# An Analysis of United States Geographical Survey Data

By Patrick Johnson

December 14, 2018

**Abstract**

This report investigates the relationship between three dimensional location of seismic events - that is latitude, longitude, and depth - and magnitude of the resulting seismic event. In this investigation, four methods of machine learning are used to analyze the relationship, as well as offer predictions for new seismic events. These four methods include the nearest neighbors algorithm, Gaussian process regression, linear regression, and a neural network model.

## 1 Introduction

The data collected for this dissertation was extracted from the USGS Earthquakes Hazards Program Database [1]. It contains all recorded seismic activity from the past thirty days across the world, and has approximately 10,000 entry vectors. Each of these data vectors contains 22 features:

- Time (YYYY/MM/DD)
- Latitude
- Longitude
- Depth
- Magnitude
- Magnitude Type
- NST
- Gap
- Dmin
- RMS
- Net
- ID
- Updated Time
- Place
- Type
- Horizontal Error
- Depth Error
- Magnitude Error
- Magnitude NST
- Status
- Location Source
- Magnitude Source

Of these 22 features, latitude, longitude, and depth of seismic event were selected to represent the independent variables. The other 18 were left out for various reasons which included but was not limited to the nature of their input (String values), the insignificance of their information, or the incompleteness of the fields.

# 2 Analysis

## 2.1 K-Nearest Neighbors Algorithm

The Nearest Neighbors Algorithm is one that takes a training set and constructs a KD tree with the vectors. This is done by cycling through the dimensions of the data set and splicing the set in half based on the median of the data in that respective dimension. A two dimensional example of this can be seen in the following figure:

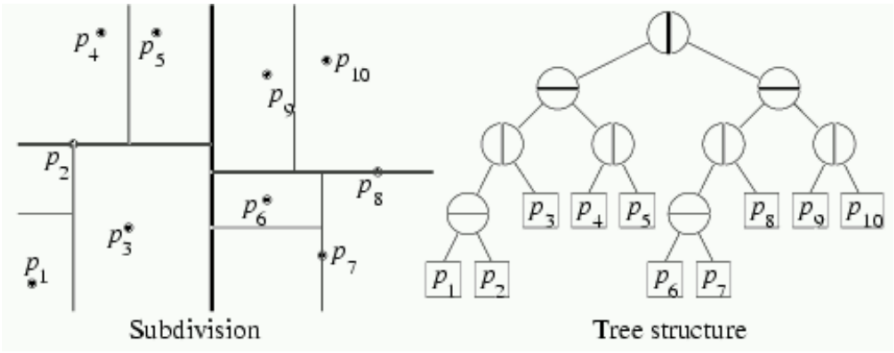


Figure 1: KD Tree In Two Dimensions[2]

### 2.1.1 Impact of Training Data Size

The first investigation applied by this algorithm was the relationship between percentage of data used for training purposes and the resulting root mean squared error of the nearest neighbor classification. It was expected that there is a strong negative correlation correlation between these two parameters. The results of this investigation can be seen in the following figure:

