Modeling Earthquake Damage

ORIE 4741 Project Final Report

Authors: Vladia Trinh (vt95), Yoanna Efimova (yie3), Toshi Tokuyama (tt426)

Table of Contents

Abstract

The Gorkha earthquake in Nepal has been one of the most severe earthquakes for the past few years. On April 25, 2015, the earthquake took around 9,000 lives and destroyed around 600,000 buildings in the city of Kathmandu and other nearby towns. Following the earthquake, the land was surveyed to see how much damage each building incurred. The "*Richter's Predictor: Modeling Earthquake Damage*" dataset consists of 260,601 observations with 39 different features. In this paper we examine the impact of those 39 factors on the damage grade caused by the Gorkha earthquake on buildings in Nepal. After performing exploratory data analysis and cleaning the data, we develop different models that predict the damage grade on buildings based on the provided features. We explore models such as ordinal regression, multinomial regression, decision trees, and random forest.

1. Introduction

The dataset we are exploring is called "*Richter's Predictor: Modeling Earthquake Damage*", provided by the Driven Data website. [1] It is collected through surveys by Kathmandu Living Labs and the Central Bureau of Statistics, which works under the National Planning Commission Secretariat of Nepal. The dataset provides information on buildings in the region that was hit by the Gorkha earthquake. There are a total of 260,601 observations and 39 features, including building id, number of floors, age of the building, area and height of the building footprint, surface condition of the land, the position of the building, and others. [2] The features in our dataset include numerical, categorical, and binary values.

The purpose of this project is to predict the damage grade caused by the earthquake to various buildings in Nepal. The variable for damage grade has 3 categories labels as 1, 2, and 3: 1 represents low damage, 2 represents a medium amount of damage, and 3 represents almost complete destruction. To accomplish this, we first preprocess our data to make fitting a model possible. This includes handling missing values, standardizing features on a 0 to 1 scale, and converting categorical features to binary using one-hot-encoding. Next, we do some exploratory data analysis to see the correlation between the damage grade and our other variables to get a rough ranking of influential features. In addition, we create some plots to see the distribution of specific features. Lastly, we explore ordinal and multinomial regression models without regularizer and with l1 and l2 regularizers. We also fit a decision tree and a random forest and use hyperparameter tuning to find the best parameters. To summarize, we want to answer the question "Can we predict which buildings in Nepal were most susceptible to earthquake damage, and which features can help us best accomplish this?".

2. Data Cleaning & Preprocessing of Data

First, we had to check if the dataset was in the appropriate format for future analysis. There were no missing or NA values in the dataset. We also checked the variable types to ensure that Python and R would interpret them correctly. In doing so, we converted the categorical variables (*land_surface_condition, foundation_type, roof_type, ground_floor_type, other_floor_type, position, plan_configuration, legal_ownership_status*) to binary variables using one-hot encoding. We created a binary column for each level of the categorical variables. In addition, we converted the variables (*area_percentage, height_percentage*) that had percentages as values to decimal numbers from 0 to 1.

To make sure that each feature is properly weighted equally, we did some standardizing. For example, for *count_floors* and *age*, we used min-max scaling, which means that the minimum value in our dataset is 0 and the maximum is 1, and all values in between are scaled accordingly. We may want to experiment with a different type of scaling later on since the distribution of *age* is not very uniform. Next, we checked for variables that did not have an impact on the damage grade of each building. We removed the variable *building_id* because it was clear that it was not related to the *damage_grade* variable. After removing that variable, we were ready to continue on with our preprocessing.

3. Exploratory Data Analysis

In addition to predicting the damage grade of each building, we wanted to understand how each attribute relates to the earthquake damage on the building. We wanted to see if there was any relationship between the damage grade and buildings' area, age, roof, foundation, and floor type, the position of the building, etc.

To start, we created a correlation plot for the variables to get a better sense of the correlations between each pair of features and to find which variables had the highest influence on the *damage_grade* variable. By looking at the heatmap (Fig. 1 in Appendix), we were able to notice that the dependent variable was highly correlated to the variables *area_percentage*, *has_superstructure_cement_mortar_stone*, *has_superstructure_cement_mortar_brick*, *has_superstructure_rc_non_engineered*, *has_superstructure_rc_engineered*. The *area_percentage* variable refers to the normalized area of the building footprint. Some of the other variables indicate if the superstructure was made of cement mortar – stone or cement mortar – brick and whether it was made of non-engineered or engineered reinforced concrete. The *has_secondary_use* variable is a flag variable that indicates if the building was used for any secondary purpose such as a hotel (*has_secondary_use_hotel*) and/or rental (*has_secondary_use_rental*).

