Natural Disaster Analysis

Atishay Jain
University of California, Riverside
ajain087@ucr.edu

Huy Dinh Tran
University of California, Riverside
htran197@ucr.edu

Daniel Li
University of California, Riverside
dli139@ucr.edu

Austin Lee
University of California, Riverside
alee235@ucr.edu

Abstract

This project provides a web application that can analyze natural disaster occurrences within the United States. The data can be filtered by state, year, and disaster type. Individual events can be detailed on click, and aggregate data can be visualized through bubble charts and timelapse videos. A standalone script runs four machine learning models that predict the trend of a given disaster event. Timeline plots will be generated that display the possibility of a disaster occurring. Our evaluation is done on multiple fronts. After 10 random selections of natural disasters, our search engine correctly contains no false positives and no false negatives. The web application has a response time of ~6197 ms on average. The machine learning models were evaluated using random inputs of state and disaster type pairs. Both classification models performed extremely well, with $\geq$ 92.5% accuracy on all inputs. Further evaluation must be done to ensure models are not overfitted and perform better than the default rate.

1 Introduction

1.1 Motivation

Natural disasters pose a serious threat to global populations. Examples include flooding, earthquakes, wildfires, and so on. The economic and environmental impacts posed by natural disasters cannot be understated. The United States alone has suffered from $2.15 trillion dollars lost in 2021 alone due to natural disasters [22]. For this project, we are implementing a search engine to filter out and output historical data on natural disasters. A forecasting disaster event feature will also be implemented to find patterns of disaster. To the best of our knowledge, there is no existing national disaster web application that filters, visualizes, and predicts natural disaster occurrences within the United States.

1.2 Problem Statement

The vast majority of disaster portals and directories are inadequate. With numerous distinct terms, catastrophe types, and custom filters, it is challenging to locate and search for a historical disaster on existing portals. There is no site that offers effective visualization capabilities and tools, even when appropriate data is available. This paper shows a developed portal or directory that retrieves historical data on natural disasters. A function/feature that allows users to filter and search disasters using certain fields will be included in the portal. Additional features such as data visualization and event forecasting are also implemented to aid the user experience.

1.3 Report Organization

The remainder of our report will be as follows: Related Works Section, Natural Disaster Analysis, Evaluation, Conclusion, Author Contributions, and References. The related works section will contain information on related works and how they are relevant to our project. The Natural Disaster Analysis will describe a high-level overview of our project and then explain each subsection in detail. The evaluation section will describe the performance metrics that will be used to evaluate our project as well as the actual performance of our project. The conclusion section will briefly summarize our project and some aspects that we could improve upon in the future. The author contributions section will summarize the contributions of each member of the team. The references section will contain all the references used in the project in standard ACM format.

2 Related Work

2.1 Pre-processing

It is important to clean and prepare the dataset before learning any models. Fortuny et. al. demonstrate that the quality of data is sometimes more important than the quantity of data [15]. And Lee et. al uses downsampling to reduce noise within a dataset [18]. Once we removed irrelevant features and empty rows, we found the rest of the data to be clean and did not need to do any downsampling.

2.2 Prediction/modeling

Due to the predictive features we implemented, we analyzed several papers related to machine learning. There are many possible models to utilize, so we turned to previous research for inspiration. Aalen utilized linear regression to predict oropharynx carcinoma treatment success rates with 94% accuracy [11]. LaValley utilized a logistic regression model to predict angina pectoris utilizing features such as total cholesterol level and sex with an accuracy of 86% [16]. Lavanya et. al. utilized a decision tree classifier to achieve 97% accuracy

on breast cancer detection data [17]. Rathore and Kumar achieved a high accuracy of 90% using a decision tree regression model to detect faults within software quality assurance [21]. Since all four different machine learning algorithms achieved strong results, we decided to try all of them to see which has the best performance for our dataset. We looked into other machine learning methodologies such as NGCU from Wang et. al. [25] and k-medoids clustering used by Prihandoko et. al. [20], but due to the irrelevance to our dataset or difficulty in implementation, we were unable to proceed with these approaches.

