Richter's Predictor

Rithvik Ganesh
Arizona State University
Tempe, Arizona, USA
rganes11@asu.edu

Poorvik Dharmendra
Arizona State University
Tempe, Arizona, USA
pdharmen@asu.edu

Rahul Kalluri Srinivas
Arizona State University
Tempe, Arizona, USA
rskallur@asu.edu

Prudhvi Mutyala
Arizona State University
Tempe, Arizona, USA
pmutyal2@asu.edu

Rohith Danti
Arizona State University
Tempe, Arizona, USA
rdanti1@asu.edu

ABSTRACT

This study investigates the potential of machine learning to predict earthquake-induced building damage using a post-disaster dataset from the 2015 Gorkha earthquake in Nepal. Our objective is to develop a reliable prediction model that leverages structural and legal ownership attributes of buildings. By employing a multi-algorithm ensemble approach with Random Forest Classifiers, Neural Networks, XGBoost, and Naive Bayes, we aim to capture both linear and non-linear relationships within the data for improved accuracy and robustness. The micro-averaged F1 score will be the primary evaluation metric, along with confusion matrices to gain insights into specific classification errors. This comprehensive approach is expected to outperform single-model methods and identify key factors influencing damage severity, ultimately contributing to improved disaster resilience and response strategies.

KEYWORDS

Random Forest Classifiers, Neural Networks, XGBoost, Naive Bayes, KNN, SVM

1 INTRODUCTION

Earthquakes pose a significant threat to infrastructure, highlighting the need for improved prediction of building damage to aid in disaster response and recovery. This study explores the potential of machine learning to address this challenge. We leverage a comprehensive post-disaster dataset from the 2015 Gorkha earthquake in Nepal, which includes detailed information on building structure and legal ownership. Our primary objective is to develop a reliable model that can predict the level of damage sustained by buildings based on these attributes. By employing a multi-algorithm ensemble approach incorporating Random Forest Classifiers, Neural Networks, XGBoost, and Naive Bayes, we aim to capture both linear and non-linear relationships within the data, potentially leading to improved model accuracy and robustness.

2 RELATED WORK

[1] investigates the potential of machine learning for rapid post-earthquake building damage assessment using a comprehensive dataset from the 2014 South Napa earthquake. Focusing on residential buildings and employing data on spectral acceleration, fault distance, and building characteristics, the study explores the effectiveness of various algorithms including discriminant analysis, k-nearest neighbors, decision trees, and random forests. While limitations exist in generalizability due to the single earthquake and potential bias from the skewed distribution of damage tags (mostly yellow), the random forest model achieved 66% accuracy in predicting damage levels (red, yellow, green), suggesting promise for machine learning in enhancing data-driven decision-making for post-earthquake response and recovery.

[2] explores the development of a machine learning model for seismic damage prediction using data from the 2017 Puebla-Morelos Mexico earthquake. Focusing on 340 buildings in Roma and Condesa neighborhoods of Mexico City, the study employs various machine learning algorithms, with Random Forest achieving over 65% accuracy in predicting damage levels. The model considers building location, seismic demand, and building height, potentially informing future seismic risk mitigation strategies. While offering a novel approach by integrating post-earthquake data with machine learning, limitations include the model's generalizability due to its focus on a specific earthquake and geographical area, a finite dataset size, and potential bias from class imbalance in the damage classification.

[3] investigates machine learning for earthquake damage prediction in Nepal, comparing Neural Networks (NN) and Random Forests (RF). Focusing on eight tectonic indicators and past vibration data, the study finds that RF outperforms NN with a notable F1 score of 74.32%. This approach offers several strengths: it comprehensively evaluates two common machine learning methods, utilizes a novel dataset incorporating real-world earthquake factors, and demonstrates practical application through a significant improvement in F1 score with RF. Additionally, the micro-averaged F1 score ensures a balanced evaluation, and the described methodology is simple for computer implementation, aiding rapid damage assessment. However, limitations exist. The focus on a single earthquake and reliance on specific data sources might limit generalizability. While the RF model performs well, further optimization of the NN architecture could be explored, and achieving higher accuracy with 750 estimators in the RF model comes at the cost of increased computational complexity, highlighting a trade-off between performance and efficiency for disaster management applications.

[4] explores the growing impact of machine learning (ML) and deep learning (DL) on disaster management across prediction, detection, and response phases. It reviews various case studies and applications to showcase how these technologies are implemented in

real-world scenarios. A key strength lies in its comprehensiveness, offering a broad view of current ML/DL applications in disaster management. Furthermore, the paper analyzes future research trends and challenges, paving the way for improved efficacy. One highlighted strength is a DL-based framework that combines VGG-19, CNN, and LSTM models, achieving an impressive F1 score of 0.857 on a large Twitter dataset. This framework demonstrates exceptional multi-task learning efficiency in disaster assessment by automating loss weighting and capturing data correlations. However, limitations exist. The rapid evolution of AI technologies necessitates frequent updates to maintain relevance. Additionally, while covering a wide range of applications, the analysis of specific methodologies and their comparative effectiveness is limited. Potential bias in case study selection could also skew the perceived effectiveness of certain approaches. Finally, the reliance on heuristic thresholds in ML algorithms for early warnings can lead to false alarms, highlighting the need for improved accuracy and reduced reliance on heuristics.

