Data 101 Project on Earthquakes- Anika Sharma and Vrinda Surati

Abstract:
The quakes dataset, built into R, contains seismic data regarding 1,000 earthquakes near Fiji between 1964-1975. Each observation includes the earthquake's latitude and longitude, depth in kilometers, Richter magnitude, and the number of seismic stations that reported the earthquake.

This project uses machine learning techniques on the data, including decision tree classification and Naive Bayes modeling, to categorize earthquakes according to magnitude. The mag feature is separated into categories ("Low", "Medium", "High") as it is a continuous variable in order to allow for classification modeling. Preprocessing techniques such as feature binning and factor conversion are implemented to prepare the data for analysis.

Models are validated using accuracy, kappa statistics, and cross-validation techniques. This report analyzes the application of models like decision trees and probabilistic classifiers like Naive Bayes in the examination of geological processes, and pattern identification regarding earthquake magnitude.

Introduction:
Earthquakes are natural catastrophes that may be disastrous and long-lasting in their effects on people and places. Knowledge of the essential characteristics of seismic activity — location, depth, and magnitude — is required for improving disaster readiness and constructing risk management reactions. The ability to classify earthquakes based on measurable characteristics can potentially aid seismologists and emergency management officials in quickly estimating the potential damage of an earthquake.

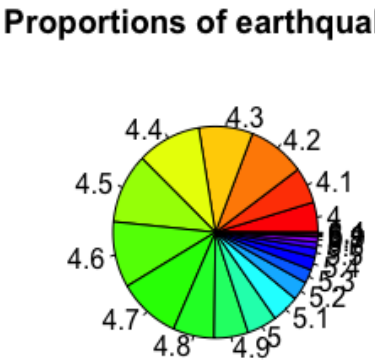
We used the quakes dataset in R for this project, which is a native dataset containing information on 1,000 earthquakes near Fiji, with the features mentioned above (latitude, longitude, depth, magnitude, and the number of seismic stations where the earthquake was recorded).To create a classification problem, we categorized earthquake magnitudes into three different levels: Low, Medium, and High.

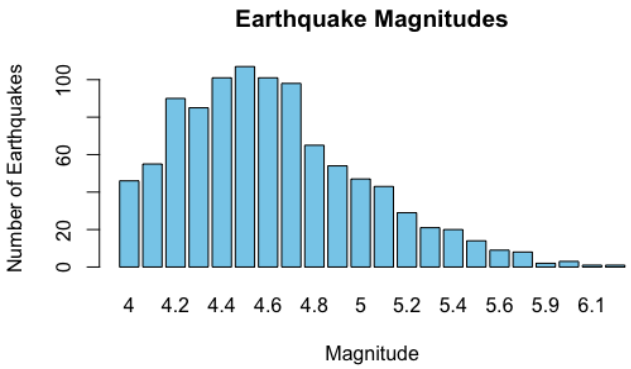
A tree-based model was constructed using the rpart package in R. To enhance model fit and prevent overfitting, techniques such as pruning, 10-fold cross-validation, and a Naive Bayes classifier were carried out and tested. The two models were compared in their effectiveness using confusion matrices and classification metrics such as accuracy and Kappa statistics. These approaches allowed us to identify which features — such as depth or location — are most predictive of earthquake magnitude, and how machine learning techniques can be applied to gain insight about earthquake data.

Materials and Methods:
For the purpose of this project, the built-in quakes dataset from R was used, which contains 1,000 observations and six variables describing earthquake occurrences in the Fiji region, including latitude, longitude, depth, magnitude, and the number of seismic stations that recorded the earthquake. A categorical variable, mag_cat, was created from the numerical magnitude values, dividing each earthquake into one of three categories: Low, Medium, or High. The R programming language was used in analysis, with the application of libraries such as rpart, rpart.plot, caret, klaR, and naivebayes.