2.3 Visualization

Ali et. al. showcased various visualization tools that are designed to handle big data, including Tableau, Microsoft Power BI, and Plotly [12]. These tools inspired us to use Kepler [10] to visualize the data as it contained several key visualization features while being easy to connect to from our API.

2.4 Literature Reviews

Finally, there were many papers that provided high-level analysis of natural disaster management, such as Tang et. al [24], Zou et. al. [28], Yu et. al. [27], Tan et. al. [23], Arinta et. al. [13], Arslan et. al. [14], and Li et. al. [19].

3 Natural Disaster Analysis

3.1 High-level Overview

This project consists of four main parts: data processing, API development, web application, and machine learning. For data processing, this project utilizes SparkSQL [8] to clean the dataset. For API development, this project utilizes FastAPI to create a Rest API to communicate between our web application and our dataset. For our web application, this project utilizes reactJS [3] to create a web-based user interface that allows the user to search and filter through our database. In our web application, we also utilize Kepler.gl in order to create a visualization of our dataset. For the machine learning section, this project utilizes MLlib in order to create a machine learning model to predict the patterns of disaster occurrences.

3.2 Data Processing

In this section, we will describe our dataset and the steps we took to clean and process it for our project. Our dataset [26] is a high-level summary of all the federally declared disasters since 1953. This dataset is frequently updated and as of writing this paper, the dataset was last updated on 12/1/2022. Our dataset consists of 63699 entries and 23 fields.

For our project, we decided to exclude 12 fields from our dataset since the majority of the data in those fields is missing or because those fields are irrelevant to our project. The excluded fields are as follows: disaster_number, declaration_type, disaster_closeout_date, fips, place_code, designated_areas, last_ia_filing_date, hash, id, last_refresh. The fields that we decided to include in our dataset are as follows: fema_declaration_string, state, declaration_date, fy_declared, incident_type, declaration_title, incident_begin_date, incident_end_date, declaration_request_number, ih_program_declared, ia_program_declared, pa_program_declared, and hm_program_declared. In order to clean our dataset, we used libraries from PySpark [6] and pandas [2] in order to prune irrelevant attributes from our dataset as well as remove any row containing missing values. Our cleaned dataset is in the form of a Resilient Distributed Dataset [7] and Figure 1 presents a sample of the output file.

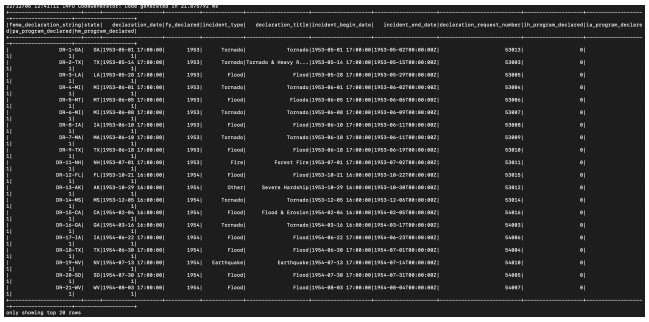

Figure 1. Sample Output File of Processed Data (After cleaning)

3.3 API Development

As a part of our project is a web application, we needed to create some type of intermediary to connect the web application and database. We decided to utilize FastAPI [9], a high-performance web framework for building API's with Python, to create a REST API communication between our web application and database. Our API executes SQL queries on our database and then transforms that data into JSON format for our web application to use. Our REST API also involves a filtering aspect where based on the input from our web application, our API will query different aspects of our database. The input should hold 3 elements; the first element is the state name, the second element is disaster type, and the third element is the year. If multiple fields in the query are not null, then our API will query for all natural disasters that fit the fields specified. Our API can also handle null values and when a null field is an input, our API will ignore the null field and query based on the non-null fields. For example, if our API received CA for state name, null for disaster type, and 1967 for the year, our API would filter through our dataset and return all of the natural disaster events that happened in California in the year 1967. After our API searches through the database and finds all the relevant entries, our API will return all relevant information about those entries for our

web application to either construct a visualization model or display in text format.

