Project Report

Introduction

This project analyzes historical data from 1900 through 2024. We aim to create a general machine learning algorithm to predict the mortality rate across a series of natural disasters, including earthquakes, floods, storms, droughts, volcanic activity, air, and wildfires. The features implemented into the learning of each prediction are latitude, longitude, start year, and start month.

Machine Learning Algorithm

I. Extreme Gradient Boosting & Motivation

Extreme gradient boosting was utilized to predict the mortality rate of each event, as it is the most suitable for the dataset's small size and the non-linear behavior of natural disasters. This algorithm generates multiple decision trees that build accuracy based on one another's residuals (errors). Particularly, an objective function is utilized and built by each subsequent tree; it consists of the addition of two functions: loss and regularization. The loss formula in this model is mean squared error; it is used to minimize the difference between future predictions and the actual values in the data. The regularization formula creates a penalty for the addition of leaves to each tree; it works to minimize "complexity" of the algorithm, preventing overfitting.

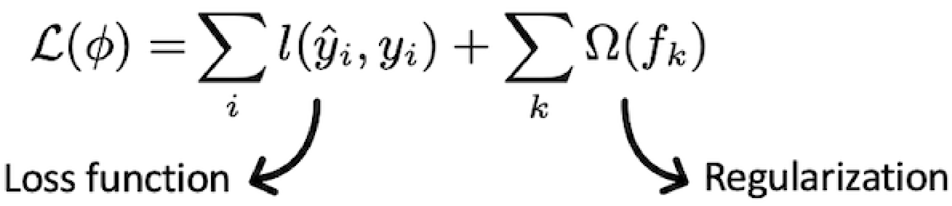
$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$
Loss function
Regularization

Figure 1. Customized Objective Function. DZone, 2025.

The "cleaning" process of the dataset consisted of filling missing latitude and longitude values according to the country where each event took place. The original data from the API contained several different features; for relevancy and consistency of this investigation, only "Start Year", "Start Month", "Latitude", "Longitude", and "Total Deaths" were kept. In addition to filtering for relevant numerical values, these key attributes were refined by extracting empty, infinite, and invalid fields such as mismatched inputs. The dataset utilized in this project was significantly reduced after cleaning, making predictions prone to overfitting, which affected the accuracy of predictions. Under this context, the chosen algorithm offers the advantage of prioritizing the minimization of overfitting that would otherwise be highly probable.

II. Process and Limitations

During the process of building this algorithm, we initially chose "Total Deaths" as the main target predicted by our model. One limitation that resulted from this choice of target is the large variance among each event. Consequently, the final mean squared error to measure accuracy was significantly below an accurate threshold. This variance was further intensified by the dataset utilized, as it was shortened in size after cleaning and in reduced accuracy before the year 1960, creating potential gaps in our predictions. The final mean squared error value varies depending on each time the model undergoes training, but the estimated value for Total Deaths is around 511340408.6791202 for earthquakes, 14655438211.750992 for floods, 27546787.093117386 for storms, 48971657886.33338 for droughts, and

4351.850545376792 for air, depending on each different event, indicating high inaccuracy. Furthermore, accuracy was restricted by the methodologies of our investigation; as a result, two separate algorithms were built in an attempt to increase the precision of our results. The first model consisted of training our algorithm using our existing data and generating “synthetic” features to make our predictions (eg, randomized coordinates and a specific range of years and months) for total deaths. The second algorithm trained a separate model to predict multiple targets individually, each of them being the same as the features learning independently, except for the years and months. The predictions were built on synthetic data that was modified by separate algorithms in sequential orders. Inputting a set year (starting at 2025) and month (1-12, 12 per year), each target was calculated individually, starting with latitude using randomly generated inputs for longitude and Total deaths, followed by Longitude, using the predicted longitude as well as the random total deaths. Lastly, total deaths utilize both predicted Longitudes and Latitudes.

III. Findings

Multiple factors in this project presented a limitation to the precision of our predictions: the dataset imposed accuracy limitations previous to the year 1960; the dataset was significantly reduced following the cleaning process, leaving gaps among events that occurred and had invalid or empty values that were filtered out; the main prediction target for our model, total deaths, had a significantly high variance which difficulted the recognition of patterns learned by each decision tree; last, the synthetic data produced to generate predictions implied that each prediction was generated based on certain randomized decisions that were independent to the accuracy of the learning models. Given the lack of precision as a result of the aforementioned circumstances, interpreting the data would be speculative. Therefore, it can be determined that the results of this project are inconclusive.