[5] proposes a machine learning approach, specifically active learning, for regional seismic risk assessment of infrastructure systems. The framework incorporates spatial data, ground motion simulation, fragility analysis, and machine learning to comprehensively evaluate seismic risk across a region. Active learning prioritizes informative data points during model training, improving efficiency and reducing computational costs. A case study on a California transportation network demonstrates the methodology's effectiveness in identifying vulnerable infrastructure elements. While the integration of machine learning strengthens traditional methods and the active learning approach optimizes resource allocation, limitations exist. The effectiveness relies heavily on data quality and availability, and the paper acknowledges the need for further research on potential biases in active learning data selection. Additionally, the framework's current focus on transportation networks calls for future studies encompassing diverse infrastructure types for a more comprehensive assessment. Finally, while the California case study validates applicability, further studies in regions with varying conditions are needed to confirm generalizability.

[6] explores the potential of deep learning for classifying earthquake-damaged buildings based solely on textual descriptions of the damage. This approach aims to improve the efficiency and accuracy of post-earthquake assessments, potentially automating building damage classification and aiding disaster response efforts. A key strength lies in its innovative use of textual data, offering insights into rapid and precise damage assessment. However, limitations exist. The reliance on text descriptions might miss nuances detectable through visual inspection, and the generalizability of the deep learning model to various geographic and structural contexts needs further validation. Additionally, integrating this approach with existing disaster response protocols requires further investigation to ensure practical implementation.

[7] explores machine learning's potential for earthquake magnitude prediction using 3000 seismic time series measurements from Turkey. It compares various algorithms including Linear Regression, Decision Trees, and Artificial Neural Networks, evaluating their performance through metrics like regression rates and mean squared error. A key strength lies in its exploration of multiple techniques, highlighting the potential of Artificial Neural Networks for local magnitude prediction (83% regression rate) and depth prediction (69% regression rate). However, limitations exist. The study focuses only on smaller earthquakes (local magnitude 1.0-5.0) and utilizes a limited dataset (3000 measurements), potentially affecting generalizability. Additionally, the paper lacks details on data preprocessing and model interpretability, and high error rates in predicting latitude and longitude suggest the need for either longer time series data or further model refinement.

[8] investigates Back-Propagation Neural Networks (BPNNs) for earthquake prediction using seismic motion data. By training a BPNN on past earthquake data from Indonesia (date, time, location), the study aims to identify movement patterns that can be linked to earthquake prediction. A strength lies in leveraging the pattern recognition capabilities of ANNs and utilizing real-world data. The paper details a specific BPNN architecture and mentions using Mean Squared Error for evaluation. However, limitations exist. The paper lacks details on data preprocessing, feature selection rationale, and the impact of potentially relevant features. Furthermore, it omits quantitative results or performance metrics, making it difficult to assess the model's effectiveness. Additionally, generalizability beyond the Indonesian case and strategies for overfitting or imbalanced data are not addressed.

3 DATA

Our analysis delves into a post-disaster dataset, offering a picture of the earthquake's impact on Nepal. This dataset, collected through surveys by Kathmandu Living Labs and the Central Bureau of Statistics, stands out as one of the largest of its kind ever assembled following a disaster. The dataset has 347,469 instances and the data captures details across 39 features, including a unique identifier for each building and various aspects relevant to damage assessment. One key aspect of the data is the class distribution, which reveals an imbalance. The most frequent damage level, accounting for 87,218 instances, is complete destruction (Class 3). This is followed by moderate damage (Class 2) at 148,259 instances, and lastly, minor damage (Class 1) at 25,124 instances. To ensure the robustness of our model evaluation, the data has been meticulously pre-split into a training set of 260,601 instances (75%) and a testing set of 86,868 instances (25%). This split allows us to train our model on the majority of the data while reserving a portion for unbiased assessment of its generalizability on unseen information.

3.1 Feature Selection

The feature selection process plays a crucial role in enhancing the predictive accuracy and efficiency of the models. We utilize the Chi-squared test to rank and select categorical features based on their statistical significance to the target variable, "damage grade".

In our approach to handling categorical predictors, we transform these features into dummy variables, converting them into a binary numerical format that is well-suited for processing by our models. This transformation is crucial because many machine learning algorithms require numerical input to perform optimally. Following this, we utilize the SelectKBest method with the Chi-squared test to evaluate and retain the most significant features. This enables us to concentrate our analysis on the variables that have the most predictive power regarding the outcome, streamlining our model's complexity and potentially enhancing its predictive accuracy and general performance.