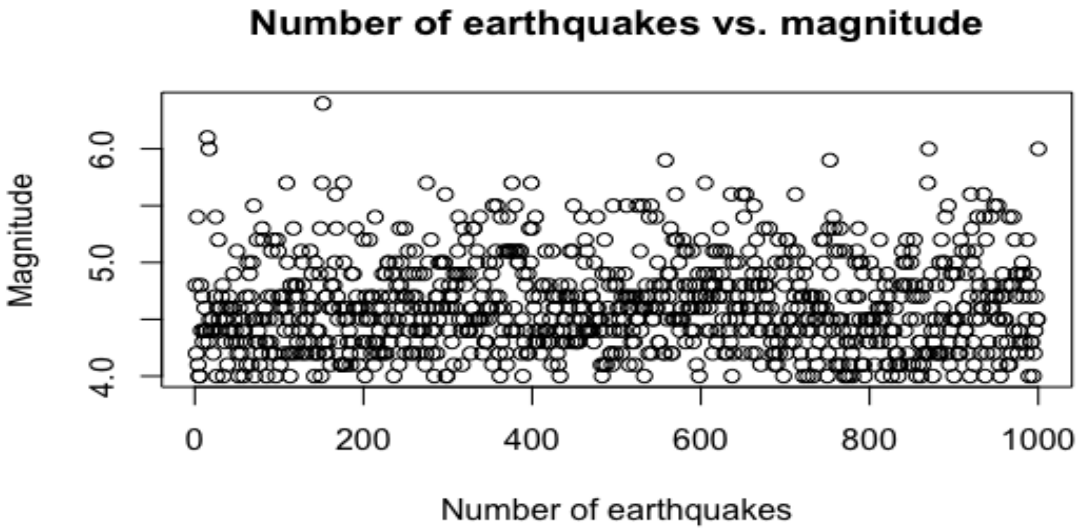
Before modeling, the data set was summarized and explored to discover distributions and check for any missing values. Different plots of the data, including bar plots, pie charts, dot plots, and boxplots, were made to illustrate the frequency and distribution of earthquake magnitudes. In terms of classification, we constructed both a decision tree model using the rpart package and a Naive Bayes classifier using the naivebayes and klaR libraries. A decision tree was built with all the features to predict the mag_cat class and plotted using rpart.plot. In the interest of improving the performance and validating the models, 10-fold cross-validation was performed using the caret package. Both models were then evaluated using confusion matrices to verify their classification accuracy and effectiveness.

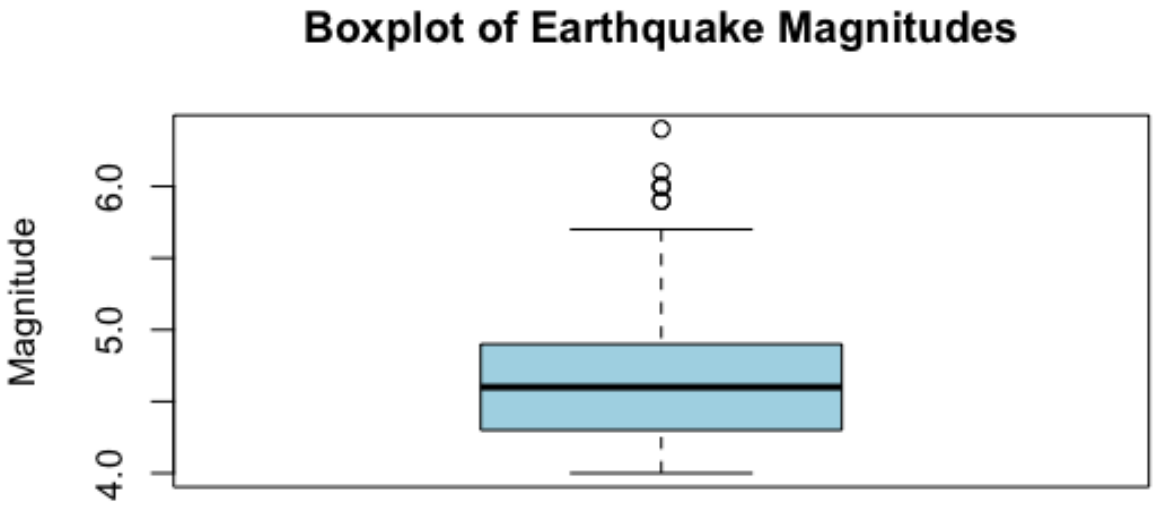
Results:
The dataset had most of the observations falling into the Low and Medium magnitude categories, with an overall range of earthquake magnitudes going from 4.0 to 6.4. Summary statistics of the dataset showed that depth of the earthquakes ranged from 40 to 680 kilometers, and the number of reporting stations varied between 10 and 132.The High magnitude earthquakes were relatively rare, which made it an imbalance in the classes. The charts that we made with R, namely the pie chart and the bar graph, clearly demonstrated that low or medium magnitude earthquakes dominate the data set.



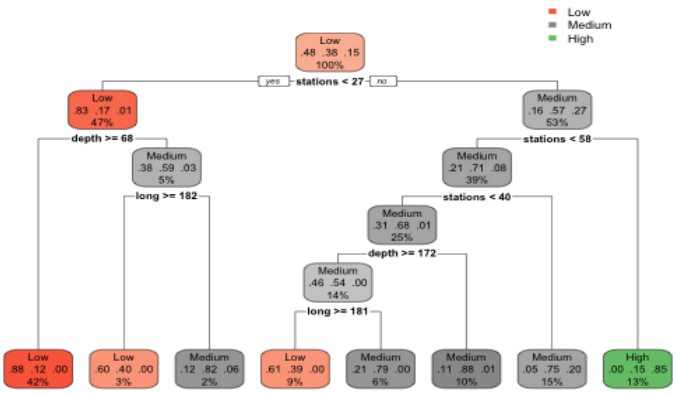


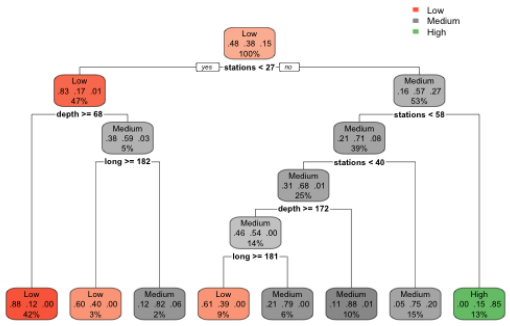
The boxplot revealed the central tendency, which was closer to the middle magnitude. The dot plot showed the distribution of the magnitudes across the data set, and more concentrated areas appear darker between the 4.2 - 4.7 range.