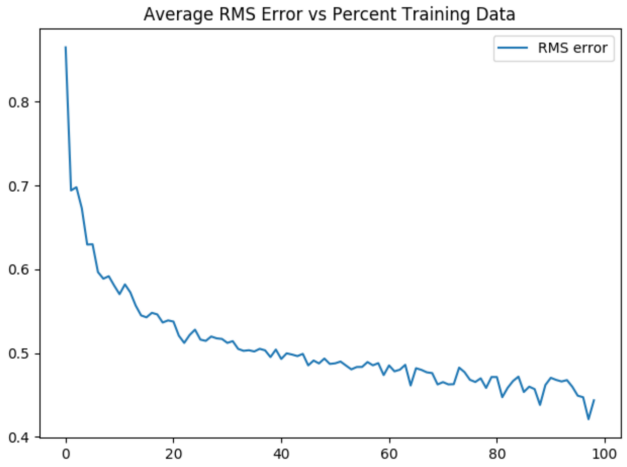


Figure 2: Average RMSE for Latitude, Longitude, and Depth as Inputs

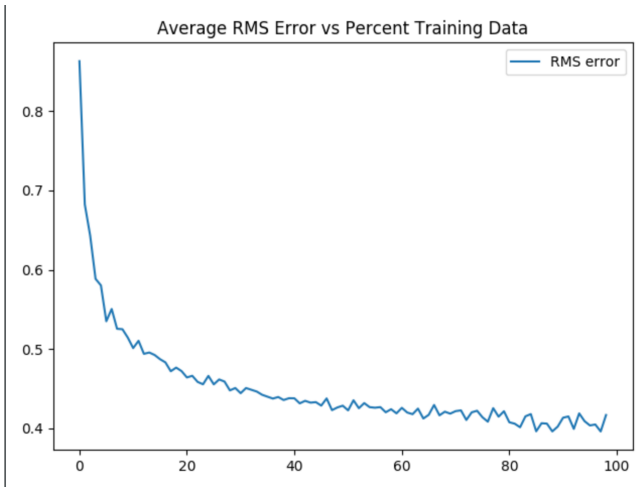


Figure 3: Average RMSE for Latitude and Longitude as Inputs

The minimum RMS value from Figure 2 was 0.41 and occurred at 98% of the data being used for training. The minimum RMS value from Figure 3 was 0.39 and also occurred at 98% of the data being used for training. This result shows that the addition of the depth parameter may worsen the results of the nearest neighbor classification process. This phenomenon is best described as the curse of dimensionality within the Nearest Neighbors Algorithm. As the dimensionality of our model increases, each data vector begins to have an over-inflated representation of the space; the distance between each point approaches the distance of the entire data space. This problem is solved as the amount of training data is driven to infinity, but of course this is not a feasible operation as there are computational and physical limitations at play.

The purpose of this analysis is to identify the optimal percentage of data to use for training without the possibility of potentially overfitting our model. While 98% was the percentage which minimized the RMS error, using 80% of the data to train on would get similar results for both sets of features, is much more plausible for real world application, and will help prevent the issue of overfitting our data to our training set.

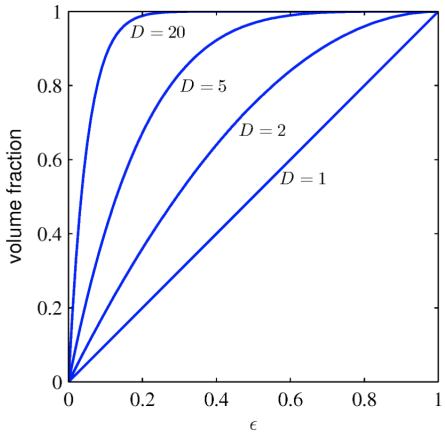


Figure 4: Curse of Dimensionality[3]

### 2.1.2 Impact of Number of Nearest Neighbors

The next investigation that the Nearest Neighbors Algorithm was used for was the relationship between number of neighbors used to classify the seismic activity and the corresponding root mean squared error. The process used for this was not necessarily the standard classification process. In most multiple neighbor classification processes, the majority label of the nearest neighbors to the test vector would be the resulting predicted label. However, this study is based on a magnitude output which is represented as a float with one decimal. As a result, there are not enough training vectors to represent all the possible magnitudes. Thus, for this classification the average of all the nearest neighbor magnitudes was the resulting predicted outcome for the given input parameters.