This focused approach not only simplifies the model but also potentially increases the training speed and generalizability by reducing overfitting. The selected features are subsequently used to train various predictive models discussed in the following section.

4 METHODS

4.1 Neural Network Model

The neural network architecture employs a single input layer feeding into two hidden layers with ReLU or Tanh activation functions. This structure aims to capture the intricate relationships between building features and damage levels. The final output layer utilizes a softmax activation function to predict damage categories (complete destruction, medium damage, low damage) as probabilities. A categorical cross-entropy loss function measures the discrepancy between these predicted probabilities and the actual damage levels. The Adam optimizer is leveraged to optimize the model's training process.

A comprehensive hyperparameter tuning process using grid search will identify the optimal configuration for learning rates and hidden layer dimensions. To assess the model's effectiveness, the micro-average F1 score, which considers both precision and recall across all damage classes, will be the primary metric. Additionally, metrics like macro-average F1 score, accuracy, precision, and recall will be reported for each damage class, providing a detailed understanding of the model's performance in classifying different levels of earthquake damage.

4.2 Naive Bayes Model

Our approach leverages the Naive Bayes classifier, a probabilistic model based on Bayes' theorem. Despite assuming independence between features, Naive Bayes has proven effective in classification tasks and is suitable for earthquake damage prediction. This model treats each feature in a feature vector as independent and utilizes different variants like Gaussian Naive Bayes (for continuous features) and Multinomial Naive Bayes (for discrete features) to estimate class probabilities. During training, the model learns these probability distributions from the data, allowing it to calculate the probability of each damage class given a specific building's features.

To assess the Naive Bayes model's effectiveness in classifying earthquake damage, we will utilize a combination of evaluation techniques. Standard classification metrics like accuracy, precision, recall, and F1-score will be calculated for each damage level (complete destruction, medium damage, low damage) to provide a comprehensive picture of the model's performance across different severities. Additionally, confusion matrices will offer insights into how well the model distinguishes between these classes. We will further explore hyperparameters specific to Naive Bayes, such as the Laplace smoothing parameter, through experimentation. To ensure robust evaluation and generalizability, k-fold cross-validation will be employed. This technique involves splitting the data into multiple folds and iteratively training/testing the model on different combinations, providing a more reliable assessment of its real-world applicability. Finally, we will analyze the model's performance across various configurations using metrics like log-likelihood, accuracy, and class-wise F1 scores. This comprehensive evaluation process will guide us in selecting the optimal configuration for the Naive Bayes model in our earthquake damage prediction task.

4.3 K Nearest Neighbors (KNN)

The K Nearest Neighbors (KNN) algorithm is well-suited for predicting earthquake damage grades in the 2015 Gorkha earthquake dataset. This complex dataset, rich with details on building structure and ownership, represents a scenario where KNN's reliance on local data point similarity can effectively capture intricate patterns without overly strong assumptions. To optimize KNN's performance, we will conduct experiments using the pre-defined data splits for training and testing. Feature selection will be employed to identify the most informative features, potentially improving model performance and reducing computational costs. Hyperparameter tuning will involve varying the number of neighbors (k) through grid search with cross-validation to find the optimal k that minimizes error. We will also experiment with different distance metrics (e.g., Euclidean, Manhattan) and weighting schemes (uniform vs. distance-based) to determine the configuration that best captures the relationships within the data and yields the most accurate damage grade predictions.

The KNN model's performance is expected to be competitive, with the optimal number of neighbors (k), distance metric, and weighting method significantly impacting accuracy. We will benchmark KNN against other models (Random Forest, Neural Networks, XGBoost, Naive Bayes) to assess its ability to capture local patterns in the data. While KNN might be outperformed by models handling complex non-linear relationships or high dimensionality, its performance can potentially be improved through feature engineering, advanced preprocessing, or ensemble methods with Random Forest or XGBoost. Fine-tuning k, distance metric, and weighting will be crucial, with a potentially small k value being optimal due to the high dimensionality of the data.

4.4 Support Vector Machines (SVMs)

Support Vector Machines (SVMs) are a strong candidate for analyzing the high-dimensional data from the Nepal earthquake due to their ability to handle complex datasets. This is ideal because the data likely includes many features (number of stories, materials) that influence building damage. SVMs work by creating a separation line (hyperplane) in a high-dimensional space that best divides the data points based on damage class (complete destruction, medium damage, etc.). This line maximizes the distance between the closest data points from each class (support vectors). By analyzing these support vectors, the model learns the key characteristics that differentiate damage levels. When encountering a new building, the SVM classifies it based on which side of the hyperplane it falls on in the high-dimensional space. To ensure optimal performance, we will explore different kernel functions (like the Radial Basis Function) to capture potential non-linear relationships between building features and damage. Additionally, we will fine-tune parameters like cost (C) and epsilon (ϵ) to balance maximizing the margin between damage classes, allowing for some training errors, and avoiding overfitting or being too sensitive to noise in the data.