Clustering Algorithm

I. Elbow Method & K-Means

To choose the number of clusters in which the data will be divided, we utilized the elbow method. The elbow method starts by running a K-means algorithm multiple times using a different number of clusters each time. For this project, we chose 10 rounds of K-Means using values of 1 through 10 clusters. Each time, the cohesion of the cluster is measured by the addition of the squared distances between each data point and the centroid of the cluster it belongs to; this is known as inertia. This generates a set of values: distance from the centroid of the cluster and the number of clusters. This creates a mathematical function that mimics the shape of an elbow. A low number of clusters results in a greater distance from each point to the centroid of the cluster. An excessive number of clusters results in very low distance from each data point to the centroid of the cluster, resulting in overfitting. In this context, overfitting results in data being grouped into excessively narrow categories. The elbow method aims to find the number of clusters that will group each data set into categories without resulting in excessive distance from the centroids of the clusters, without overfitting. Using the aforementioned relationship, this method captures the number of clusters that becomes an inflection point between utilizing insufficient and excessive clustering to find the ideal number.

Once the ideal number of clusters is determined by the elbow method, a random centroid is generated for each cluster. The Euclidean distance is computed to determine the distance between each centroid and the data points. The centroid is recalculated until convergence is reached. Convergence state is achieved once the distance between data points and centroids stops shifting after reaching an optimal location. Each climate event resulted in a different number of clusters according to the different predictions generated.

Time Series Analysis Algorithm

Time series analysis was utilized in this study to establish the stationarity of disaster data and identify anomalies in the overall death tolls. The death toll due to disasters can have a wide range. Therefore, anomaly detection is not an easy task because outliers can be actual events and not errors or exceptional circumstances. To combat this issue, stationarity was tested using the Augmented Dickey-Fuller (ADF) test, and two statistical techniques, Z-score analysis and the Interquartile Range (IQR) method, were employed to identify possible anomalies. Prior to any of this analysis being able to be performed, the data was imported and preprocessed, including both the original disaster dataset and a projected dataset.

The initial dataset was filtered, which included the addition of variables such as "Start Year," "Start Month," "Latitude," "Longitude," and "Total Deaths," while also eliminating entries that had missing values. The predicted data, which included projected deaths due to disasters, were obtained from a CSV file. With both datasets now cleaned and ready for use, I performed the ADF test for stationarity. The ADF test was employed to determine if the mean and variance of total deaths remain constant over time. If the p-value of the test was greater than 0.05, I considered the series non-stationary and concluded that there were trends or other extraneous variables affecting the data.c

In order to detect anomalies, two approaches were employed. The first approach employed Z-score analysis, which measures the number of standard deviations that a particular data point is away from the mean. A Z-score greater than a threshold value of 4 was flagged as an outlier. This method was suitable for data that was normally distributed but is sensitive to outliers, so since total deaths is very variable, I also used the IQR method. The IQR method approximates the range between the 75th and 25th percentiles and flags values that fall far beyond this range, using a multiplier of 4 to identify extreme outliers. It is less sensitive to skewed data and provides a useful complement to Z-score analysis. After I had run both of the anomaly detection algorithms, I combined their outputs to prevent missing large anomalies. The final list of anomalies was then exported to CSV for graphing and possible future analysis.

By combining stationarity testing and multiple methods of anomaly detection, I fixed the variability in total deaths while still picking out unusual patterns that might be representative of reporting differences or large outlier events.

Distribution of Work

Robert Romero worked on data cleaning and unit testing, Isabella Prince handled the machine learning algorithm and clustering algorithm, Madeline Bramson focused on time series analysis and the frontend, and Madeline Whitton worked on the frontend and visualization.

Works Cited

DZone Article:

DZone. (n.d.). *XGBoost: A deep dive into boosting*. DZone. Retrieved April 9, 2025, from https://dzone.com/articles/xgboost-a-deep-dive-into-boosting

Conference Paper:

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *arXiv preprint arXiv:1603.02754*. https://arxiv.org/abs/1603.02754

R Package Vignette:

Chen, T., & He, T. (2024). xgboost: eXtreme Gradient Boosting. *R package vignette*. https://cran.ms.unimelb.edu.au/web/packages/xgboost/vignettes/xgboost.pdf

Time Series Analysis:

Pierre, S. (2024, May 10). A guide to time series analysis in Python. Built In. https://builtin.com/data-science/time-series-python

Banerjee, P. (2020, August 30). *Complete Guide on Time Series Analysis in python*. Kaggle. https://www.kaggle.com/code/prashant111/complete-guide-on-time-series-analysis-in-python

Arif, A. (2024, June 20). *Guide to time-series analysis in Python*. Timescale Blog. https://www.timescale.com/blog/how-to-work-with-time-series-in-python

Bianchi, F. M. (2024). *Time series analysis with python*. Time series analysis with Python - Time series analysis with Python. https://filippomb.github.io/python-time-series-handbook/notebooks/00/intro.html