ML & Climate | Final Paper

Melina Garza [mjg2290] and Dhwani Sreenivas [ds4022]

May 11, 2025

1 Introduction

Natural disasters, particularly floods, are becoming increasingly frequent and severe (1), posing significant risks to infrastructure, communities, and economic systems. Flooding not only disrupts daily life but also impacts local real estate markets, as property values often decrease in affected areas. Understanding and quantifying the economic consequences of floods is critical for stakeholders such as governments, property investors, and homeowners, who need to assess risk and plan for resilience. However, reliable forecasting of such events and understanding of their long-term effects remain a challenging problem due to the complexity of environmental data and the difficulty in predicting future anomalous weather events such as major floods. Our project explores whether ML models can help quantify and forecast the economic impact of environmental risks, focusing on flood events.

2 Problem Formulation

2.1 Technical Tasks

This project addresses two interconnected challenges:

Predicting heavy rainfall events as a proxy for flood risk: We investigated whether ML models can predict heavy rainfall events, which serve as a proxy for flood risk. Using decision tree-based models such as XGBoost and Random Forest, we classified rainfall events and predicted their intensity.
Quantifying the impact of flood events on regional housing markets: We examine how flood events, in terms of their severity and displacement effects, correlate with changes in Housing Price Indices (HPI) at the ZIP code level. We apply a variety of regression models, including Random Forest, CatBoost, and TabPFN, to predict the impact of floods on housing prices.

Additionally, due to the limited availability of data, we use a synthetic data generation technique, Gaussian Mixture Model (GMM), to improve the models' performance and generalization.

2.2 Key Assumptions

We assume that heavy rainfall serves as a suitable proxy for flood events, as high-resolution precipitation data is more widely available and consistently recorded at the ZIP code level compared to the sparse and inconsistent nature of historical flood records. We also assume that housing markets are influenced by nearby environmental disasters, and that historical patterns in precipitation and flooding provide meaningful information for forecasting future impacts.

2.3 Constraints

This project is limited by the availability of high-quality, granular flood impact data across both space and time. Modeling heavy rain events also poses challenges due to severe class imbalance, as such events are relatively rare. Furthermore, the dataset used to study the economic impact of flooding is small, which necessitated the use of synthetic data generation to improve model stability and generalization.

3 Data

Weather & Precipitation Data: We used hourly precipitation data from 2007 to 2014 for Massachusetts weather stations, sourced from the National Centers for Environmental Information (NCEI) – NOAA (2). In addition, we incorporated a 2010 hourly normals dataset (e.g. temperature, wind speed, dew point) for two Massachusetts stations. Our initial modeling focused solely on precipitation data for classifying heavy rainfall events, but we later combined it with the normals dataset to improve prediction accuracy.

Flood Event Data: We used a global flood archive that spans 1985 to 2021(3), which includes detailed attributes such as severity, duration, affected area, and estimated casualties and displacement. For our analysis, we filter this dataset to include only U.S. flood events and aggregate the data by ZIP code and year to align with the structure of our housing dataset.

Housing Market Data: Housing Price Index (HPI) data from the Federal Housing Finance Agency, organized by ZIP code and year (4). We used this dataset to quantify the economic impact of floods on local housing values.

Synthetic Data: To address data scarcity in the HPI modeling task, we generated synthetic samples using Gaussian Mixture Models trained on real flood-event features. This expanded dataset is used to improve model robustness and generalization.

4 Models & Implementation

Each of the two main tasks—precipitation prediction and assessing the housing market impact of flooding—required tailored ML pipelines. This section details the modeling rationale, preprocessing steps, and implementation strategies.