To achieve optimal performance on the Nepal earthquake data, we will meticulously fine-tune the SVM model's hyperparameters. Through grid search, we will systematically train and evaluate the model on various combinations of cost (C), epsilon (ϵ), and kernel functions. This involves using a subset of the data for training and another unseen subset for validation. The configuration that yields the best F1 score on the validation set will be selected for the final model. Additionally, k-fold cross-validation will ensure the model generalizes well to unseen data. Here, the data is split into k sections, with the model being trained on k-1 sections and tested on the remaining one. This process is repeated k times, providing a reliable assessment of the model's real-world applicability. To gauge the model's effectiveness, we will employ a combination of evaluation metrics like micro-averaged F1 score (overall accuracy), macro-averaged F1 score, accuracy, precision, and recall. By capitalizing on SVM's strengths and meticulously tuning its parameters, we aim to develop a reliable model for classifying earthquake damage using the Nepal earthquake data. This model has the potential to be a crucial tool for prioritizing post-earthquake response efforts and directing resources toward the most heavily impacted buildings.

4.5 Random Forest Classifier

The Random Forest classifier is well-suited for our earthquake damage prediction task. This model is robust to overfitting and offers valuable insights through feature importance ranking. It can efficiently process both numerical and categorical data, making it a strong choice for analyzing post-disaster datasets. To optimize performance, we will conduct experiments with key parameters like the number of trees in the forest (n_estimators) and the maximum depth of each tree (max_depth). 5-fold cross-validation will ensure robust evaluation, where the data is split into five sections for training the model on four sections and testing on the remaining one, repeated five times. Grid search or random search will be employed within this setup to identify the parameter combination yielding the best performance on the validation data. Finally, the model will be evaluated on the unseen test dataset using metrics like accuracy, F1-score, and potentially a confusion matrix to assess its generalizability and ability to distinguish between different damage severities.

The Random Forest model's performance will be assessed using a combination of metrics. Accuracy will measure the overall proportion of correctly predicted damage grades. To delve deeper and account for the multi-class nature of the problem (multiple damage levels), we will employ the micro-averaged F1 score. This metric balances the model's precision (ability to identify true positives) and recall (ability to capture all actual positives) across all damage classes. Additionally, a confusion matrix will be utilized to provide a more granular view of the model's performance for each damage level (complete destruction, medium damage, low damage). This will help identify any potential biases the model might have towards certain damage severities.

4.6 Extreme Gradient Boosting (XGBoost)

As a powerful machine learning tool, XGBoost excels in both accuracy and efficiency, making it suitable for classifying earthquake damage on buildings. XGBoost builds on gradient boosting frameworks to create a sequence of trees that learn from each other's mistakes, allowing it to capture complex patterns in earthquake damage data. This is particularly useful for handling large and potentially sparse datasets (missing or zero-inflated values) often encountered in post-disaster scenarios. Furthermore, XGBoost's strengths lie in its ability to work with these large datasets efficiently due to its scalability and efficient resource utilization. Additionally, built-in cross-validation and early stopping mechanisms help prevent overfitting and ensure the model generalizes well to unseen data. XGBoost also offers flexibility in hyperparameter tuning, allowing us to tailor the model to the specific characteristics of earthquake damage data. Feature importance scores from XGBoost can further guide us in prioritizing building mitigation strategies. To optimize performance, we will utilize XGBoost's built-in 5-fold cross-validation with early stopping and experiment with key parameters like the number of trees, tree depth, learning rate, and regularization parameters.

The XGBoost model's performance will be gauged through a combination of metrics. Accuracy will be the primary indicator of how effectively the model predicts the correct damage grades overall. To account for potential imbalances in the data (where some damage severities might be more frequent than others), we will employ the micro-averaged F1 score, which incorporates both false positives and false negatives. This will provide a more comprehensive picture of the model's performance by analyzing its ability to distinguish between all damage classes (complete destruction, medium damage, low damage).

5 EXPERIMENTATION AND RESULTS

5.1 Neural Networks

To optimize the Neural Network's performance, a grid search was conducted. This technique evaluated various combinations of learning rates (0.0001, 0.0005, 0.001, 0.01) and hidden layer dimensions (64, 128, 256, 512) using the Adam optimizer and cross-entropy loss function. The search was performed on both the training and test data to identify the configuration yielding the best F1-score.
The grid search identified a learning rate of 0.001 and a hidden layer dimension of 128 as optimal on the training data. A final model was trained using these hyperparameters and evaluated on the unseen test data, achieving an F1-score of 0.65. To gain a comprehensive understanding of the model's performance, additional metrics like precision, recall, and a confusion matrix (Figure 1) were employed.