From our preliminary correlation analysis of *damage_grade* with our other features, we noticed that *count_floors* was positively correlated and *area_percentage* was negatively correlated with *damage_grade*. Figure 1 represents a scatter plot of the two variables including a dimension for *damage_grade*. From the graph, these correlations do not look too significant, but there seems to be a higher *damage_grade* centered around x = 0.25 and a lower *damage_grade* as *area_percentage* increases.

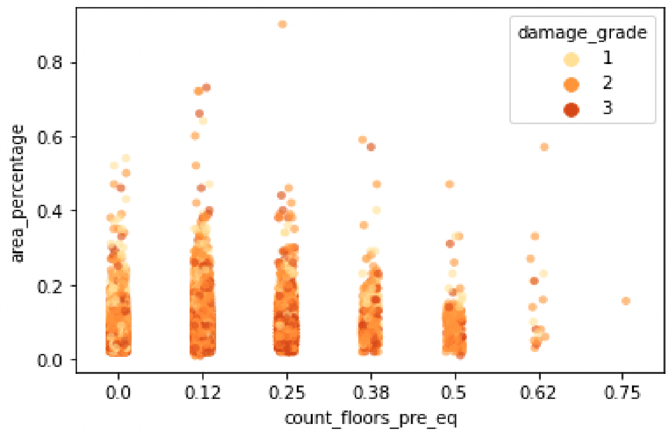


Figure 1. Scatter plot of area_percentage and count_floors_pre_eq with damage_grade

Figure 2 examines the relationship between the buildings' roof type and the damage grade. By looking at the graph, we were able to see that the most stable roof type was type x. The most unstable one was roof type 'n' with more than 100,000 buildings of damage grade 2 and more than 60,000 buildings of damage grade 3. Moreover, we could see that there were also many buildings with roof type q that had more severe damages of grades 2 and 3.

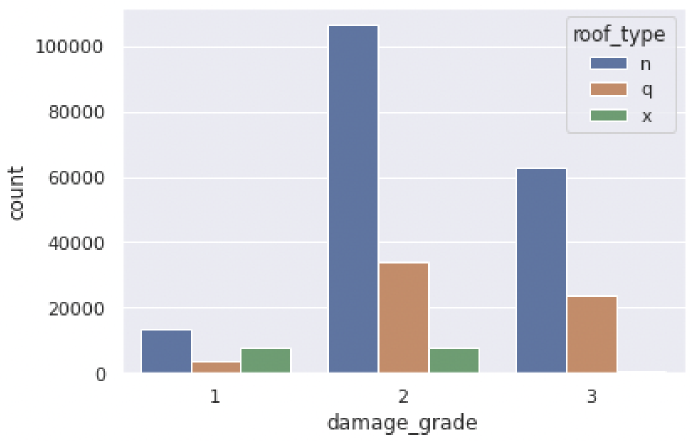


Figure 2. Histogram of damage_grade According to roof_type

Figure 3 shows histograms for three variables: age, count_floors_pre_eq, and height_percentage. The histogram shows that our data is highly skewed; therefore, we should apply a transformation, so the variables become normally distributed. The histogram of age shows some buildings that are very old (greater than 100) but do not have a large count. Some buildings were aged 995, which are most likely historic buildings. We will need to consider whether to remove buildings that are aged 995 because the second oldest building is around 300 years old. The large difference between these values can add more complexity to our models. The histogram of count_floor_pre_eq suggests that the variable can be turned into a categorical variable. Since a house with 2 floors seems to be the most common, dummy coding can be used in which count_floors_pre_eq=2 is set as the base. Then if we have a linear model, we can compare the coefficients with the 2nd floor to make interpretations. Furthermore, the histogram of height_percentge shows that most of the height_percentage is between the range 2% ~ 13%. The data is highly skewed, so transformation should be applied. Overall, the histograms show that most of the damage grade is at the 2nd and 3rd level. Therefore, the challenge would be to predict houses that have damage_grade = 1.