4.1 Task 1: Precipitation Prediction

Implemented in **ml_climate_precipitation_prediction.ipynb**
Datasets:

- **precipitation_data.csv**: Hourly precipitation from 2007–2014 at the station level.
- **2010_weather_MA.csv**: Hourly weather normals (temperature, wind, dew point, etc.) for two Massachusetts stations.
- **combined_weather_precipitation.csv**: Merged dataset joining precipitation and weather data

Preprocessing: Initial data cleaning on **precipitation_data.csv** included removing invalid/missing precipitation values (HPCP), generating time-based features (hour, day of week, month), and adding lag HPCP variables. **The target variable was HPCP, which represents hourly precipitation in hundredths of inches** (5).
For the combined dataset, we joined the weather and precipitation files on `DATE` and `STATION`, then imputed missing weather features using KNN (k-Nearest Neighbors) imputation (6).

Initial Dataset: Model Implementations

- **XGBoost Classifier:** Selected for its strong performance on structured tabular data and its robustness to class imbalance through built-in weighting (7). Tested with and without the elevation feature, using either SMOTE or class weighting (scale_pos_weight) to handle imbalance.
- **Random Forest Classifier:** Chosen for its ability to model non-linear interactions and its resilience to noise, making it well-suited for environmental data. All runs used **Borderli-**

neSMOTE (8); configurations varied in use of elevation and whether hyper-parameters were optimized via **GridSearchCV**.

- **Random Forest Regressor:** Used to predict continuous precipitation amounts. Hyperparameters were tuned using RandomizedSearchCV, and models were run both with and without elevation.
- **Neural Network**: A simple feedforward architecture with two hidden layers (16 and 8 neurons, ReLU activation), trained with the Adam optimizer and cross-entropy loss over 200 epochs. Class imbalance was mitigated using SMOTE oversampling.

Combined Dataset: Model Implementations

- **XGBoost Classifier**: Tested with and without elevation as a feature. Class imbalance was addressed using class weighting (scale_pos_weight).
- **Random Forest Classifier**: The same model configuration was used as the initial dataset and tuned hyperparameters (from GridSearchCV) were reused.
- **Random Forest Regressor**: The same model configuration was used as the initial dataset and tuned hyperparameters (from RandomizedSearchCV) were reused.

4.2 Task 2: Housing Price Impact of Flooding

Implemented in **natural_disaster_housing_prediction.ipynb**
Datasets:

- **united_states_floods.csv**: U.S. flood data (1985–2021), aggregated by ZIP code and year
- **united_states_housing.csv**: Housing Price Index (HPI) by ZIP code and year (1975-2024)
- **merged_housing_flood_data.csv**: Merged flood & housing dataset.

Preprocessing: ZIP codes were inferred from flood event coordinates. Flood features (e.g., flood_count, max_severity, median_displaced, median_area, etc) were aggregated annually by ZIP code. To capture regional trends, we computed 10-year rolling averages for ZIP code prefixes (first 3 digits), creating features such as max_severity_avg_10, median_dead_avg_10, median_duration_avg_10, etc. **The target variable was HPI(Housing Price Index).**

Synthesized Data: Due to the small sample size ($\approx$ 200 rows), synthetic data was generated using Gaussian Mixture Models (GMMs). Numeric flood features were used to fit a GMM with 3 components, which then generated $\approx$ 1000 new samples. These were aligned with real data using distance-based (**pairwise_distances_argmin_min**) matching to preserve realistic feature relationships. **All models were trained and evaluated across four dataset variants: real and synthetic versions of both the raw and 10-year rolling aggregated flood data.**

Models and Configurations

- **Random Forest Regressor**: Used as a strong baseline due to its ability to handle non-linear dependencies and its resilience to overfitting on smaller datasets (9). Trained with both default and RandomizedSearchCV-optimized parameters.
- **TabPFN Regressor**: Chosen because it achieves strong performance on small tabular datasets without requiring task-specific training (10). Used its default configuration.
- **CatBoost Regressor**: Selected for its performance on small tabular data and native support for categorical features (11). Default configuration with early stopping (100 rounds) and RMSE as the loss function.

Analyzing the best models: Feature importance and residuals were analyzed for the top-performing models, specifically CatBoost and TabPFN, to assess how different flood-related variables influenced housing prices.