3.4 Web Application

Our web application contains four pages, a search page, a results page, a result details page, and a visualization page. This subsection will discuss each page in detail.

3.4.1 Search Page. The first page that you will initially see when we start up our web application is the home page which has a search bar to take input from the user. Once the input has been received, we will parse through the string and look for the keyword/keywords state, disaster, or year and a value following the keyword.. The input must be in the form of <keyword>: <value>. A visual example is shown in Figure 2. Once we have finished parsing the string, we will use our API to query our dataset and receive the relevant data. We will then take the data and transform it into JSON format to be displayed on our Results Page.

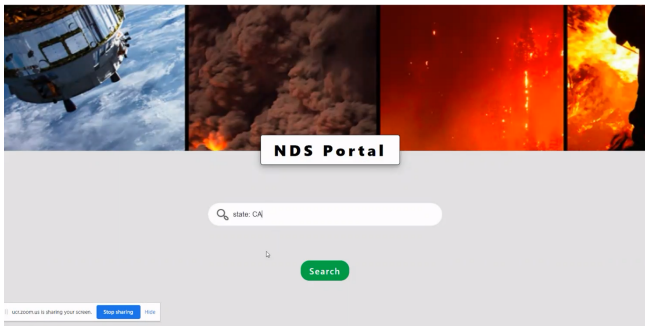

Figure 2. Example of Search Page searching the dataset for natural disasters in the state of California

3.4.2 Results Page. The Results Page will display the total number of matching entries on the top left and the matching entries' corresponding information will be listed below in the format of Disaster Type, Disaster Declaration, Date, State, Fema Declaration, and a picture for easy visualization. A visual example will be displayed in Figure 3. Our results page also has the added functionality for filtering through the matching entries by state, disaster type, and year. To utilize this functionality, the user must click on Categories which will create a drop-down menu to choose which category the user will want to filter by. Once the user has chosen the category, they must input the word they want to filter by and then click the filter button to display the filtered results. A visual example will be displayed in Figure 4.

3.4.3 Result Details Page. The Result Details Page can be accessed by clicking on any of the entries displayed in the Results Page. The Result Details Page will display additional information about the selected natural disaster. The following information will be displayed in text format: disaster

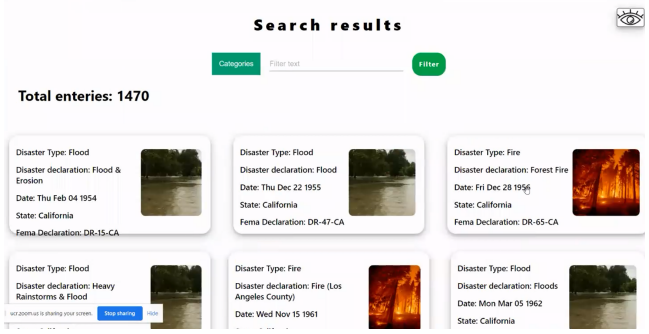

Figure 3. Example of Results Page with no filtering

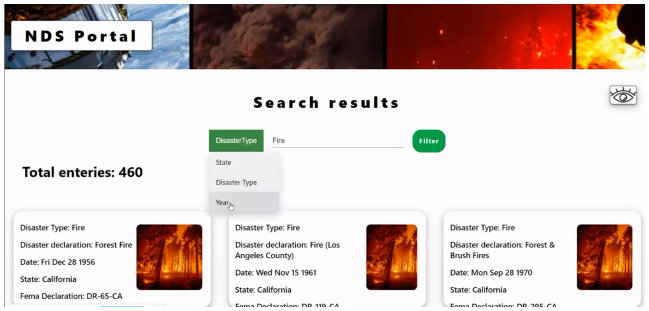

Figure 4. Example of Results Page with filtering

type, date, state, fema declaration, individuals and households program, individual assistance program, public assistance program, hazard mitigation program, and declaration request number. A visual example will be shown in Figure 5.