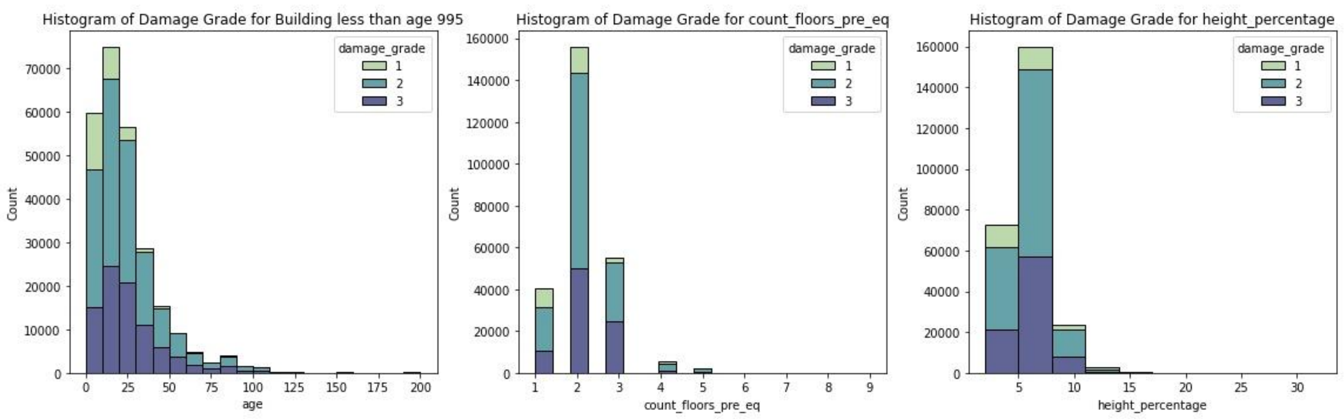


Figure 3. Histogram of age, count_floor_pre_eq, and height_percentage

To further explore some of the categorical variables, we focused on three variables: *land_surface_condition*, *foundation_type*, and *other_floor_type* with respect to the damage grade. From Figure 4, we can see trends that certain types of floors, foundations, and land surfaces are more susceptible to earthquakes than others. However, the plot does not provide a clear picture of the relationship between the damage grades. The ratios between the damage types are similar for each variable. After conducting a chi-squared test of association on all three variables against *damage_grade*, we can conclude from the test that there is association between the variables.

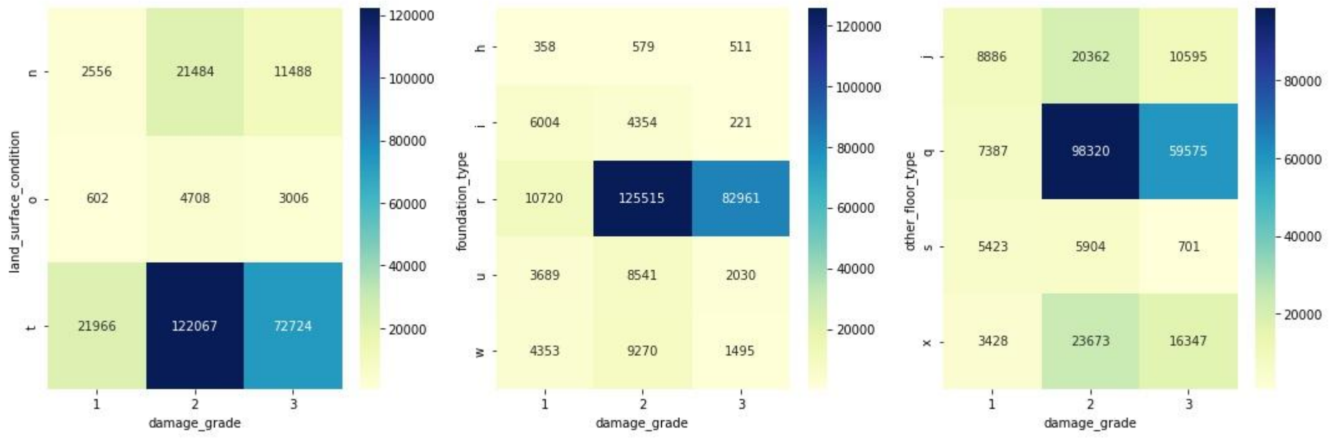


Figure 4. Crosstab of land_surface_condition, foundation_type, and other_floor_type according to damage_grade