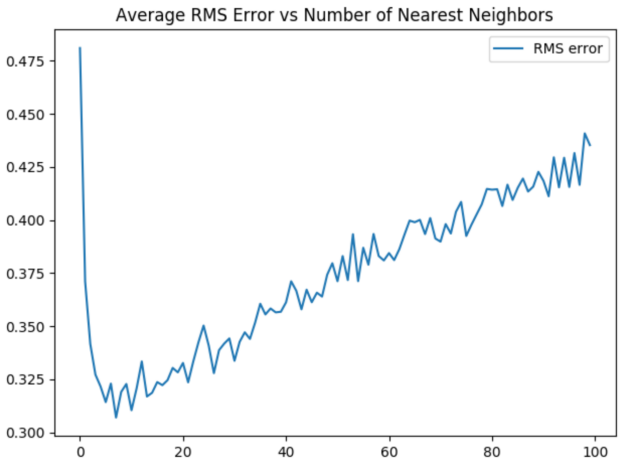


Figure 5: Average RMSE for Latitude, Longitude, and Depth as Inputs

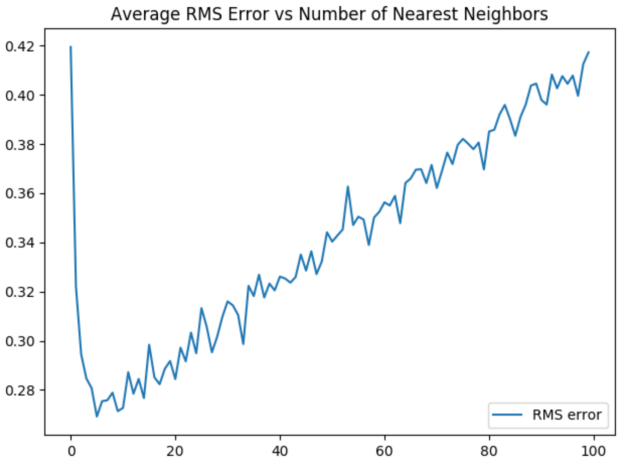


Figure 6: Average RMSE for Latitude and Longitude as Inputs

The minimum RMS value from Figure 5 was 0.314 occurred at a nearest neighbor count of 9. The minimum RMS value from Figure 6 was 0.269 occurred at a nearest neighbor count of 7. This analysis shows how the classification accuracy is extremely dependent on the number of nearest neighbors across both sets of features. This is because when K is small, the classification is dependent on too small of a data set to accurately encompass the seismic magnitude. That is, the small bin sizes create very spiky responses in magnitude. When K gets too large, the large bin sizes make the classification dependent on too large of a data set and includes data that is not related to the test vector. This is also seen as too smooth of a response from the prediction model as the bins get wider.

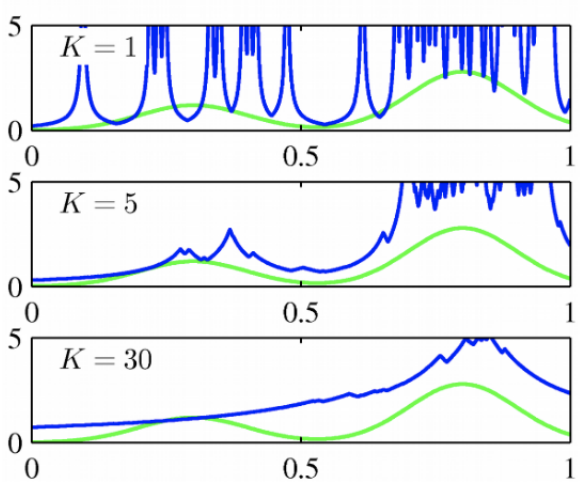


Figure 7: Effect of K on Probability Density [4]

## 2.2 Polynomial Regression

Polynomial regression is a machine learning technique that takes in a training set of data and finds the corresponding weights for the polynomial function that will minimize the RMS error of the data points and the line of best fit. The degree of the polynomial function is a hyperparameter which can be fine tuned to improve the accuracy of the model.