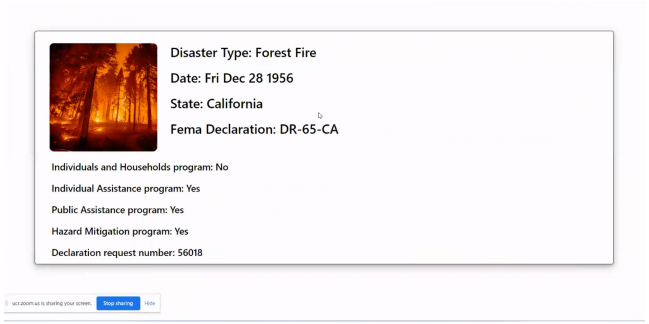

Figure 5. Example of result details page

3.4.4 Visualization Page. The Visualization Page can be accessed by clicking on the eye button on the top right corner of the Results Page. The Visualization Page utilizes Kepler.gl, an open-source geospatial analysis tool for large-scale datasets, to create a visualization model of the queried dataset. The visualization model will show a map of the United States with different colored points to represent the different natural disasters that happened in a state. There

will be a list of options to customize the visualization model. Some of the options include color, outline, radius, and label. Any of these options can be customized to scale based on a factor that the user can choose. For example, the user could scroll down to the radius options and choose to scale the size/radius of the points based on the intensity or in other words number of disaster occurrences. A visual example will be shown in Figure 6. Our visualization model can also simulate a time-lapse of our dataset based on a factor that the user can choose. This can be done by going to the Filters option and selecting the field tpep_pickup_datetime. Once this field has been selected a bar graph will be displayed at the bottom of the screen. The user can then click on the Y-Axis button and choose a factor to create a time-lapse of. For example, the user can go to filters, select the field tpep_pickup_datetime, and choose the factor of intensity. The following would create a time-lapse showing the date and the number of occurrences of natural disasters in each state on that date. A visual example of this will be shown in Figure 7.

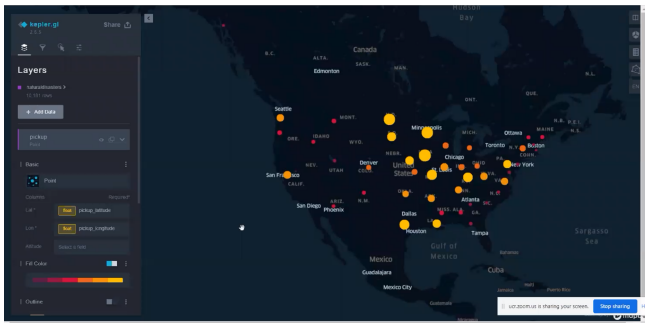

Figure 6. Visualization Page with point radius scaled to intensity/# of occurrences

Figure 7. Visualization Page with time-lapse based on intensity/#of occurrences

3.5 Machine Learning

Spark is equipped with MLlib [5] as a tool to perform machine learning models. There are two versions of MLlib which are the data frame [8] and RDD versions. Since the RDD version is being minimally maintained and is more complicated to use compared to the data frame version, we chose the data frame version as our tool to perform machine learning models. The machine learning models will predict the future dates of the possibility of event occurrences. The model will present a timeline graph for the next x amount of years. With that in mind, we chose to implement 2 regression and 2 classification models which are Logistic Regression Classification, Linear Regression, Decision Tree Classification, and Decision Tree Regression. Figure 8, 9, 10, and 11 show the result of running the script with user inputs of CA, Flood, and 3 prediction years.