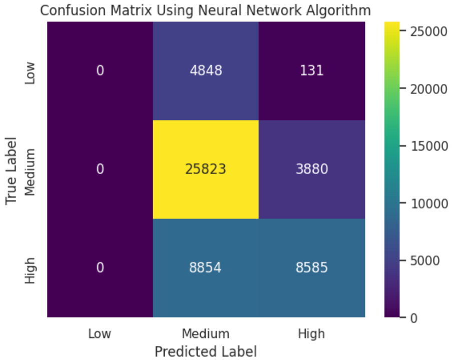

Figure 1: Confusion Matrix for Neural Network

- There seems to be a significant class imbalance in the data. Class 1 (low damage) has no instances correctly classified, suggesting there might be very few examples of low-damage buildings in the data.
- The model performed well on Class 2 (likely medium damage) with a precision of 0.87 (identified most Class 2 buildings correctly) and an F1-score of 0.75 (balanced performance between precision and recall).
- The model's performance on Class 3 (high damage) is moderate with a precision of 0.50 (identified half of the high-damage buildings correctly) and an F1-score of 0.57.

Additional performance metrics were computed to provide a holistic view of the classifier's effectiveness:

- Average Test F1 Score (Micro): **0.65**
- Average Test Precision Score (Micro): **0.65**
- Average Test Recall Score (Micro): **0.65**

The limited data available for low-damage buildings (Class 1) creates a significant bias in the model, causing it to excel at identifying buildings with medium (Class 2) and high damage (Class 3). This imbalance inflates the overall accuracy, which is heavily influenced by the abundance of Class 2 and Class 3 data. To improve low-damage detection, we can either acquire more data for Class 1 or explore deeper network architectures. With a richer dataset or a more complex model, the network could potentially learn the specific patterns associated with low-damage buildings and achieve a more balanced performance across all damage severities.

5.2 Naive Bayes

For our investigation into predicting earthquake-induced building damage, we employed a Naive Bayes classifier as part of our multi-algorithm ensemble approach. Naive Bayes is a probabilistic classifier that assumes independence among features, making it particularly suitable for our task given the diverse set of building attributes we are considering.

In constructing the Naive Bayes model, we utilized the Gaussian Naive Bayes implementation provided by scikit-learn. The model was trained using 10-fold cross-validation, a widely used technique for estimating the performance of machine learning models.

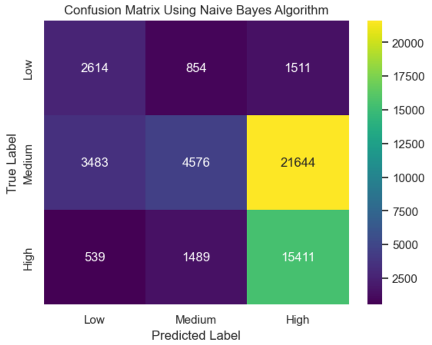

Figure 2: Confusion Matrix for Naive Bayes

The initial results of our experimentation revealed an average training F1 score (Micro) of approximately **0.431**, indicating moderate performance during the training phase. Upon evaluation on the test dataset, the Naive Bayes model achieved an average F1 score (Micro) of around **0.434**, reflecting similar performance to the training phase.

Analyzing the confusion matrix (Figure 2) generated during testing provides insights into the model's performance across different damage grades:

- The model exhibits higher precision in predicting damage grades 1 and 3, with precision scores of **0.39** and **0.40** respectively.
- It struggles with grade 2 predictions, where the precision score is notably higher at **0.66** but is accompanied by a low recall of **0.15**, indicating a higher rate of false negatives for this class.

5.3 K-Nearest Neighbors (KNN)

The K-Nearest Neighbors (KNN) model was utilized to assess the damage grade of buildings after the Gorkha earthquake. The model setup included data preprocessing, where categorical data was transformed using one-hot encoding and numerical data was scaled. Notable features were selected through univariate analysis with the Chi-squared test. The KNN model's parameters were default except for adjustments during the modeling process.

A 10-fold cross-validation was employed to validate the model's performance, providing insights into its generalization across various data segments. The KNN model achieved an average F1 score (micro-averaged) of 0.656 during cross-validation, suggesting moderate effectiveness in balancing precision and recall across classes. Further testing was conducted on a held-out portion of the dataset. The performance metrics from this phase were detailed in a confusion matrix (Figure 3) and included precision, recall, and F1 scores. Key observations from the test phase include:

- The model was moderately successful at classifying medium damage cases but showed some challenges in accurately predicting low and high damage scenarios.
- It tended to misclassify low-damage cases as medium or high, similar to the Random Forest model but with less accuracy.

