Predicting Earthquake Magnitude

Stavros Cherpelis, sgc76
Vincent Moscarelli, vsm29
5/10/24

ORIE 4741 Project: Data Analysis Project
https://github.com/StavrosCherpelis/ORIE_4741_Project/tree/main

Predicting Earthquake Magnitude

Abstract:

The National Earthquake Information Center catalog provides requestable data on earthquakes in any region over any time period dating back to 1900. For this project, a dataset of California earthquakes from 2022 were specifically requested and combined. We will use this dataset to explore if we can build an accurate model that can predict earthquake magnitude. It is critical that we improve our understanding of earthquakes and their effects in order to better prevent injuries and fatalities caused by these events. In context with surroundings, earthquake magnitude can be used to qualitatively determine the potential damages and range of the earthquake. Being able to quickly discern earthquake magnitude is critical for government institutions to coordinate effective and efficient disaster relief. In our attempt to solve this problem we created a number of models including OLS regression, polynomial regression, and ridge regression. Ultimately we found that ridge regression provided the best model we could find for predicting earthquake magnitude

1. Data Analysis

1.1 Description

The dataset consisted of a list of earthquakes across California dated from June 15, 2022 to December 31, 2022. It was published by the National Earthquake Information Center and contained 19,674 entries with 22 features. The dataset is largely complete, only missing a few of the “error” variables on occasion, and rarely a “dmin” variable. Among the 22 variables, there are 14 continuous variables, 6 nominal, 2 time series. These variables tell us information like location, the seismic networks in the area of the earthquake, the particular magnitude scale used, the magnitude source, and information about things like earthquake depth. Most features in this dataset are known to be correlated with earthquake magnitude so we are confident that we can fit at least a decent model to this dataset.

1.2 Outliers / Data Distribution

First we wanted to test our dataset for outliers and inconsistencies, in order to do this we created boxplots for our continuous variables, and bar charts for our categorical variables.

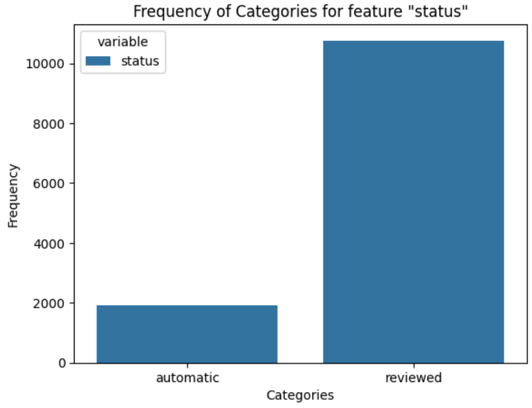

Figure 1 Status Frequencies

All of our bar charts covering nominal data showed a fairly equal distribution, with the exception of the variable "status". As depicted in **Figure 1**, the "reviewed" category for the feature was over five times as frequent as "automatic". The variable itself details whether the data has been reviewed by a human to any extent or was automatically generated and yet to be verified. This reviewal status comes with a clear correlation: bigger earthquakes draw more attention, and are significantly more likely to be reviewed. Additionally, the extremely high frequency of low-magnitude earthquakes makes it simply infeasible to review all of them. While skewed distributions are typically a cause for concern in this case the fact that so much data was verified by a human actually gives us more confidence in our dataset.

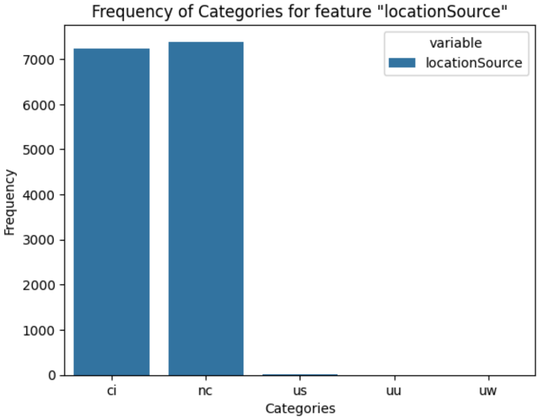

Figure 2: Frequency of Location Source

Some categorical variables like location source had categories with only a few (<50) data points while otherwise having data that was evenly distributed among the remaining categories. We simply removed these data points from our model since they made up an almost insignificant part of our dataset and occurred because of an anomaly with how data is pulled from the NEIC database.