For the implementation, firstly, our model will get 3 user inputs which are the state and the disaster type, and the number of years that the user wants to predict. The model then filters out everything else that is not related to the input and treats it as training data. We are using UNIX time [1] for dates since it's simpler to train using the models. The training data frame has two features. One is the date in UNIX time which is used for the training feature and the second is the "occur" feature which is the target that says if the disaster happened on this day or not. Since the dataset only gives the date that an event occurred, we need to add in the days that the events did not occur. We just appended days that event didn't occur starting from the earliest to the latest recorded event. For testing data, the user can set the number of years that they want the model to predict and it will give 4 timeline graphs of the potential event that will happen for the next X amount of years.

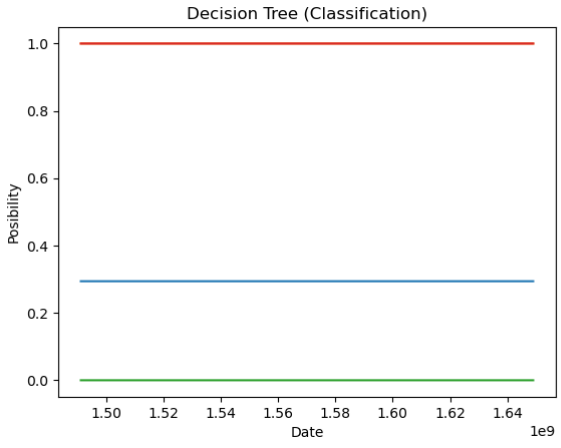

Figure 8. Decision Tree Classification

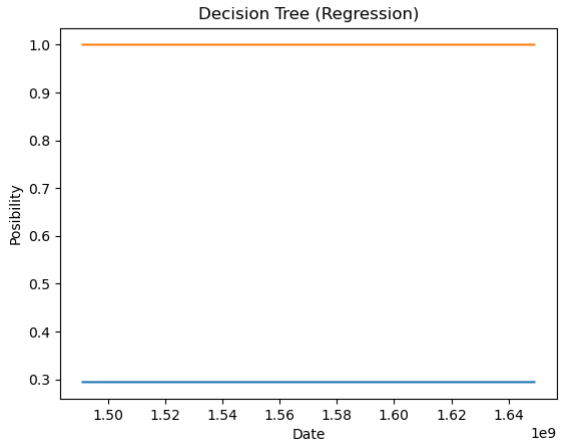

Figure 9. Decision Tree Regression

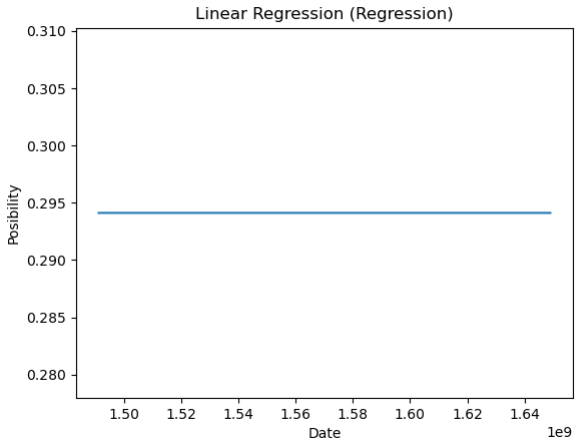

Figure 10. Linear Regression

4 Evaluation

To evaluate our project, we came up with three performance metrics.