4. Model Selection

4.1 Ordinal Regression

After exploring the dataset and gaining insights into it, we began working on building a simple regression model. For our initial model, we chose to use an ordinal regression model because the variable that we wanted to predict (*damage_grade*) is categorical with ordered categories: 1, 2, 3. [3] We first split the data into 75% training set and 25% test set and preprocessed the data as described previously. We only trained the model on the subset of features we found to be important during our exploratory data analysis to prevent overfitting and reduce computational complexity. Similarly to the logistic regression model, the ordinal regression model returns a probability (the probability that a building has a certain damage grade). In order to classify each building, we took the maximum probability among the three probabilities we got for each damage grade (1, 2, and 3). To measure the performance of the model, we used accuracy, which is just the number of correct predictions divided by all predictions. Our model performed similarly on both the training and test data, with an accuracy of around 57%. This is quite low, which is expected since we only used the default parameters of the python package.

In addition to that model, we decided to look at models with a regularizer. Instead of training these models on only a subset of features as before, we decided to train them on all features. We explored ordinal regression models with l1, l2, and elastic net penalty. We did this by using the *ordinalNet* package in R. [4] To use lasso regularization we had to set the argument *alpha* of the *ordinalNet* function to be equal to 1. Similarly, we had to set *alpha* to be equal to 0 for ridge regularization and *alpha* to be between 0 and 1 for elastic net regularization. Because we added many new variables during the one-hot encoding, we expected the l1 regularization to perform well because it shrinks some of the coefficients to 0 and encourages sparsity. However, the accuracy of the model did not improve. The accuracy we got for each model was still around 57-58%, which was too low.

To further analyze these models, we constructed a confusion matrix for each of them. By looking at figure 5, we were able to see that most of the buildings were predicted to be of damage grade 2. Part of the buildings with damage grade 1 were correctly predicted to be of damage grade 1. Some of the buildings of grade 2 were predicted to be of other damage grades (1 and 3). However, only a small part of the buildings that had an actual damage grade of 3 were predicted to be of grade 3. We believe that predicting a building to be of a higher damage grade won't be such a problem. The only difference would be that the building might undergo more substantial reconstruction. However, predicting a building to be of a lower damage grade than its actual grade would be a problem because it can have negative consequences on the nepali residents of the building.

4.2 Multinomial Logistic Regression

For our second model we decided to use multinomial logistic regression because the outcome variable is categorical and has more than 2 levels. We first split the data into a 60% training set, 20% validation set, and 20% testing set. Our goal was to use the validation set to find the optimal λ and coefficients that maximizes the accuracy score for the classification. The objective function of the multinomial logistic regression is defined as the following:

$$h(x) = \frac{1}{m}\sum_{i=1}^{m} log(1 + exp(-Y_i W_i X)) + \lambda||X||_2^2$$

We decided to use $l_2$regularization as to reduce multicollinearity as a lot of the columns included values that are related to house structures, and $l_2$regularization produces more optimal predictions because the coefficients are not reduced to 0. We decided not to create several binary logistic regression models because we believed that knowing the probability of being classified into either level 1, 2, or 3 for *damage_grade* would provide better insight. For example assume that a row was predicted to have level 1 has 30%, level as 50%, and level 3 as 20%. Then we would know that we should prioritize a row with level 1 as 20%, level as 50%, and level 3 as 30% as we know that repairing houses with higher chance of more damage is more important.

We use hyperparameter tuning to find the best parameters that would have the highest validation accuracy. We search through two parameters *C* and *solver*, which is used to control the regularization strength and the algorithm used in the optimization problem. After fitting each possible combination on 5 validation sets, we found that $C = 0.5$and the *saga solver* to find the best accuracy score. After finding the optimal hyperparameters we refit the model with the hyperparameters. The accuracy we got for the model was 74.1%.

4.3. Decision Tree & Random Forest

Next, we tried the DecisionTreeClassifier from the scikit-learn package. From the data exploration we've done above, there is strong reason to believe that our data is not linearly separable. Therefore, we decided to use a decision tree classifier to partition the feature space. This classification could potentially be more accurate since geographic regions of Nepal, which are a strong predictor of earthquake damage, could be partitioned accordingly using the classification function $\hat{f}(X) = \sum_{m=1}^{M} c_m 1(X \in R_m)$, where c_m corresponds to our label and R corresponds with M regions. With a max_depth of 25 to prevent overfitting and using the Gini Index as a measure to minimize impurity, we achieved an accuracy score of 0.63, with residual error of 162.15. We could have potentially improved this accuracy through hyperparameter tuning but decided to move on to an ensemble method to increase model complexity instead, using this model just as a baseline. Additionally, we determined which features had the most predictive power using the feature_importance_ attribute (Fig. 2 in Appendix). This allowed us to remove unimportant features to further prevent overfitting in our next model.