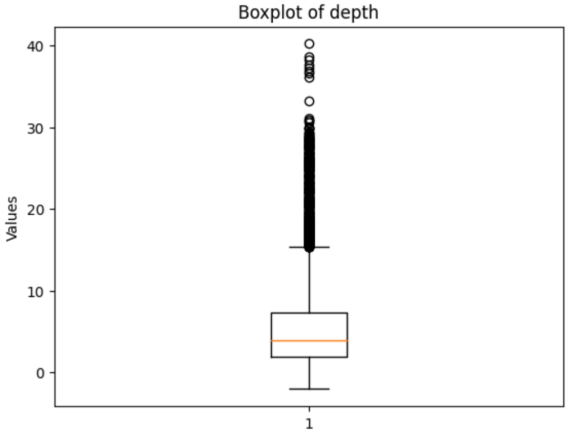

Figure 3 Depth Boxplot

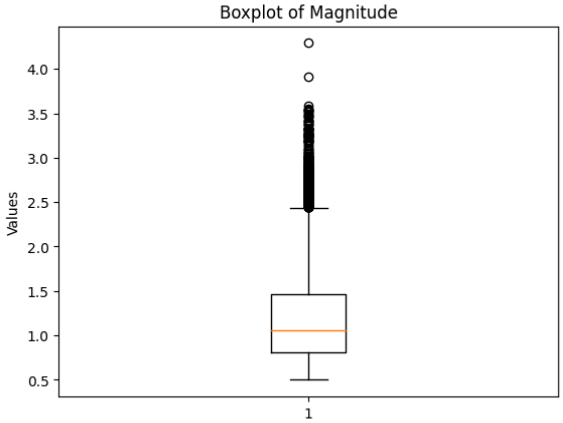

Figure 4: Magnitude Boxplot

By contrast, most of our box plots showed heavily skewed data. For example in **Figure 3**, we see that shallower depths, under 10 kilometers, were much more frequent than larger depths. This is because larger depths are associated with less frequent, larger scale earthquakes. Large earthquakes are rare while small magnitude earthquakes are more common, so large magnitudes and the data associated with them appear as outliers in their respective distributions.

1.3 Gutenberg-Richter Law

Before diving fully into our models, it is important to understand the notion of the Gutenberg-Richter law. The law shows that earthquake frequency correlates logarithmically with magnitude almost perfectly for all earthquakes with magnitude roughly higher than two. Once the low magnitudes are reached, seismometers struggle to read certain earthquakes due to their small scale, and some data is lost, leaving the lower magnitude earthquakes to have less records than they should.

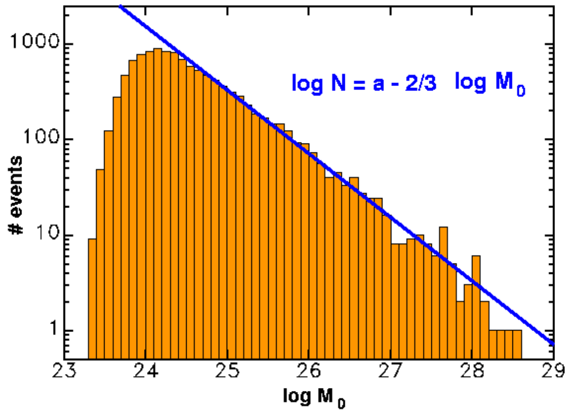

Figure 5 Gutenberg-Richter Law (via Lamont-Doherty Earthquake Observatory)

The effect this has on our models is significant. Our original dataset contained all earthquakes from magnitude zero upward. We found this dataset would lead to overfitting because the majority of the data were low-magnitude, fickle earthquakes, and would therefore prove to be somewhat lousy predictors. To remedy this, we attempted to move to a dataset of magnitudes 2.5 and up, but we found that this struggled; the outlier earthquakes that have very high magnitudes are always going to be hard to predict because there are so few of them due to the logarithmic scale. We finally compromised on a new "medium" dataset with 19,674 entries of magnitude 0.5 and up earthquakes, and found that this worked best for predicting the medium and high magnitudes that we were specifically hoping to predict more accurately. While we ultimately settled on this approach, an alternate approach might have been to train two models, one on lower magnitude earthquakes and another on higher magnitude earthquakes.