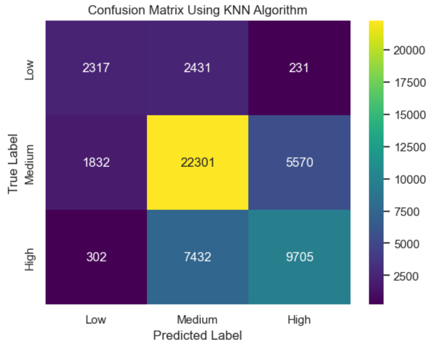

Figure 3: Confusion Matrix for KNN

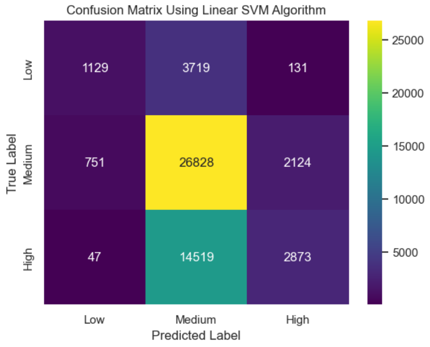

Figure 4: Confusion Matrix for SVM

- High-damage cases were occasionally underestimated as medium damage, which could be critical in practical disaster response situations.

Additional metrics calculated post-testing further illustrated the model's performance:

- Average Test F1 Score (Micro): **0.659**
- Average Test Precision Score (Micro): Values not directly provided but reflected in classification report scores.
- Average Test Recall Score (Micro): Also reflected in the classification report, highlighting the model's ability to correctly identify true positives across damage levels.

Despite some challenges with specific damage categories, the consistency in the KNN model's performance across different evaluation metrics underscores its utility in scenarios requiring quick and reasonably accurate classification, such as preliminary assessments in disaster response. However, its limitations in handling high-dimensional data and noise suggest that more complex models or additional feature engineering might be necessary for improved accuracy.

5.4 Support Vector Machine (SVM)

The Support Vector Machine (SVM) model was implemented using the SVC class from scikit-learn, which is an implementation of the linear Support Vector Classification algorithm. Hyperparameter tuning was performed using GridSearchCV to find the optimal values for the regularization parameter C and kernel. The dataset was split into training and test sets, with 80% of the data used for training and 20% for testing.

The C parameter, also known as the penalty parameter, controls the trade-off between achieving a good fit to the training data and minimizing the misclassification of training examples. Higher values of C impose a larger penalty for misclassified instances, prioritizing minimizing the misclassification errors, while lower values of C prioritize a simpler decision boundary, potentially leading to increased bias but reduced variance. The kernel parameter defines the underlying kernel function used to map the data into a higher-dimensional space, enabling the SVM to find a separating hyperplane that maximizes the margin between classes. The linear kernel assumes the data is linearly separable in the feature space, while the radial basis function (RBF) kernel, also known as the Gaussian kernel, can handle non-linear relationships by mapping the data into an infinite-dimensional space, but at the cost of increased computational complexity.

The GridSearchCV explored a range of values for C and gamma (used for the RBF kernel), as well as the linear and RBF kernels, to find the optimal combination of hyperparameters that maximized the cross-validation score, typically accuracy or F1-score. The final SVM model was trained on the entire training set using the best hyperparameters found during the grid search.

The GridSearchCV explored the following parameter values:

- C: [0.1, 1, 10, 100]
- gamma: [0.001, 0.01, 0.1, 1] (used for the RBF kernel)
- kernel: ['linear', 'rbf']

After finding the best hyperparameters through the grid search, the final SVM model was trained on the training set.

The model's performance was evaluated on the test set, and the following metrics were computed:

- Average F1 Score (Micro): **0.592**
- Average Precision Score (Micro): **0.592**
- Average Recall Score (Micro): **0.592**

While the hyperparameter tuning process aimed to find the optimal settings, the SVM model's performance on this dataset appears to be relatively poor compared to other models mentioned. The micro-averaged F1-score, precision, and recall of 0.592 are lower than the scores achieved by the Random Forest and XGB models, indicating poorer overall performance in accurately classifying the instances across all classes.

The reasons for the suboptimal performance could include the complexity of the problem, the presence of non-linear relationships in the data that the linear or RBF kernels may not have captured effectively, or the need for further feature engineering or data preprocessing steps.

5.5 Random Forest Classifier

The Random Forest Classifier (RFC) model was implemented with key parameters including ‘n estimators‘ set to 100, providing a robust ensemble of decision trees to make predictions by averaging

