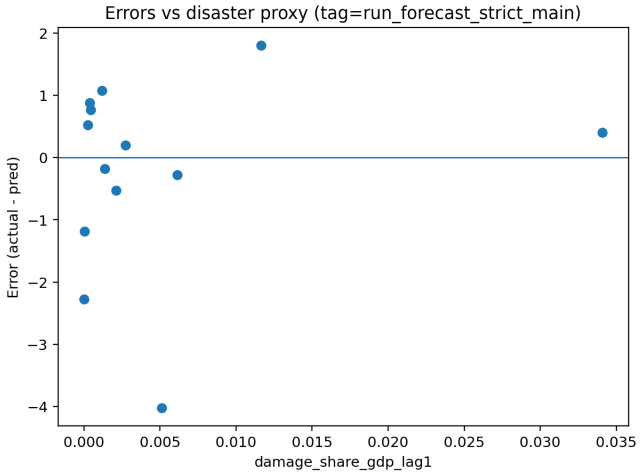

Figure 7: Errors vs disaster intensity (strict forecast, main).

Commentary. A weak relationship between error magnitude and disaster intensity is consistent with the finding that annual disaster aggregates provide limited incremental predictive content once GDP persistence is accounted for.

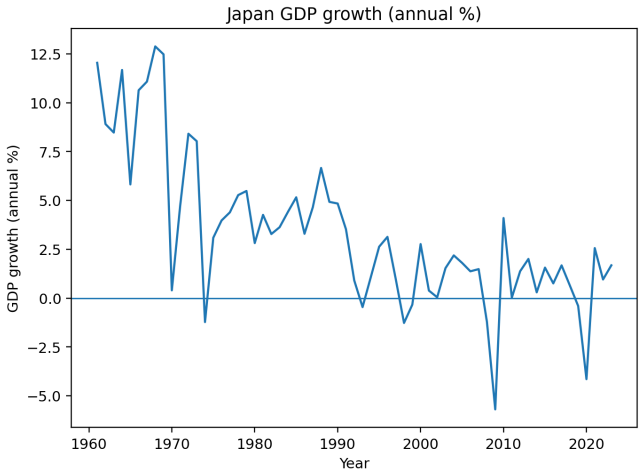

Figure 8: EDA: Japan annual GDP growth (distribution and time pattern).

Commentary. The series is dominated by a few extreme downturns; forecasting accuracy is therefore driven disproportionately by rare events.

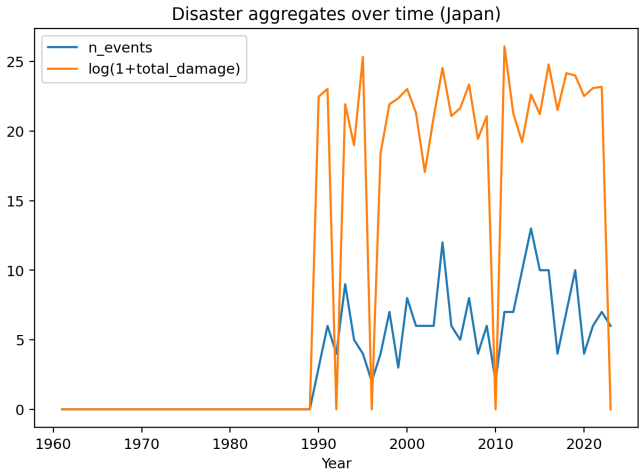

Figure 9: EDA: Disaster counts and outcomes over time (annual aggregates).
Commentary. Disaster aggregates show intermittent spikes, but annual aggregation can blur timing and mechanisms, limiting predictive usefulness.

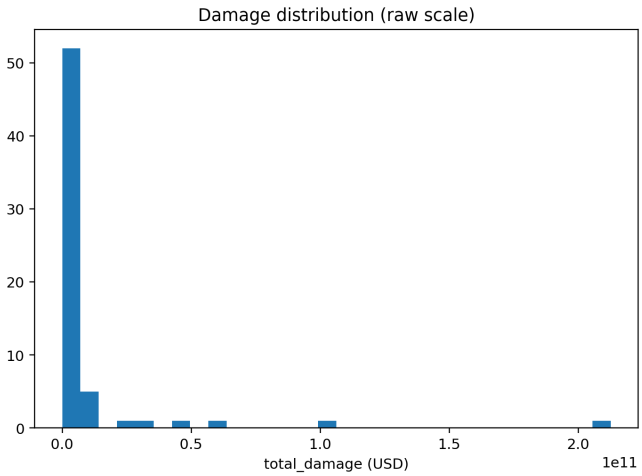

Figure 10: EDA: Distribution of reported damages (heavy tails).
Commentary. Heavy-tailed damages justify log transforms; otherwise a few events can dominate the learning signal and the loss.