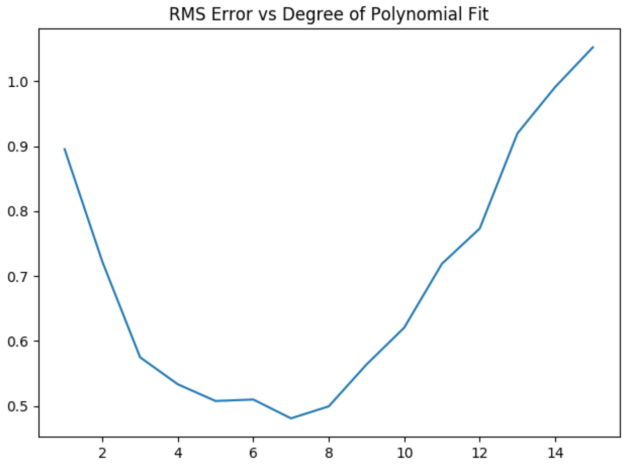


Figure 8: RMS Error Response for Latitude, Longitude, and Depth

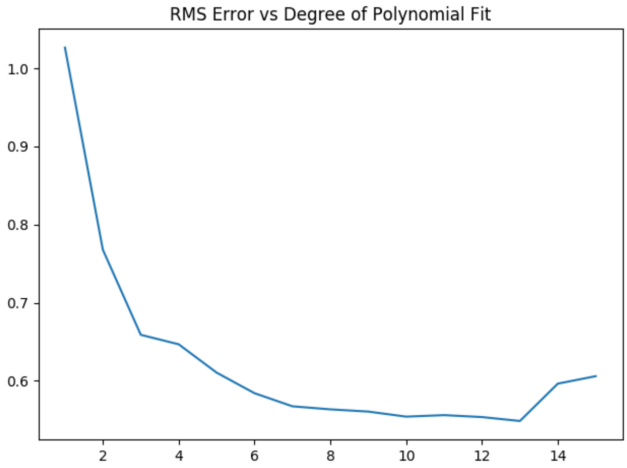


Figure 9: RMS Error Response for Latitude and Longitude

Figure 8 shows the degree of polynomial which minimizes the RMS error is 7, corresponding to an error of 0.488. Figure 9 varies slightly, giving the indication that a 13th degree polynomial of best fit will minimize the RMS error of classification to 0.564. Both of these figures show the overall response of error to the degree of polynomial, a common problem known as overfitting.

This issue comes about when the line of best fit matches the training data perfectly, but fails to express the underlying relationshib between the inputs and outputs. Therefore, when test vectors that lie between the perfectly fit training vectors, they lie on the steep hills of the overfit curve.

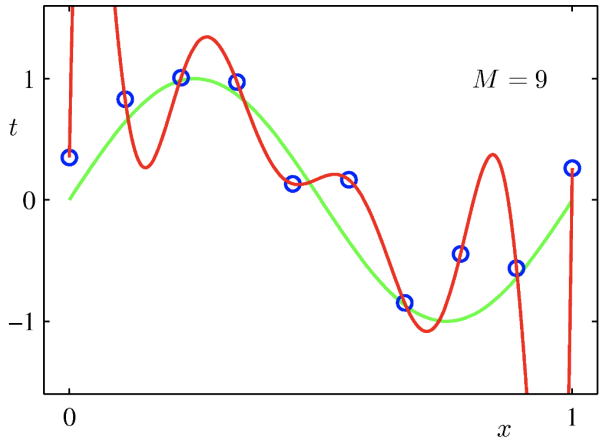


Figure 10: Example of Overfitting Data [5]

## 2.3 Gaussian Process Regression

Gaussian Process Regression is a technique used to model relationships between data by using a kernel function to associate closely distanced points and issue a classification in proportion to a test vector's distance to all points in the space. This form of supervised learning most commonly uses the squared exponential or regular exponential kernel function to accomplish this. The squared exponential kernel will be used for this analysis.

### 2.3.1 Impact of Standard Deviation

The standard deviation of the kernel function will be the hyperparameter of choice to fine tune to improve the accuracy of the model.