By experimenting with an ensemble method, we can lower the bias of our previous model without increasing variance too much, since we are just aggregating many weak learners. The ensemble method we chose is the RandomForestClassifier from scikit-learn, which fits multiple decision tree classifiers on various subsets of the training data and makes predictions based on the majority label. The most important hyper-parameters for this model are n_estimators, max_depth, max_features, and min_samples_leaf. Since only a subset of hyperparameters are important for this model and we wanted to survey a larger range of distinct values for each parameter, we opted for random search for a more efficient running time. To counteract overfitting, we used the RandomizedSearchCV which has built-in 5-fold validation in its hyperparameter sampling algorithm. The best parameters are as follows: {'n_estimators': 228, 'min_samples_leaf': 2, 'max_features': 'sqrt', 'max_depth': 65, 'bootstrap': True}.

After fitting the random forest classifier on training data with the best parameters from validation, we achieved training accuracy of 0.88 and testing accuracy of 0.72, with residual error of

140.04. As anticipated, the bias of our new bagged model is lower than the decision tree model. However, due to the hyper parameter tuning choosing the most accurate model on the training data, the random forest model does seem to overfit despite using bootstrap sampling and aggregation techniques.

5. Discussion

5.1 Weapons of Math Destruction

We believe that our model can potentially be a weapon of math destruction. Although the correctness of prediction is extremely important to save potential lives, families and individuals who live in an area where high earthquake damage is predicted may receive higher insurance rates. This can be a serious issue as the average income in Nepal is around 820 U.S dollars. Furthermore, a potential danger of our model is when we predict a damage grade of 1 when it is actually 3. This may actually prevent people from getting the help they need in a crisis. From analyzing the confusion matrices of our models (discussed further in the conclusion), we noted that this is one of the rarest misclassifications in our model, so overall, our model would do more good than harm in terms of classifying damage grade.

5.2 Fairness

Our group believes that fairness is always a crucial factor to be mindful of, but in terms of this project we did not have to actively employ any methods to prevent discrimination. From the attributes collected in the dataset there were no protected attributes that can be used to as a potential discriminatory feature. Although there were some features of location, since these were labeled as ids it is very difficult to locate the specific location given to each id. So, perhaps there was some locational data that is highly correlated with certain protected attributes, but due to the nature of the problem, removing them would greatly decrease the predictive power of our models. There is no way to ensure complete fairness in machine learning models, but recognizing possible biases in the model can help reduce the possibility of harm discussed in Section 5.1.

6. Conclusion

In this project we were able to explore different models in order to explain which factors had the greatest impact on the damage grade on buildings caused by earthquakes in Nepal. Overall, we achieved the following misclassification rates from our models:

Model	Misclassification Rate
Ordinal Regression	0.43
Multinomial Logistic Regression	0.26
Random Forest	0.28

Table 1. Misclassification Rates for Classification Models

From table 1, the ordinal regression model with and without regularizer performed the worst among all models we explored. Multinomial logistic regression and random forest performed significantly better. With these two models we were able to predict the damage grade of the buildings with 74% and 72% accuracy.

Figure 5 represents the confusion matrix of each classifier. To assess whether we are confident in our results and willing to use them in production, we would have to assess the number of misclassifications of the following cases: predicted = 1 & true = 3, predicted = 1 & true = 2, and predicted = 2 & true = 3. The intuition behind these cases is that we are willing to overestimate the potential damage, but not underestimate the damage since we would want to save as many lives as possible.

From the confusion matrix the cases in which predicted = 1 & true = 3 are minimal. The chances of our classifier predicted 1 when true is 3 is less than 1%. Similarly the chances of the classifier predicting 1 when true is 2 is around 1.3%. However, the chance of the case predicted = 2 & true = 3 occurring ranges from 12% to 20% depending on the model. Considering that having misclassified a large number of these cases can have a severe impact on nepali citizens, our group believes that the misclassification rate for this specific case is very high. Therefore, further work needs to be performed to make the models more informative and valuable. With our current model our group suggests the government use the model, but with the help of experts who have an abundance of knowledge regarding Nepali geography and architecture to mitigate the danger of the case specified above.