4.1 Search Engine Accuracy

We can measure the accuracy of our search engine. We evaluated the accuracy of our search engine by taking a randomly selected FEMA declared natural disaster and seeing if we could find it through our search engine. We also compared the information of the randomly chosen natural disaster to the result returned by our search engine to verify that our search engine was accurate. We repeated this test 10 times with 100% accuracy but for the sake of brevity, we will explain and display the results for only one of the tests. As displayed in Figure 12, we selected a random FEMA declared natural disaster which we then searched for in our search engine. As displayed in Figure 13, we were able to find the

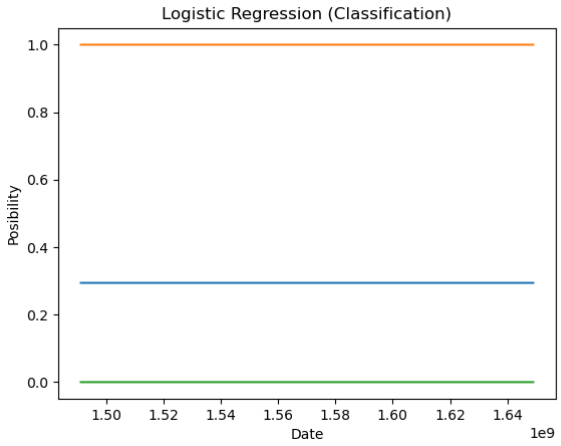

Figure 11. Logarithmic Regression Classification

randomly selected natural disaster through our search engine and verify that all the information displayed was correct.

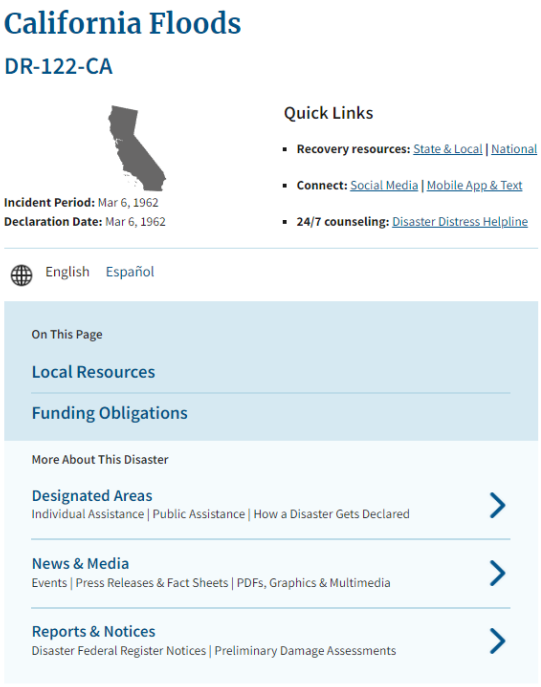

Figure 12. Randomly selected FEMA declared natural disaster

4.2 Execution Time of Search Engine

We can measure the execution time of our search engine. We evaluated the execution time of our search engine using the python library time [4] to measure how long it takes for our API to return the queried data. After running this test

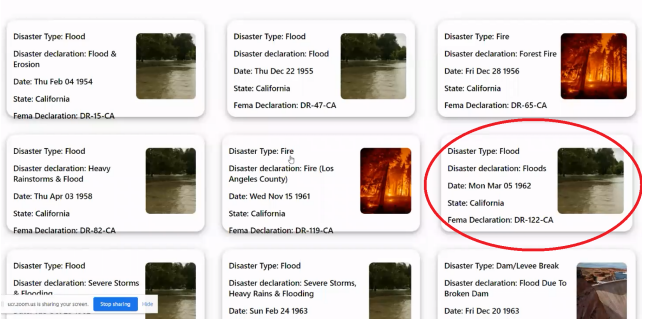

Figure 13. Accuracy of Search Engine Results

30 times, we found the average execution time of our search engine to be ~6197 milliseconds. Although the average execution time of our search engine is ~6197 milliseconds, we believe that this statistic is not the optimal way of testing the efficiency of our search engine as the actual execution time of a query heavily depends on the amount of data returned by our API. For example, when querying our database for all natural disasters of the type flood, the execution time was ~12874 milliseconds which is almost double the average execution time of our search engine. Another example of this is when we queried our database for all natural disasters that occurred in California, we received an execution time of ~4037 milliseconds which is quite a bit faster than the average execution time.