6 Discussion

6.1 Interpretation

The results suggest that disasters-only country-year aggregates provide limited predictive signal for Japan's annual GDP growth at a one-year horizon. This is consistent with the broader literature, which emphasizes heterogeneous impacts and the difficulty of identifying stable aggregate patterns from disaster databases [4, 5]. The restricted specification often produces near-zero or negative R^2, meaning that predictions are not reliably better than a mean baseline.

The oil-augmented restricted model performs substantially better, suggesting that oil-related variation captures macro shocks that matter for Japan's growth and are not fully redundant with lagged GDP features. This aligns with the general idea that aggregate GDP fluctuations are frequently driven by broad macro forces rather than localized events.

6.2 Unexpected findings and interpretation

Two unexpected patterns are particularly informative. In restricted specifications, Gradient Boosting sometimes behaves like a constant predictor that matches the mean baseline. This can occur when the remaining features provide limited variance, contain missing values that reduce effective training information, or fail to create splits that improve out-of-sample loss. The practical implication is that a powerful model does not automatically generate predictive power when the feature set is not informative at the annual level.

A second pattern is the clear overfitting of the MLP model in some runs. Extremely low training error combined with substantially worse test error indicates that the model is learning noise rather than stable structure. This supports the report's emphasis on regularization, baselines, and cross-validation stability when selecting the final model.

6.3 Limitations

The primary limitation is sample size: annual data provide relatively few observations, and some specifications further shorten the usable window due to feature availability. Structural breaks (such as the COVID recession) are rare but dominate error, and no annual feature set can fully encode these breaks without contemporaneous information. Disaster aggregates are also coarse proxies for economic exposure; they omit sectoral composition, geographic concentration, and policy response, all of which matter for macro outcomes.

7 Conclusion

This project delivers a reproducible ML pipeline to forecast Japan's annual GDP growth using lagged disaster aggregates and optional controls. The results show that in a long-sample setting with GDP persistence features, Gradient Boosting improves meaningfully over rolling baselines, indicating incremental predictive content. In contrast, disaster-only aggregates offer limited forecasting power at annual frequency, and flexible models can overfit under small-sample constraints. The strongest restricted performance occurs when adding an oil-related feature, suggesting that broad macro shocks are more predictive for Japan's annual growth than disaster aggregates alone. Future work would

benefit from higher-frequency data, richer exposure measures, and explicit modeling of structural breaks.

References

[1] World Bank. *World Development Indicators (WDI).*

[2] CRED / UCLouvain. *EM-DAT: The International Disaster Database.* Dataset used for disaster aggregates.

[3] Ilan Noy. "The macroeconomic consequences of disasters." *Journal of Development Economics*, 88(2), 221–231, 2009.

[4] Eduardo Cavallo and Ilan Noy. "Natural Disasters and the Economy — A Survey." *International Review of Environmental and Resource Economics*, 2011.

[5] Gabriel Felbermayr and Jasmin Gröschl. "Naturally negative: The growth effects of natural disasters." *Journal of Development Economics*, 111, 92–106, 2014.

[6] Solomon M. Hsiang and Amir S. Jina. "The Causal Effect of Environmental Catastrophe on Long-Run Economic Growth: Evidence From 6,700 Cyclones." *NBER Working Paper* 20352, 2014.

[7] Vasco M. Carvalho, Makoto Nirei, Yukiko U. Saito, and Alireza Tahbaz-Salehi. "Supply Chain Disruptions: Evidence from the Great East Japan Earthquake." *Quarterly Journal of Economics*, 136(2), 1255–1321, 2021.

[8] Leo Breiman. "Random Forests." *Machine Learning*, 45, 5–32, 2001.

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

[10] Arthur E. Hoerl and Robert W. Kennard. "Ridge Regression: Biased Estimation for Nonorthogonal Problems." *Technometrics*, 12(1), 55–67, 1970.

[11] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, 12, 2825–2830, 2011.

Appendix (supplementary material)

The remainder of this document provides supplementary material (implementation details, additional figures, and reproducibility notes).

A Reproducibility (optional)

Minimum run (safe for graders).

```
python main.py --only-main
```

Run all configurations.

```
python main.py
```

Build dashboard (optional).

```
python dashboard/build_dashboard.py
```

Key outputs.

- `results/model_metrics_*.csv`: summary metrics per run.
- `figures/predictions_*.csv`: year-by-year predictions.
- `dashboard/*.html`: interactive dashboards.