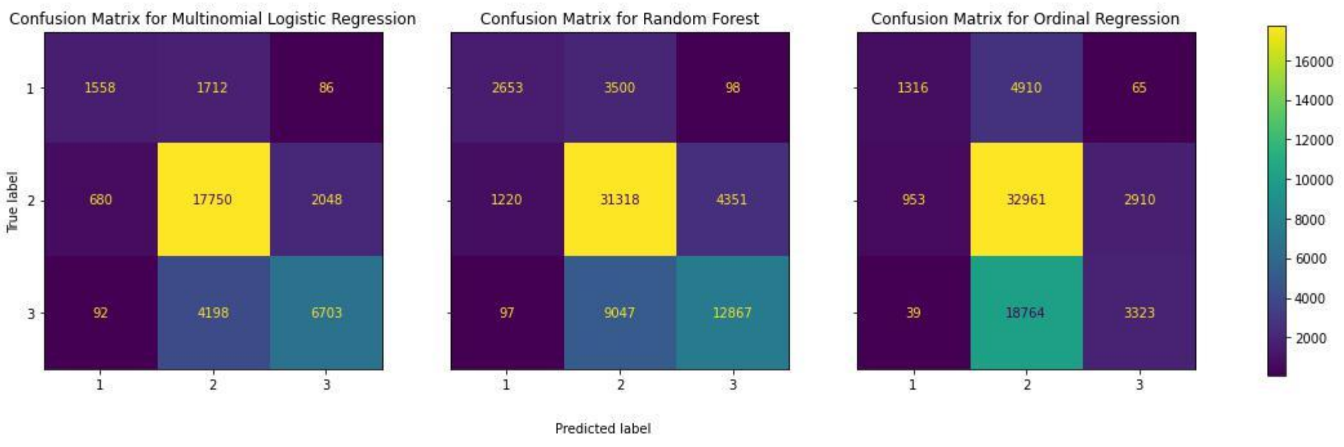


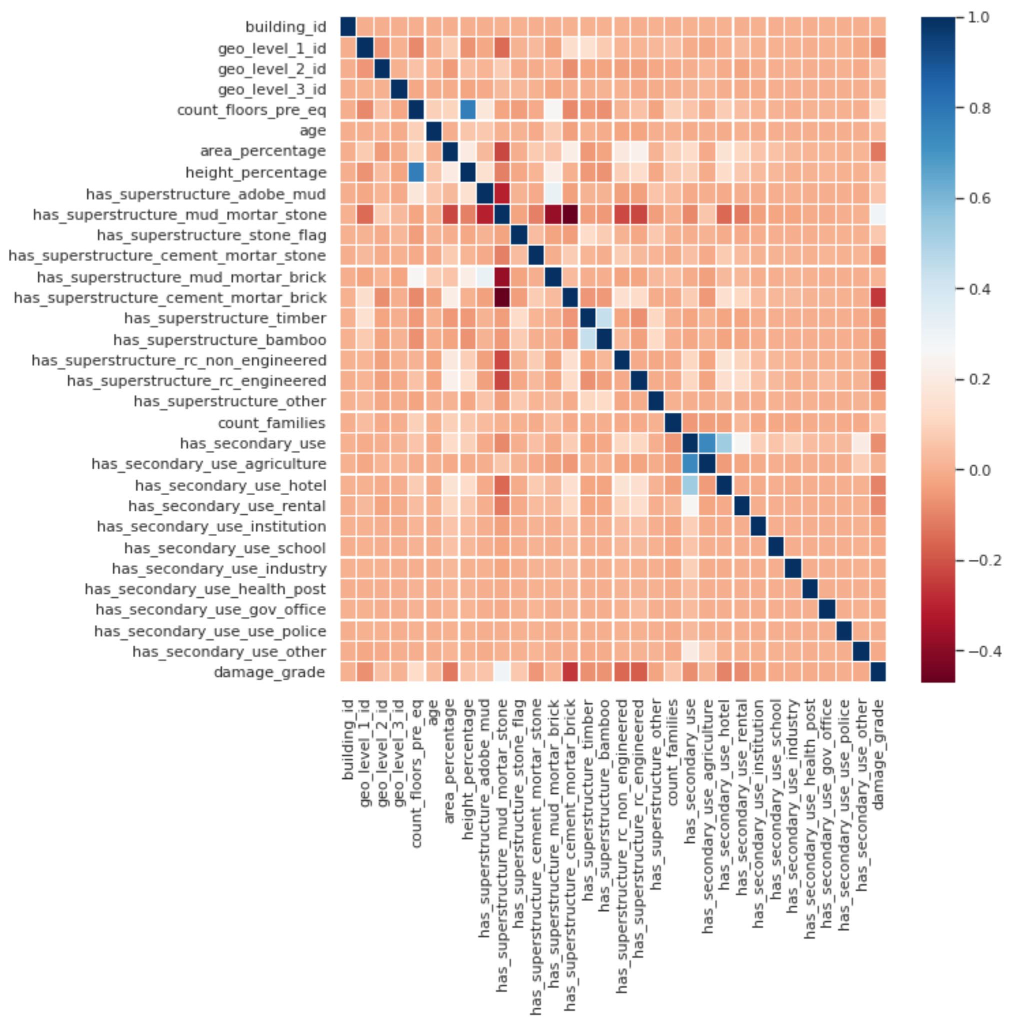
Figure 5. Confusion Matrices

In conclusion, we would say that the results from our models can be used in adequately identifying which buildings are eligible for government assistance and which buildings will benefit from house reconstructions. Our models can be used as a tool for predicting damage grade, but should not be the sole decider in insurance claims or building quality decisions.

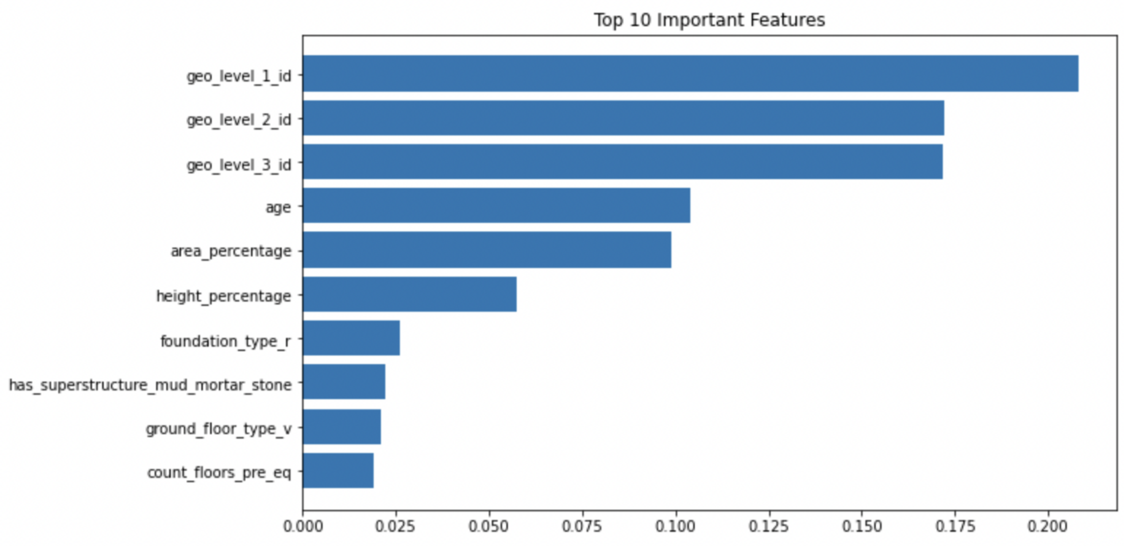
For future work we can possibly consider creating classifiers for some specific building damages such as binary classifiers for damage_grade = 2 and damage_grade = 3. Such a model will allow more specific predictions and we will be able to understand the cause for misclassification better. Another possible improvement if we had additional resources, would be to collect damage data from other similar regions around the world. This would allow us to create a model that generalizes better to other regions and predict damage of earthquakes in areas other than Nepal.

7. Appendix

[Figure 1]



[Figure 2]



8. References

[1] “Richter's Predictor: Modeling Earthquake Damage”
https://www.drivendata.org/competitions/57/nepal-earthquake/page/134/
[2] Features descriptions, “Modeling Earthquake Damage”
https://www.drivendata.org/competitions/57/nepal-earthquake/page/136/
[3] Ordinal Regression Documentation
https://www.statsmodels.org/dev/examples/notebooks/generated/ordinal_regression.html
[4] OrdinalNet package in R
https://rdrr.io/cran/ordinalNet/man/ordinalNet.html