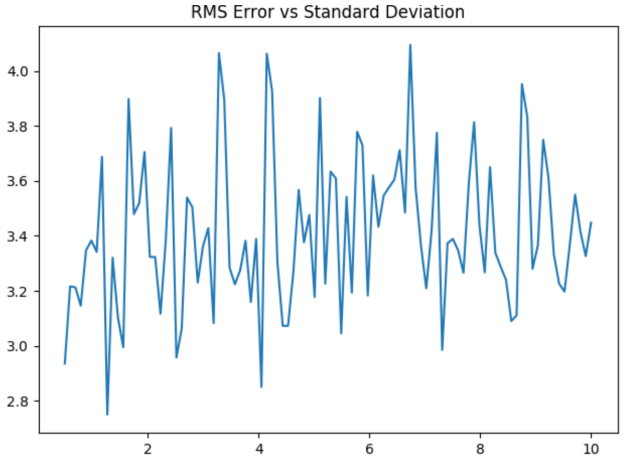


Figure 11: RMS Error for Latitude and Longitude

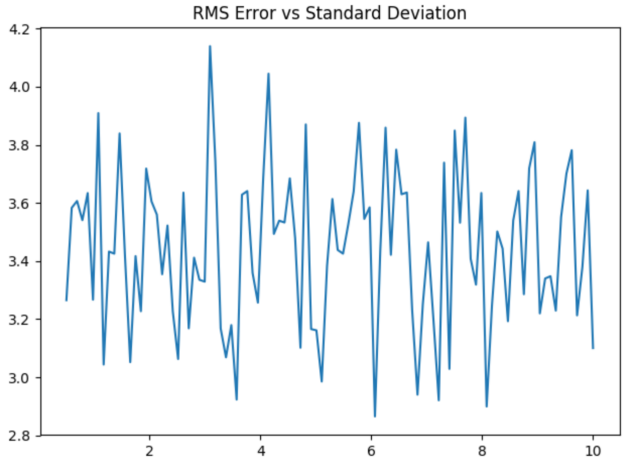


Figure 12: RMS Error for Latitude, Longitude, and Depth

These two figures offer no real correlation between the standard deviation of the kernel function and the corresponding RMS errors. Because a selection needed to be made, a standard deviation of 1.3 was selected for the next analysis.

### 2.3.2 Impact of Random Restarts

Gaussian Process Regression will attempt to converge on the local minimum of RMS error, but there is no telling if the local error it is converging on is the global minimum. The method of random restarts attempts to fix this by restarting the convergence process to ensure that the pocket of weights the process has converged upon does in fact yield the minimum RMS error, or if there is a better pocket somewhere else.

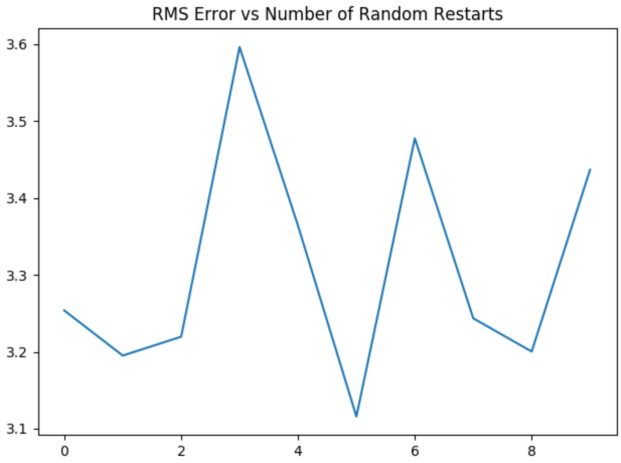


Figure 13: RMS Error for Latitude and Longitude

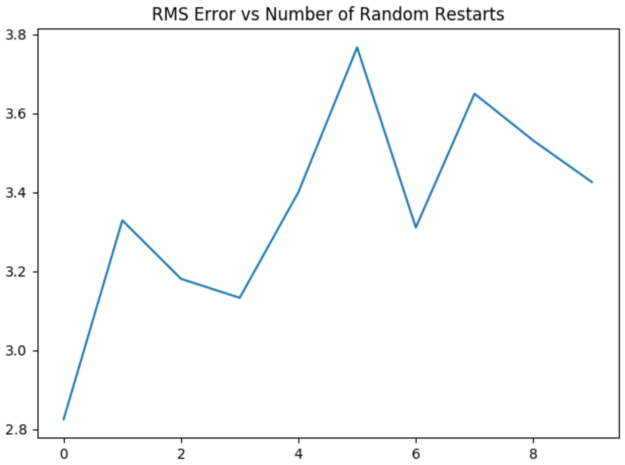


Figure 14: RMS Error for Latitude, Longitude, and Depth

Again, these two figures offer no real correlation between the number of random restarts and the RMS error. Even if there were a negative correlation, the amount of time it takes for the additional computation is not worth the slight possible decrease in RMS error.

Overall, the Gaussian Process Regression model did a poor job in classifying the test seismic events' magnitudes. In fact, it underperformed to such an extent that it was necessary to put rails on the output that it produced so that the predicted magnitudes could not go above 7 or below 0. These values make sense in the physical world because it is so unlikely, according to past data, that a seismic event will have a magnitude over 7. Also, it is impossible to record a magnitude of below zero, so it only makes sense to clip results below it.

## 2.4 Neural Network

A neural network is one of the most popular machine learning techniques used to model and classify data. Its popularity can be attributed to its accuracy and overall speed with big data sets, which is why it fits perfectly into the scope of this study. The neural network operates by having several nodes in place throughout several layers which all perform different transformations on the data that passes through it. The result of the transformations is used to model the data. This inevitably has some error involved, and the error is propagated back through the network so the weights of the transformation functions can be adjusted in such a way that it minimizes the RMS error of the classification. Because neural networks classify categorical data, it was necessary to group the data into integer magnitudes. That is, all of the seismic activity magnitudes were rounded to their nearest whole number so that the neural network could appropriately classify it.