4.3 Machine Learning Accuracy

Table 1 shows accuracy and Root Mean Squared Error (RMSE) for the four models. We calculated the accuracy and RMSE by dividing the training data into 70% training and 30% testing data. For the results, we can see that the models give high accuracy or low RSME outputs. However, based on the outputs, we concluded that the models need more quantitative and correlated features and additional data entries to improve the capabilities of the models.

5 Conclusion

The Natural Disaster Analysis tool is a web application that provides filtering, visualization, and prediction capabilities for natural disaster data within the United States. Data can be filtered on multiple features, including state, year, and disaster type. Four different machine learning models were trained. We evaluated our application based on the search engine accuracy, website response time, and machine learning model accuracy. We can say with confidence that our project's search engine is very accurate. Additionally, many of the classification models have high accuracies of over 90%. The average response time data is potentially inconsistent, which leads to inconclusive evidence for that aspect of our application.

5.1 Future Work

These are multiple aspects of our project we aim to improve in the future.

- Filter functionality
  We want the user to be able to choose from more filter options on the search results page.
- Improved Visualization
  We hope to present several anchors on the map in our visualization section that would indicate various disaster events that are connected to the first search query.
- Better dataset
  The dataset we used contained few characteristics and duplicate rows and values. Therefore, we wish to combine different datasets to construct our own. We would be able to build better machine learning models if we use a good dataset.

6 Author Contributions

Table 2 shows individual contributions from each author.

References

[1] 1970. Unix Epoch. https://en.wikipedia.org/wiki/Unix_time
[2] 2008. Pandas. https://pandas.pydata.org
[3] 2013. React. https://reactjs.org
[4] 2013. Time - Time Access and conversions. https://docs.python.org/3/library/time.html
[5] 2014. MLlib: Apache Spark. https://spark.apache.org/mllib
[6] 2014. PySpark. https://spark.apache.org/docs/latest/api/python/
[7] 2014. RDD. https://spark.apache.org/docs/latest/rdd-programming-guide.html
[8] 2014. Spark SQL. https://spark.apache.org/sql
[9] 2018. FastAPI. https://fastapi.tiangolo.com
[10] 2018. kepler.gl. https://kepler.gl
[11] Odd O. Aalen. 1989. A linear regression model for the analysis of life times. *Statistics in Medicine* 8, 8 (1989), 907–925. https://doi.org/10.1002/sim.4780080803
[12] Syed Mohd Ali, Noopur Gupta, Gopal Krishna Nayak, and Rakesh Kumar Lenka. 2016. Big data visualization: Tools and challenges. In *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*. 656–660. https://doi.org/10.1109/IC3I.2016.7918044
[13] Rania Rizki Arinta and Emanuel Andi W.R. 2019. Natural Disaster Application on Big Data and Machine Learning: A Review. In *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*. 249–254. https://doi.org/10.1109/ICITISEE48480.2019.9003984
[14] Muhammad Arslan, Ana-Maria Roxin, Christophe Cruz, and Dominique Ginhac. 2017. A Review on Applications of Big Data for Disaster Management. In *2017 13th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*. 370–375. https://doi.org/10.1109/SITIS.2017.67
[15] Enric Junqué de Fortuny, David Martens, and Foster Provost. 2013. Predictive Modeling With Big Data: Is Bigger Really Better? *Big Data* 1, 4 (2013), 215–226. https://doi.org/10.1089/big.2013.0037 arXiv:https://doi.org/10.1089/big.2013.0037 PMID: 27447254.
[16] Michael P. LaValley. 2008. Logistic Regression. *Circulation* 117, 18 (2008), 2395–2399. https://doi.org/10.1161/CIRCULATIONAHA.106.682658