1.4 Multicollinearity

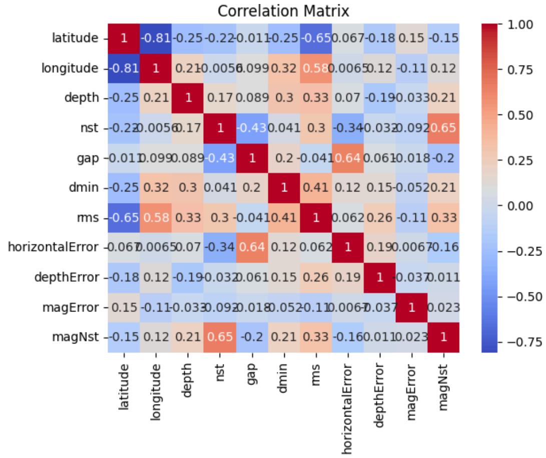

Figure 6: Correlation Matrix

Finally we wanted to analyze the multicollinearity of our continuous features, to do that we created a correlation matrix for those features. Most features were not highly correlated with each other but a few were. Notably, some features had small correlations with longitude and latitude which makes sense, since fault lines don't move significantly in the span of a year, and each fault is associated with certain types of earthquakes. This small amount of multicollinearity might prevent us from achieving a perfect model using linear regression and is something to be aware of.

2. Feature Engineering

Before fitting models to our data we need to address three main issues with our dataset, and we will use feature engineering to achieve this.

2.1 Nan Values / Missing Values

When dealing with Nan values we decided that it was best to simply drop and data with missing values, this was because we have a large dataset and will not be losing much information.

2.2 Feature Reduction

We also reduced the number of features of our dataset by 2. Since the place feature is redundant with the longitude and latitude features. In addition the time an earthquake occurs has no impact on magnitude so we removed it to simplify the model.

2.3 Dealing with Skewness

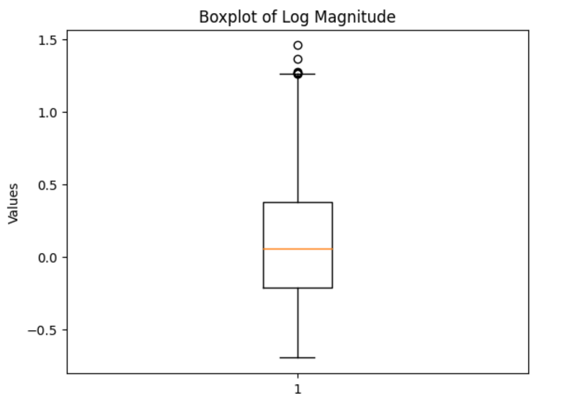

Figure 7: Log Magnitude

The second issue we had to deal with were the skewed values. Because of the Geutenberg-Richter law that earthquake frequency correlates logarithmically with magnitude, we knew we could address skewness by taking the logarithm of both our skewed features and our target variable. This should add more stability to our model by decreasing model variance. Comparing **Figure 4** and **Figure 7** shows the before and after effects of taking the log of our target variable.

2.4 One Hot Encoding

Lastly we had to deal with our categorical variables. To deal with the categorical variables we implemented one hot encoding, For example variables like location source could take on five possible values (ci, nc, us, uu, uw), for each of these location sources a new variable was added that contained the value 0 or 1 representing whether or not a location source took on either (ci, nc, us, uu, uw).

3. Model Validation

In order to validate our results we will first split our data using a train-test split. 80% of our data will be used for training while 20% will be used for testing. We will use the root mean square error (RMSE) as our primary evaluation metric for measuring how well our models perform.

4. Models

4.1 OLS Regression (with MSE Loss):

Linear regression seeks find a vector w that will minimize the following:

$$\sum_{i=1}^{n} (y_i - w^T x_i)^2$$

Where y_i are the labels, w is the weight vector, and x_i are the feature vectors. This approach, known as ordinary least squares

(OLS) regression will be our first try at fitting a model to our data. We will use MSE (mean squared error) as our loss function. Since area data is not extremely multicollinear we thought this would be a good place to start for our model.

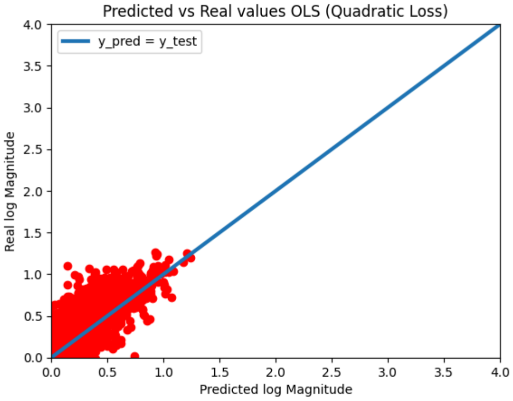

Figure 8 OLS Model

After fitting the model it had a RMSE of 0.252 on the train dataset and a RMSE of 0.257 on the test dataset. This result is decent overall and it is noteworthy that the model is not overfitting. On average our magnitude measurements are off by about 1.07 on the Richter scale, which while not perfectly accurate, can still help drive informed decisions about disaster response. Our model had an R² value of about 0.61 which further reinforces that we have a good model but not a perfect model. Knowing this, we wanted to try a polynomial regression model to see if we could obtain more accurate results by using more complexity.