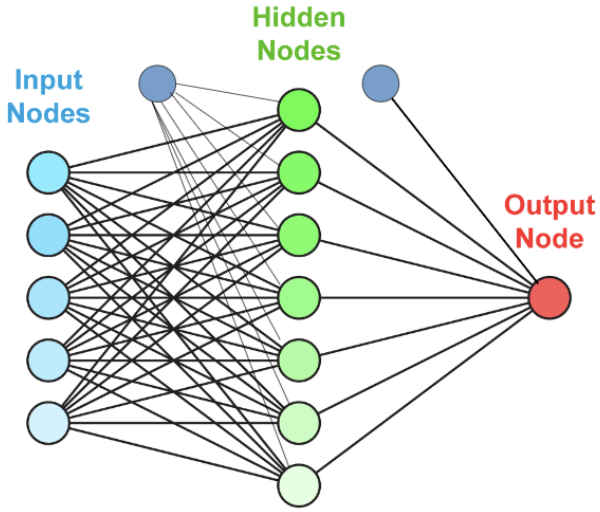


Figure 15: Depiction of Neural Network Model [6]

### 2.4.1 Impact of Number of Nodes

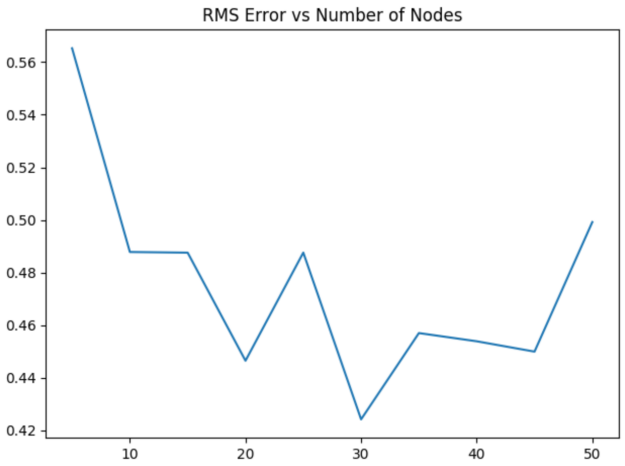


Figure 16: RMS Error for Latitude, Longitude, and Depth

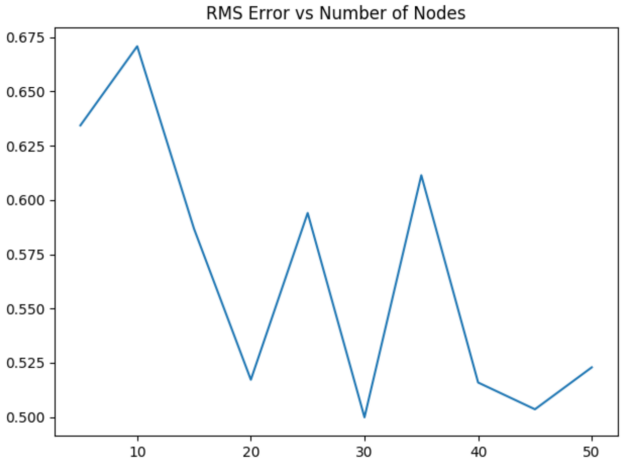


Figure 17: RMS Error for Latitude and Longitude

Each node in the neural network acts as an identifier for the overall model. That is, when a incoming test vector comes through it is weighted according to its likelihood of belonging in one of the predetermined classes. When the number of nodes grows too large, this is yet another example of overfitting data. Because there are so many nodes, they begin to behave too strongly as identifiers. This means that they can identify one specific point very well, but beyond that they struggle to represent the underlying relationship between the input parameters and the seismic event magnitude. This phenomenon can be observed in both Figure 15 and Figure 16. As the number of nodes increases, the RMS error dips down to its minimum and then rises back up.

### 2.4.2 Impact of Number of Layers

Unlike the number of nodes in each layer, the number of layers is not connected to the issue of overfitting that has been seen in the previous investigations. The following figures show the overall negative correlation between the number of layers in the neural network model and the RMS error experienced through classification on the test vectors.