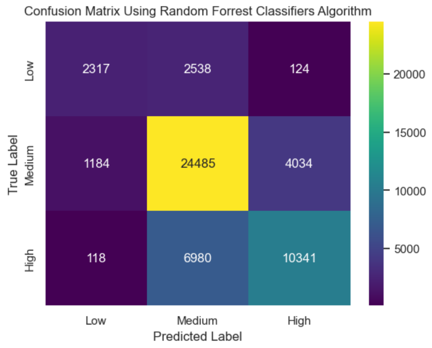

Figure 5: Confusion Matrix for RFC

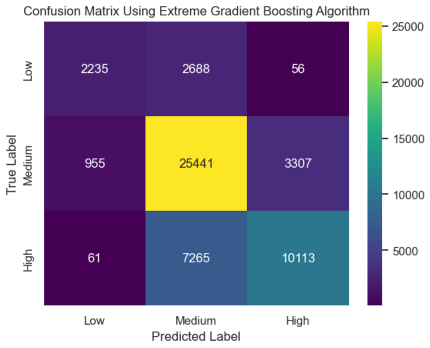

Figure 6: Confusion Matrix for XGB

their outputs. No specific depth restriction ('max depth') was defined, allowing the trees to expand deeply enough until the nodes are pure or contain less than the minimum sample split. This model was also trained using a 10-fold cross-validation approach, ensuring that the model's assessment was thorough and generalized well across different subsets of the data.

The initial results from the cross-validation process for the Random Forest model produced an average micro-averaged F1 score of approximately 0.715. This score reflects a solid baseline performance, indicating a good balance between precision and recall across the model's predictions for the multiple classes. Post-training, the model was utilized to make predictions on a test dataset, and the performance was again evaluated using a confusion matrix along with precision, recall, and F1 score metrics. The confusion matrix shown in Figure 5 provided the following insights:

- The model was quite effective at correctly predicting medium damage cases, which had the highest number of accurate predictions.
- Similar to the XGB model, the Random Forest classifier tended to overpredict damage when assessing low-damage scenarios, often classifying them as medium or high damage.
- High damage cases were sometimes underestimated as medium damage, posing potential risks in scenarios where accurate and timely damage assessment is crucial.

Additional performance metrics were computed to provide a holistic view of the classifier's effectiveness:

- Average Test F1 Score (Micro): **0.713**
- Average Test Precision Score (Micro): **0.713**
- Average Test Recall Score (Micro): **0.713**

These scores underscore the model's consistent performance across the different evaluative dimensions, ensuring that it neither overly penalizes nor favors any particular class. The consistency observed in the Random Forest model's ability to identify damage levels across different scenarios makes it a valuable tool for applications requiring reliable classification, such as in environmental impact studies or infrastructure damage assessments post-natural disasters.

5.6 Extreme Gradient Boosting

The Extreme Gradient Boosting (XGB) model was configured with default parameters for initial experimentation. This model was trained using a 10-fold cross-validation, ensuring the data was shuffled and split into 10 different subsets to validate the model's performance iteratively, which mitigates the risk of overfitting and ensures a robust evaluation across the training dataset.

The initial results from the cross-validation yielded an average micro-averaged F1 score of approximately 0.724, indicating a strong performance in balancing precision and recall across the multiple classes of the dataset. After training, the model was used to predict the classes of a separate test dataset.

The results were quantified using a confusion matrix, which detailed the number of true positive, false positive, false negative, and true negative predictions across three classes (low, medium, and high damage). The confusion matrix shown in Figure 6 indicated:

- Correct predictions for medium damage were most frequent, suggesting good model sensitivity and specificity for this class.
- Low damage cases were often overestimated as medium or high, indicating a potential bias or a lack of distinctive features aiding in the low damage detection.
- High damage was occasionally underestimated as a medium, which could have severe implications in practical scenarios where accurate damage assessment is critical.

Further metrics were calculated to provide a comprehensive view of the model's performance:

- Average Test F1 Score (Micro): **0.725**
- Average Test Precision Score (Micro): **0.725**
- Average Test Recall Score (Micro): **0.725**

These scores suggest that the model performs consistently across different evaluation metrics, offering a balanced identification of each class without significant bias toward any particular class. This consistency is crucial for practical applications where misclassification can have different implications based on the context, such as disaster response planning and damage assessment.

5.7 Final Model

As seen above, RFC & XGB are the best-performing classifiers for this dataset. We will continue to optimize the performance of these

Evaluation Metrics			
Model	F-1 Score	Precision	Recall
Naive Bayes	0.433625	0.433625	0.433625
SVM	0.591508	0.591508	0.591508
KNN	0.658525	0.658525	0.658525
Neural Net	0.658549	0.658549	0.658549
Random Forest	0.712630	0.712630	0.712630
XGB	0.725024	0.725024	0.725024

Table 1: Model Performance Summary

models. A systematic approach was employed using a Randomized Search CV. This method involves searching through a predefined grid of hyperparameters, selecting random combinations to train the model, and validating the results using cross-validation. This approach not only explores the hyperparameter space more efficiently but also avoids the exhaustive computation of Grid Search.

The best parameters obtained from the Randomized Search for the XGB model were n estimators set to 200 and max depth set to 40. The model with these optimized parameters achieved a best cross-validated F1 micro score of approximately 0.742, reflecting an improvement over the baseline model. This optimized model likely benefits from a balanced complexity that adequately captures the underlying patterns in the data without fitting excessively to the training data.