4.2 Polynomial Regression

We define a basic polynomial transformation and model as follows:

$$\phi(x) = (1, x, x^2, x^3, \ldots, x^{d-1})$$

$$w^T\phi(x) = w_1 + w_2x + w_3x^2 + \ldots + w_dx^{d-1}$$

Where $\phi(x)$ is our transformation function, x is a vector of single variables, and w is the weight vector. Obviously in our case, we have several variables, so we will have to multiply them together in creation of a much more complex model. For our model we will use a polynomial transformation of degree 2. We still used MSE loss as our loss function.

When fitting and evaluating our model we had a RMSE of 0.202 on our training dataset but a RMSE of 3530.9 on our test dataset. While the more complex model was able to better represent our training dataset it generalized extremely poorly and overfitted our data. You can see this effect by comparing **Figure 9** and **Figure 10**. The former has difficulty predicting the higher magnitude earthquakes while the latter has overall better predictions. Since our model is overfitting the data, trying a higher degree model did not make sense since it would just further increase model complexity and overfit the data more.

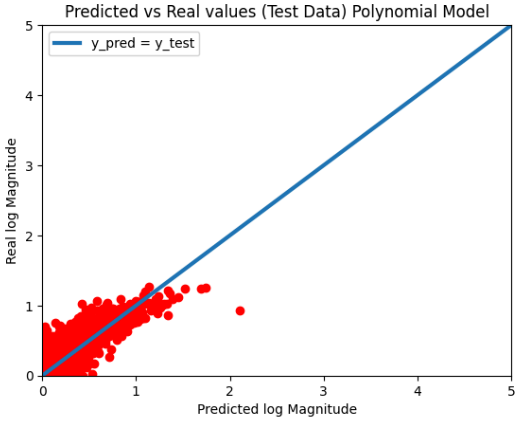

Figure 9 Polynomial Model (Test Data)

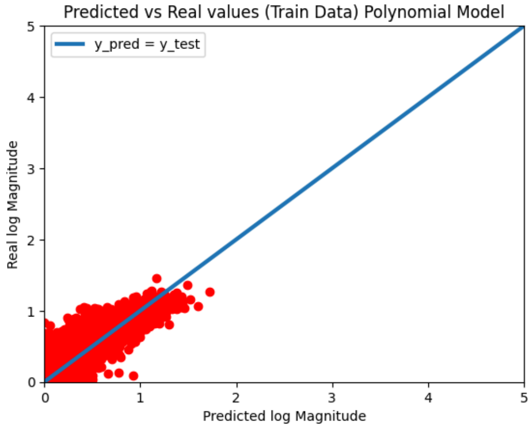

Figure 10: Polynomial Model (Train Data)

We suspected that the multicollinearity between some of our features might have been what was throwing off our model since the more complex model could have been fitting the noise associated with the multicollinear features and not the signal. To fix the overfitting we thought to try and add ℓ₂ regularization to our model.

4.3 Ridge Regression

We define ridge regression as a minimization of the following function:

$$||y - Xw||^2 + \lambda||w||^2$$

Where y is the vector of labels, X is the matrix of features, w is the weight vector, and $\lambda > 0$ is the regularization parameter. By introducing the penalty term $\lambda||w||^2$ to our loss function we reduce the variance of the coefficient estimates. When multicollinearity is high this might prevent the model from fitting overly complex patterns that are not actually supported in the data.

We will use ridge regression in conjunction with a prior 2nd degree polynomial regression model. In addition to this we used 5-fold cross validation inorder to optimize the value of λ.

After fitting and testing our model we found that ℓ₂ regularization had indeed fixed the overfitting in our previous model. Our new model had a train RMSE of 0.219 and a test RMSE of 0.22 Our 5-fold cross validation found the optimal λ to be 109.85. This is the best overall model we found that did not overfit the data. The predicted vs actual log magnitudes are seen in figure 11.