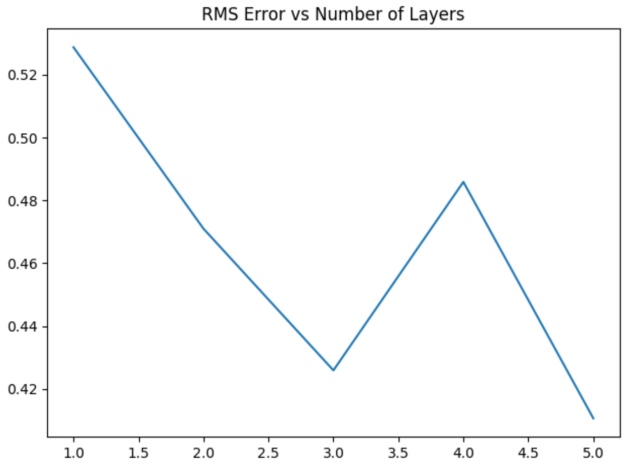


Figure 18: RMS Error for Latitude, Longitude, and Depth

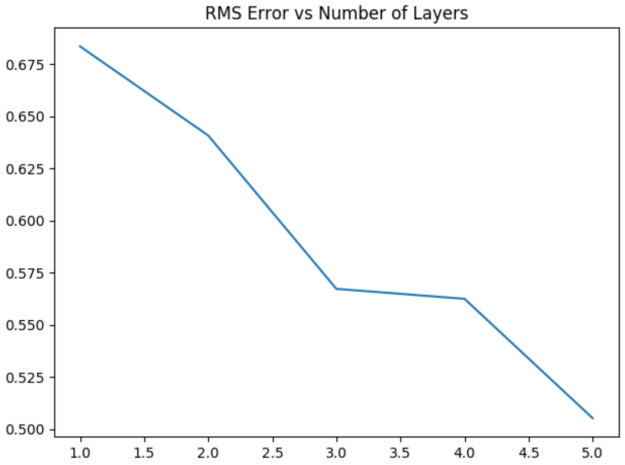


Figure 19: RMS Error for Latitude and Longitude

# 3 Discussion

The four approaches taken to model and classify the USGS data all have their advantages as well as pitfalls. Unfortunately, there is no one size fits all solution

for this problem of getting the optimal accuracy and minimum computational complexity trade-off.

Table 1: Performance Results

| Method | RMS Error | Run Time (s) |
|---|---|---|
| Nearest Neighbor | 0.288 | 0.068 |
| Poly Regression | 0.502 | 0.042 |
| Gaussian Process | 3.627 | 30.88 |
| Neural Network | 0.436 | 6.98 |

This table was constructed using the optimal hyperparameter conditions that had been discovered through the previous investigations. These results indicate that the Nearest Neighbors Algorithm was the most accurate, while the Polynomial Regression was the fastest in the way of computational complexity.

# 4 Conclusion

Overall, three of the four methods used to model the underlying relationship between the location of a seismic event and the magnitude of event did a decent job. This analysis shows how all four of the methods can be applied to a data set. While this application is just one of many, being able to learn behaviors such as this will allow conclusions to be drawn and predictions to be made about catastrophic events before they have the ability to devastate areas. There is no free lunch when it comes to finding the best model for a data set, but these investigations show that more complex model does not correlate to a more accurate prediction. Through the process of hyperparameter fine tuning, it was found that the optimal model for this data set was a KD tree using the three dimensions of latitude, longitude, and depth. The optimal classification algorithm was then decided to be the nearest neighbors classification process with 9 nearest neighbors used to sample.

# References


[1] United States Geographical Survey Data, Earthquake Hazard Program, https://earthquake.usgs.gov/earthquakes/feed/

[2] Dr. Christopher Crick, Oklahoma State University, Lecture 8/24 Slide 9

[3] Dr. Christopher Crick, Oklahoma State University, Lecture 8/24 Slide 7

[4] Dr. Christopher Crick, Oklahoma State University, Lecture 8/24 Slide 13

[5] Dr. Christopher Crick, Oklahoma State University, Lecture 9/24 Slide 13

[6] Dr. Christopher Crick, Oklahoma State University, Lecture 10/3 Slide 18