Predicting Damage Based on 2015 Gorkha Earthquake Dataset

Team Members: Shaun Guyette, William Lewis, Nicholas Stone
Advisor: Dr. William Eberle

Problem Statement

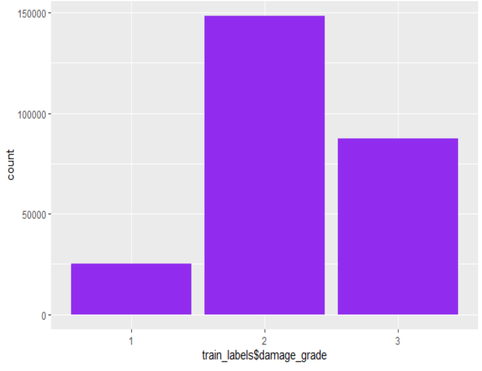
On the 25th of April 2015, a magnitude 8 earthquake killed nearly 9,000 people and injured nearly 22,000 in the Gorkha District of Nepal. These harrowing statistics show the importance of proper preparation for disasters such as this. One way communities can be prepared is to be able to predict the amount of damage different buildings would suffer in the event of an earthquake.

To this end we set out to study a dataset containing data for over 260,000 buildings damaged in the earthquake in order to predict the variable **damage_grade**. This attribute is an ordinal variable which can take on values 1, 2, or 3 which represent increasing damage to buildings. The dataset includes more than 30 different properties of the buildings such as age, size, and construction type. Our goal was to create a predictive model based on the training set in order to accurately predict damage_grade.

The dataset was taken from a data science competition on drivendata.org. The site for the competition states that our metric for success should be the micro-f1 score of the model. For this reason, we will be using micro-f1 score to compare models and to determine the success of our models.

Methods: Data Preparation

- We checked the dataset for outliers and invalid data. We also cleaned the dataset to make processing easier.
- Because the test dataset did not include damage_grade, we could not validate our models with that set. We split the training set into train and test to build our models.
- The majority of the items in the dataset have a damage_grade = 2. As a result, we used random sampling to guarantee relative ratios in the training and test set. We used a ratio of 9:1 training items to testing items.



Methods: Evaluation and Models

- Our main evaluation metrics is the micro-f1 score. See figure below.

$$F_{micro} = \frac{2 \cdot P_{micro} \cdot R_{micro}}{P_{micro} + R_{micro}}$$

$$P_{micro} = \frac{\sum_{k=1}^{3} TP_k}{\sum_{k=1}^{3} (TP_k + FP_k)}$$

$$R_{micro} = \frac{\sum_{k=1}^{3} TP_k}{\sum_{k=1}^{3} (TP_k + FN_k)}$$

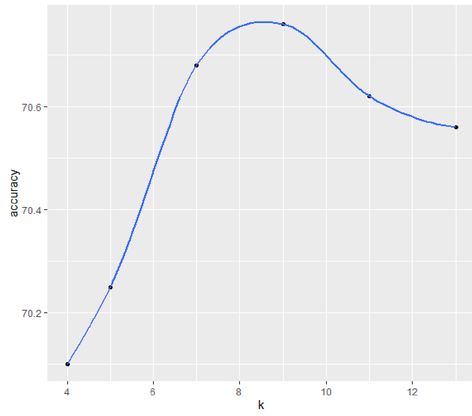
- In multiclass classification problems, the micro-f1 score is equivalent to the accuracy of the model, or simply true positives divided by total predictions.
- Our approach was to try different classification methods and compare the results to determine the best model for the task.
- The first model we tried was kNN, or k-Nearest Neighbors algorithm. Various configuration options and feature selections were tested in order to optimize our micro-f1 score.
- Random forest was another algorithm we used. We experimented with different features to obtain the best micro-f1 score we could get.
- We determined our set of features from regression analysis. These attributes included the number of floors before the earthquake, the number of families living in the building, the area percentile, and the presence of certain features in the superstructure. Geo_level_id attributes (similar to ZIP codes) were included as tie breakers for kNN.

Tools

- RStudio was our main tool used for creating models and analyzing data.
- Python was briefly used to prepare the data.
- PowerPoint was utilized to create this poster.
- Teams was used to communicate and share data and code.

Results

- K-Nearest Neighbors:
  - Through a series of trials, we determined that a small k value was better at classifying damage_grade. Our most accurate models used k values between 5 and 11. Use.all=TRUE was used in our models.
  - Various feature selections were tested for optimal micro-f1 score. The results below were computed with the set of features that produced the greatest accuracy.
  - Utilizing kNN, we were able to obtain a micro-f1 score of around 70%.



- Random Forest:
  - We used similar training and testing feature sets to those used with kNN.
  - Our random forest models were created using 300 trees.
  - Utilizing random forest, we were able to obtain a micro-f1 score of around 72.60%.

```
pred1     1     2     3
    1  1101   465    33
    2  1342 12606  3592
    3    31  1662  5168
[1] 0.7259615
```

  - While random forest outperformed kNN, there may be other options to explore that increase accuracy for either method.

Discussion/Conclusions

With the methods we used, the greatest accuracy we were able to obtain was around 72.6%. Our findings indicate that the number of floors, the superstructure, the number of families, the area percentage, and the foundation type were good predictors of damage_grade. We found that kNN performs decently at predicting damage_grade but could be better. Random forest had the greatest accuracy of the methods we used.

The models were able to best identify entries with damage_grade = 2. The reason for this tendency is most likely due to the amount of those buildings in the dataset.

We will continue investigating different ways to increase the accuracy of our models. We will try alternate selections of features to better fit the data. For random forest, we will try varying amounts of trees. Another idea is to modify the age value for entries with large age values. We will investigate introducing noise into our discrete values in order to possibly remove geo_level_id atrributes from our features.

Additionally, we will attempt other algorithms and models to better fit our problem statement. Our future efforts will focus on support vector machines as well as ordinal logistic regression as potential solutions.

References

- de Oliveira, Saulo Pires. "A Very Basic Introduction to Random Forests Using R." *blopig.com,* https://www.blopig.com/blog/2017/04/a-very-basic-introduction-to-random-forests-using-r/. Accessed 11 April 2021.
- DrivenData. "Richter's Predictor: Modeling Earthquake Damage." *Drivendata.org*, https://www.drivendata.org/competitions/57/nepal-earthquake/ Accessed 11 April 2021.
- Shah, Chirag. *A Hands-On Introduction to Data Science.* Cambridge University Press, 2020.
- Sirohi, Kshitiz. "K-nearest Neighbors Algorithm with Examples in R." *Towards Data Science,* https://towardsdatascience.com/k-nearest-neighbors-algorithm-with-examples-in-r-simply-explained-knn-1f2c88da405c. Accessed 11 April 2021.

Acknowledgments: Dr. Eberle