5 Experiments and Results

5.1 Precipitation Prediction

For all models, 0.2 inches is classified as moderate rain and 0.3 inches is classified as heavy rain.

5.1.1 XGBoost Classifier

Experimental Setup: All initial models used an 80/20 train-test split. For the combined data models, we used a 70/30 train-test split.

With Initial Data:

Model Configuration	Elevation	Precision (1)	Recall (1)	F1-Score (1)
No Balancing	No	0.31	0.08	0.13
No Balancing	Yes	0.26	0.08	0.12
SMOTE	**No**	**0.24**	**0.26**	**0.25**
SMOTE	Yes	0.14	0.34	0.20
Class Weights	No	0.11	0.39	0.17
Class Weights	Yes	0.11	0.41	0.17

Table 1: XGBoost Classifier Performance for Heavy Rain Prediction (Positive Class = 1)

With Combined Data:

Model Configuration	Precision (1)	Recall (1)	F1-score (1)	Accuracy
Without Elevation	0.31	0.31	0.31	0.98
With Elevation	**0.44**	**0.31**	**0.36**	**0.98**

Table 2: XGBoost Classifier Results with Combined Data (Threshold ≥ 0.3)

Interpretation: In the initial models, SMOTE without elevation produced the highest F1, suggesting that it improves the model's sensitivity towards heavy rain events. When using combined data, our overall F1 scores for the minority class improved, and adding elevation further improved the F1 score to 0.36. This indicates that elevation adds predictive value for heavy rain. However, overall XGBoost still performed poorly with even the best f1-score being quite low.

5.1.2 Random Forest Classifier

Experimental Setup: All initial models used an 80/20 train-test split. For the combined data models, we used a 70/30 train-test split.

With Initial Data:

Setup	Precision (Class 1)	Recall (Class 1)	F1-score (Class 1)	Accuracy
No elevation, default RF	0.10	0.06	0.08	0.98
With elevation, default RF	0.10	0.06	0.08	0.98
No elevation, tuned RF	**0.23**	**0.23**	**0.23**	**0.97**
With elevation, tuned RF	**0.19**	**0.30**	**0.23**	**0.97**

Table 3: Performance of Random Forest models under different feature and tuning configurations.

With Combined Data:

Model Configuration	Precision (1)	Recall (1)	F1-score (1)	Accuracy
No Elevation, Threshold ≥ 0.2	0.58	0.37	0.45	0.97
No Elevation, Threshold ≥ 0.3	0.33	0.08	0.12	0.98
With Elevation, Threshold ≥ 0.2	**0.55**	**0.40**	**0.46**	**0.97**
With Elevation, Threshold ≥ 0.3	0.33	0.08	0.12	0.98

Table 4: Random Forest Classifier Results with Combined Data

Interpretation: Of our initial models, the hyperparameter-tuned ones performed the best, with a dramatic increase in F1 scores. Our best performing random forest classifier was trained on combined data with elevation with a threshold of ≥ 0.2. Lowering the heavy rain threshold to ≥ 0.2 boosts recall, which is expected. Overall, the models still perform poorly, and the high accuracy is only reflective of the majority class.

5.1.3 Random Forest Regressor

Experimental Setup: All initial models used an 80/20 train-test split. For the combined data models, we used a 70/30 train-test split.

With Initial Data:

Model Configuration	Mean Squared Error (MSE)	R^2 Score
No Elevation	**0.00519**	**0.2994**
With Elevation	0.00521	0.2968

Table 5: Performance of Random Forest Regressor for Precipitation Prediction

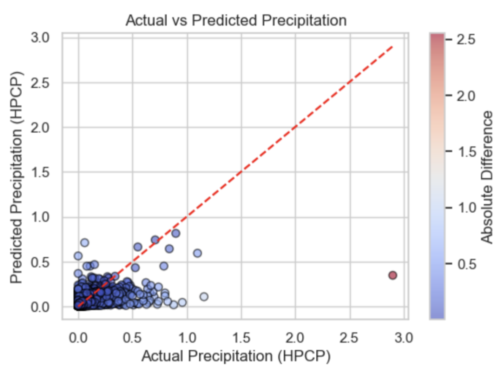

Figure 1: Random Forest Regressor for Precipitation Without Elevation

With Combined Data:

Model Configuration	CV Mean MSE	Test MSE	R^2 Score
No Elevation	**0.00482**	**0.00677**	**0.252**
With Elevation	0.00498	0.00680	0.248

Table 6: Random Forest Regressor Results with Combined Data

Interpretation: Both the initial models perform similarly, with the model without elevation having the best R^2 value. From the combined data model results, we can see that elevation might actually worsen performance. The low overall R^2 values (around 0.3) suggest pretty poor predictive power.

5.1.4 Neural Network

Experimental Setup: We implemented a feedforward neural network trained on only the initial dataset. We used an 80/20 train-test split.

With Initial Data:

Class	Precision	Recall	F1-score	Support
No Rain / Light Rain	0.98	0.54	0.69	7700
Heavy Rain	0.03	0.61	0.06	175
Accuracy			0.54	7875
Macro Avg	0.51	0.57	0.37	7875
Weighted Avg	0.96	0.54	0.68	7875

Table 7: Neural Network Classifier Results for Heavy Rain Prediction (using Initial Data)

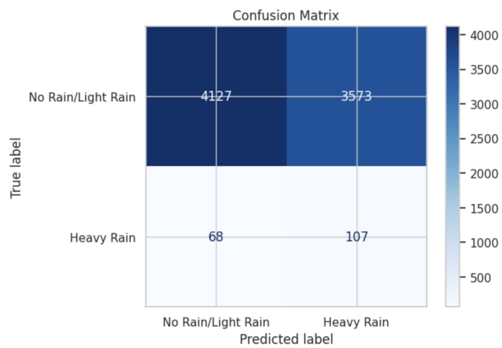

Figure 2: Neural Network Confusion Matrix

Interpretation: The neural network yielded the weakest performance among all models, with an F1-score of just 0.06. Despite being trained on the larger initial precipitation dataset, it struggled to accurately predict heavy rainfall events. Given its poor results, we excluded it from further iterations.

5.2 Housing Price Impact of Flooding

5.2.1 Random Forest Regressor

Experimental Setup: All models used a 70/30 train-test split.

Results:

Model Configuration	R^2 Score	MAE	Relative Error	Baseline MAE	Baseline R^2
Current Year Only, Real Data	0.058	124.70	47.98%	132.51	-0.02
With 10 Year Prior Averages, Real Data	0.120	131.84	49.42%	139.57	-0.03
Current Year Only, Synthetic Data	**0.528**	**89.13**	**39.39%**	111.92	-0.00
With 10 Year Prior Averages, Synthetic Data	0.131	116.86	49.33%	122.00	-0.00

Table 8: HPI Random Forest Regressor Performance Across Model Configurations

Interpretation: Our best performing model did not use historical data but used synthetic data. It is the highest R^2 score by a large margin, suggesting historical averages are not that important. All models beat the baseline, and our strongest model shows a promising moderate predictive power.

5.2.2 TabPFN Regressor

Experimental Setup: All models used a 50/50 train-test split.

Results:

Model Configuration	R^2 Score	MAE	Relative Error	Baseline MAE	Baseline R^2
Current Year Only, Real Data	0.0025	114.97	46.99%	113.08	-0.01
With 10 Year Prior Averages, Real Data	0.0598	127.14	51.44%	116.14	-0.02
Current Year Only, Synthetic Data	**0.4246**	**82.01**	**35.98%**	109.53	-0.00
With 10 Year Prior Averages, Synthetic Data	0.1793	114.33	50.31%	129.05	-0.00

Table 9: TabPFNRegressor Performance Across Model Configurations

Interpretation: As with the random forest regressor, TabPFN performs the best with the use of synthetic and current year data. The most effective predictor of HPI shifts appears to be synthetic flooding data. Overall, TabPFN performed worse than the random forest regressor.