Since our model was no longer overfitting we again tried to increase the complexity this time increasing the degree of the polynomial regression to 3. This led to even better model performance but we started to see some slight overfitting. The train RMSE was 0.188 and the test RMSE was 0.202.

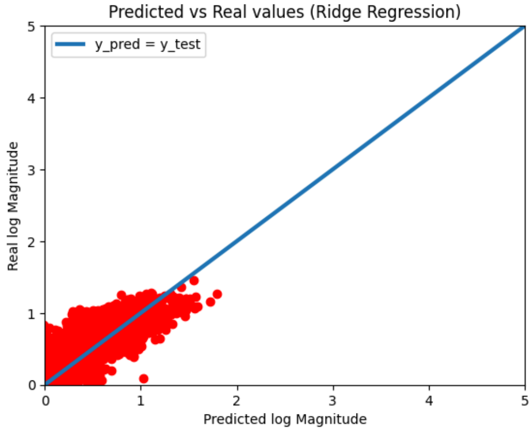

Figure 11: Ridge Regression w/ 2nd Order Transformation

5 Adjustments

5.1 Addressing Fairness and A possible Weapon of Math Destruction

Since our data does not directly interact with humans, we do not have to worry about a natural bias towards certain groups of people. Additionally, the nature of seismicity is unaffected by whatever models we make of them, so employing these models serves no harm in that regard. The only case we need to be careful with in our model is the event in which it becomes used for prediction and alerting for safety procedures. If this happens, we must take extra precautions to ensure that our model does not under-predict earthquake magnitudes, creating a false sense of security that could result in needless harm. Alternatively, we could leave the model unchanged and communicate effectively that certain earthquakes may be under-predicted, facilitating employment of proper safety precautions or extra oversight of the model.

5.2 Future Improvements

Despite improvements, there still exist inaccuracies in our best models. The lack of high-magnitude earthquake data when compared to the abundance of low-magnitude earthquake data is an important problem to tackle; one that may only be solvable fully as we get more data on large earthquakes. Despite being able to improve our model's overfitting with ℓ_2 regularization, multicollinearity might still be a reason behind higher order models tending to overfit. The chief opportunity to curb this problem is by use of principal component analysis (PCA) on our feature matrix to reduce complexity and hopefully multicollinearity. By employing PCA, we could reduce these correlations in our model, generating a more accurate model while hopefully successfully reducing our overfitting problem.

5.3 The High-Magnitude Problem

As alluded to prior, the most significant problem facing our datasets and model creation was the lack of high magnitude earthquakes recorded. Obviously more data would help, but a potential solution that was not implemented in this project would involve requesting from the NEIC a dataset that covers as far back as possible, in order to retrieve as many high magnitude earthquakes as possible. Considering there are typically only about 15 magnitude 7.0 or higher earthquakes globally per year (and about 1 per year in California), this new dataset would still lead to models that perform poorly at the very high ranges, but would likely much more accurately cover magnitudes 4, 5 and 6. Practically speaking, we could also consider omitting magnitudes 0 through 3.0 from our datasets, as these small-scale earthquakes are typically not even felt by humans, and thus have less significance for safety. By deleting these entries and increasing our high-magnitude entry count, our model would naturally shift to accommodate the more dangerous earthquakes, and we could sustain increased accuracy in prediction of these more specific types of earthquakes.

6 Conclusion

Earthquake magnitude is an important measurement for safety and research purposes, and understanding seismicity by any means possible is paramount to advancing these causes. The best model we were able to produce was a polynomial regression model that used ℓ_2 regularization with MSE as our loss function. This model has a RMSE of 0.22 and can provide a good ballpark estimate, typically within about 1.05 of the true magnitude on the Richter scale. This is enough for providing a decent warning, but only if systems are fast enough to record these variables in time before an earthquake's more destructive surface waves arrive. Regardless of whether a model like this is efficient and accurate enough to be used in earthquake warning systems, it provides crucial insight into how earthquake magnitude is determined by auxiliary variables.

Team Contributions:

Stavros: Coding, Idea Generation, worked on project report

Vincent: Provided the data. Knowledge about earthquakes and the dataset. Idea Generation, worked on project report

Sources:

National Earthquake Information Center (NEIC) Catalog, run by United States Geological Survey (USGS): https://earthquake.usgs.gov/earthquakes/search/

CMT Earthquake Gallery, run by Lamont-Doherty Earth Observatory (Columbia University): https://www.ldeo.columbia.edu/~gcmt/projects/CMT/EQgallery_old/EQgallery.html