The optimal combination for the RFC model discovered through Randomized Search was n estimators at 200, max depth at 40, the criterion to Gini, and max features set to Sqrt. This configuration resulted in a best cross-validated F1 micro score of approximately 0.734.

Armed with these parameters, both models were deployed to predict the classifications on a set of unlabeled test data. The predictions were then submitted to the competition. Our submissions achieved impressive results:

- RFC Model: Achieved a micro-averaged F1 score of **0.725**.
- XGB Model: Outperformed with a micro-averaged F1 score of **0.7424**.

6 DISCUSSION & FUTURE WORK

The accuracy scores positioned us within the top 700 entries of the competition, approaching the highest recorded score of 0.7558. This performance not only demonstrates the efficacy of our models but also underscores the success of our systematic approach to model tuning and validation within a competitive framework. This achievement highlights our models' capabilities in robustly handling and making predictions on complex datasets, marking a significant accomplishment in the competitive arena.

Upon reviewing higher-ranked submissions, we noted a significant trend: the integration of neural networks for feature engineering, particularly with geospatial features, seemed to considerably enhance model performance. These models employed techniques such as dimensionality reduction on geo-level identifiers to extract dense embeddings, effectively capturing the underlying spatial relationships and patterns not immediately apparent with the raw data.

Specifically, these approaches involved:

- Geo Dimensionality Reduction: Using neural networks to perform dimensionality reduction on Geo Level IDs, transforming these categorical inputs into dense embeddings that succinctly capture spatial hierarchies and relationships.
- Geo 3 Rollup: Implementing a supervised learning framework where neural networks predict higher-level geo identifiers (Geo Level 1 and 2) from the lowest level (Geo Level 3). This not only aids in embedding generation but also ensures that the embeddings are meaningful and predictive of spatial groupings.

In retrospect, employing a similar strategy of generating enhanced feature representations through neural networks could have improved our model's performance. An ensemble approach, leveraging the strength of neural networks for feature extraction combined with robust classifiers such as Random Forest or Extreme Gradient Boosting, appears promising. This hybrid method would capitalize on the neural networks' capability to abstract and encapsulate complex patterns in new features while utilizing the predictive power and stability of ensemble tree-based methods.

7 CONCLUSION

In our work, we successfully applied a range of machine learning models to predict building damage from the 2015 Gorkha earthquake, with our approach ranking within the top 700 in the competition. Our evaluation highlighted the effectiveness of ensemble methods like Random Forest and XGBoost, which provided the highest accuracy among the tested models. Insights from higher-ranked submissions indicate significant potential in utilizing neural networks for generating new features from geo-spatial data, suggesting a direction for future work. Moving forward, we plan to focus on integrating advanced neural network techniques for feature extraction and exploring a hybrid modeling approach that combines these features with robust classification algorithms. This strategy is aimed at enhancing the model's predictive capabilities and generalizability across various seismic scenarios, ultimately contributing to better disaster preparedness and infrastructure resilience.

REFERENCES

[1] Sujith Mangalathu, Han Sun, Chukwuebuka C Nweke, Zhengxiang Yi, and Henry V Burton. Classifying earthquake damage to buildings using machine learning. *Earthquake Spectra*, 36(1):183–208, 2020.

[2] Samuel Roeslin, Quincy Ma, Hugon Juárez-Garcia, Alonso Gómez-Bernal, Joerg Wicker, and Liam Wotherspoon. A machine learning damage prediction model for the 2017 puebla-morelos, mexico, earthquake. *Earthquake Spectra*, 36(2_suppl):314–339, 2020.

[3] Kuldeep Chaurasia, Samiksha Kanse, Aishwarya Yewale, Vivek Kumar Singh, Bhavnish Sharma, and BR Dattu. Predicting damage to buildings caused by earthquakes using machine learning techniques. In *2019 IEEE 9th International Conference on Advanced Computing (IACC)*, pages 81–86. IEEE, 2019.

[4] Vasileios Linardos, Maria Drakaki, Panagiotis Tzionas, and Yannis L Karnavas. Machine learning in disaster management: recent developments in methods and applications. *Machine Learning and Knowledge Extraction*, 4(2), 2022.

[5] Sujith Mangalathu and Jong-Su Jeon. Regional seismic risk assessment of infrastructure systems through machine learning: Active learning approach. *Journal of Structural Engineering*, 146(12):04020269, 2020.

[6] Sujith Mangalathu and Henry V Burton. Deep learning-based classification of earthquake-impacted buildings using textual damage descriptions. *International Journal of Disaster Risk Reduction*, 36:101111, 2019.

[7] Enes Celik, Muhammet Atalay, and ADIL Kondiloglu. The earthquake magnitude prediction used seismic time series and machine learning methods. *Proc. ENTECH*, 12, 2016.

[8] Jhon Veri and The Ying Wah. Earthquake prediction based on the pattern of points seismic motion. In *2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*, pages 209–212. IEEE, 2012.