5.2.3 CatBoost Regressor

Experimental Setup: All models used a 70/30 train-test split.

Results:

Model Configuration	R^2 Score	MAE	Relative Error	Baseline MAE	Baseline R^2
Current Year Only, Real Data	0.004	123.69	47.59%	132.51	-0.02
With 10 Year Prior Averages, Real Data	0.159	131.56	49.31%	139.57	-0.03
Current Year Only, Synthetic Data	**0.556**	**81.28**	**36.30%**	106.62	-0.00
With 10 Year Prior Averages, Synthetic Data	0.258	111.15	48.75%	127.10	-0.00

Table 10: CatBoost Regressor Performance Across Model Configurations

Interpretation: The CatBoost regressor performed the best when predicting HPI, with the current year only, synthetic data model having an R^2 of 0.556. It outperforms the other models in both R^2 score and MAE, and shows a similar pattern of synthetic current data being the best combination. Its results are promising for future work.

6 Conclusion

For heavy rainfall prediction, the XGBoost classifier trained on combined data with elevation achieved the best F1 score of 0.36, however we did get better results with the Random Forest Classifier when lowering the threshold for heavy rain to 0.2. This achieved an F1 score of 0.46, but with the normal heavy rain threshold of 0.3 it performed worse than XGBoost. Overall, for normal heavy rain classification, XGBoost performed the best.

For housing price index(HPI) prediction, CatBoost was our best regressor. Overall performance across all models showed that synthetic data significantly improved performance, and CatBoost outperformed all others with an R^2 value of 0.556. Another key insight gained was that historical climate averages did not improve predictive power, which was somewhat unintuitive.

Future work could focus on expanding the temporal and spatial coverage of the datasets to improve generalization, and exploring additional models (e.g., LSTMs for time series or ensemble meta-learners) to better capture complex patterns and long-term dependencies.

References

[1] Climate Mapping for Resilience and Adaptation (CMRA), “About this map — cmra.” Climate Mapping for Resilience and Adaptation, n.d. Accessed: 2025-05-11.

[2] National Centers for Environmental Information (NCEI), “Climate data online (cdo).” https://www.ncei.noaa.gov/cdo-web/. Accessed: 2025-05-11.

[3] G. Brakenridge, “Global active archive of large flood events, 1985–present.” http://floodobservatory.colorado.edu/Archives/, n.d. Accessed: 2025-05-11.

[4] Federal Housing Finance Agency (FHFA), “Annual house price indexes (zip5).” https://www.fhfa.gov/hpi/download/annually/hpi_at_bdl_zip5.xlsx. Accessed: 2025-05-11.

[5] National Centers for Environmental Information (NCEI), “Hourly precipitation data (precip_hly) documentation.” https://www.ncei.noaa.gov/pub/data/cdo/documentation/PRECIP_HLY_documentation.pdf, n.d. Accessed: 2025-05-11.

[6] scikit-learn developers, “sklearn.impute.knnimputer — scikit-learn documentation.” scikit-learn, n.d. Accessed: 2025-05-11.

[7] RITHP, “Xgboost and imbalanced datasets: Strategies for handling class imbalance.” Medium, 2022. Accessed: 2025-05-11.

[8] imbalanced-learn developers, “imblearn.over_sampling.borderlinesmote — imbalanced-learn documentation.” https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.BorderlineSMOTE.html, n.d. Accessed: 2025-05-11.

[9] GeeksforGeeks, “Random forest regression in python.” GeeksforGeeks, 2025. Accessed: 2025-05-11.

[10] Prior-Labs, “Tabpfn-v2-reg.” https://huggingface.co/Prior-Labs/TabPFN-v2-reg, n.d. Accessed: 2025-05-11.

[11] GeeksforGeeks, “Regression using catboost.” GeeksforGeeks, 2025. Accessed: 2025-05-11.