State & Event	CA & Flood	TX & Fire	PA & Severe Storm	HI & Volcanic Eruption
Linear Regression (Regression) RMSE	0.116	0.285	0.105	0.013
Decision Tree (Regression) RMSE	0.106	0.238	0.102	0.017
Logistic Regression (Classification) accuracy	0.986	0.910	0.989	0.999
Decision Tree (Classification) accuracy	0.987	0.925	0.988	0.999

Table 1. RMSE and Accuracy for different models given random input state and disaster type

Name	Contribution
Atishay Jain	Spark, Web application, Rest API
Austin Lee	Report, Spark
Daniel Li	Report, Spark
Huy Dinh Tran	Report, Machine Learning Models

Table 2. Contributions of each group member

[17] D. R. Lavanya and K. Usha Rani. 2012. ENSEMBLE DECISION TREE CLASSIFIER FOR BREAST CANCER DATA. *International Journal of Information Technology and Computer Science* 2 (2012), 17–24.

[18] Wonjae Lee and Kangwon Seo. 2022. Downsampling for Binary Classification with a Highly Imbalanced Dataset Using Active Learning. *Big Data Research* 28 (2022), 100314. https://doi.org/10.1016/j.bdr.2022.100314

[19] Yunyao Li, Ziyang Liu, and Huaiyu Zhu. 2014. Enterprise Search in the Big Data Era: Recent Developments and Open Challenges. *Proc. VLDB Endow.* 7, 13 (aug 2014), 1717–1718. https://doi.org/10.14778/2733004.2733071

[20] Prihandoko, Bertalya, and Muhammad Iqbal Ramadhan. 2017. An analysis of natural disaster data by using K-means and K-medoids algorithm of data mining techniques. In *2017 15th International Conference on Quality in Research (QiR) : International Symposium on Electrical and Computer Engineering*. 221–225. https://doi.org/10.1109/QIR.2017.8168485

[21] Santosh Singh Rathore and Sandeep Kumar. 2016. A Decision Tree Regression Based Approach for the Number of Software Faults Prediction. *SIGSOFT Softw. Eng. Notes* 41, 1 (feb 2016), 1–6. https://doi.org/10.1145/2853073.2853083

[22] ADAM B. SMITH. 2022. 2021 U.S. billion-dollar weather and climate disasters in historical context. (2022).

[23] Ling Tan, Ji Guo, Selvarajah Mohanarajah, and Kun Zhou. 2021. Can we detect trends in natural disaster management with artificial intelligence? A review of modeling practices. *Natural Hazards* 107 (07 2021), 1–29. https://doi.org/10.1007/s11069-020-04429-3

[24] Ling Tang, Jieyi Li, Hongchuan Du, Ling Li, Jun Wu, and Shouyang Wang. 2022. Big Data in Forecasting Research: A Literature Review. *Big Data Research* 27 (2022), 100289. https://doi.org/10.1016/j.bdr.2021.100289

[25] Jingyang Wang, Xiaolei Li, Jiazheng Li, Qiuhong Sun, and Haiyao Wang. 2022. NGCU: A New RNN Model for Time-Series Data Prediction. *Big Data Research* 27 (2022), 100296. https://doi.org/10.1016/j.bdr.2021.100296

[26] U.S. Government Works. 2022. US Natural Disaster Declarations. https://www.kaggle.com/datasets/headsortails/us-natural-disaster-declarations

[27] Manzhu Yu, Chaowei Yang, and Yun Li. 2018. Big Data in Natural Disaster Management: A Review. *Geosciences* 8, 5 (2018). https://doi.org/10.3390/geosciences8050165

[28] Benyuan Zou, Jinguo You, Quankun Wang, Xinxian Wen, and Lianyin Jia. 2022. Survey on Learnable Databases: A Machine Learning Perspective. *Big Data Research* 27 (2022), 100304. https://doi.org/10.1016/j.bdr.2021.100304