The classification tree showed that the number of stations, depth and longitude were the most important predictors for magnitude classification, and then the pruned decision tree had a 73.7% accuracy on the test set. The pruned tree looks exactly like the original decision tree because the optimal cp did not actually prune away any additional nodes because of the smaller size.





```
Confusion Matrix and Statistics

          Reference
Prediction Low Medium High
     Low     133      40    1
     Medium   18      54   14
     High      0       6   34

Overall Statistics

               Accuracy : 0.7367
                 95% CI : (0.683, 0.7856)
    No Information Rate : 0.5033
    P-Value [Acc > NIR] : < 2.2e-16

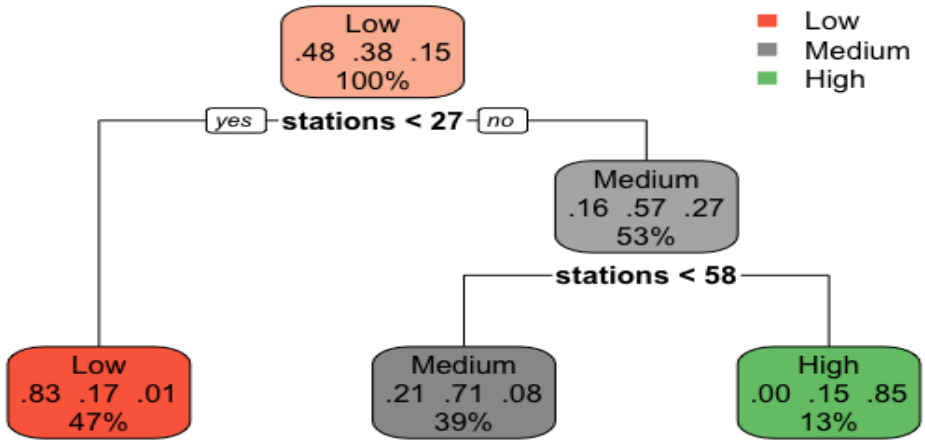
                  Kappa : 0.5542

 Mcnemar's Test P-Value : 0.005732

Statistics by Class:

                     Class: Low Class: Medium Class: High
Sensitivity              0.8808          0.5400       0.6939
Specificity              0.7248          0.8400       0.9761
Pos Pred Value           0.7644          0.6279       0.8500
Neg Pred Value           0.8571          0.7850       0.9423
Prevalence               0.5033          0.3333       0.1633
Detection Rate           0.4433          0.1800       0.1133
Detection Prevalence     0.5800          0.2867       0.1333
Balanced Accuracy        0.8028          0.6900       0.8350
```

When we did the 10 fold cross validation, it found that the best cp was 0.02179837, and it produced a tree with an accuracy of 72.7%. The confusion matrix showed some confusion between the low and medium classifications of magnitudes.




```
Confusion Matrix and Statistics

          Reference
Prediction Low Medium High
    Low    124      34    1
    Medium  27      60   14
    High     0       6   34

Overall Statistics

               Accuracy : 0.7267
                 95% CI : (0.6725, 0.7763)
    No Information Rate : 0.5033
    P-Value [Acc > NIR] : 2.627e-15

                  Kappa : 0.5439

 Mcnemar's Test P-Value : 0.1716

Statistics by Class:

                     Class: Low Class: Medium Class: High
Sensitivity              0.8212         0.6000       0.6939
Specificity              0.7651         0.7950       0.9761
Pos Pred Value           0.7799         0.5941       0.8500
Neg Pred Value           0.8085         0.7990       0.9423
Prevalence               0.5033         0.3333       0.1633
Detection Rate           0.4133         0.2000       0.1133
Detection Prevalence     0.5300         0.3367       0.1333
Balanced Accuracy        0.7931         0.6975       0.8350
```

The Naive Bayes model was trained using the same predictor values, however we received warnings about numerical zero probabilities on certain observations, which meant that we did not get the final accuracy results. This is a known limitation of the Naive Bayes model, indicating that feature combinations in the test set were not observed during training which means that they cannot be calculated.

```
Naive Bayes

700 samples
  4 predictor
  3 classes: 'Low', 'Medium', 'High'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 631, 630, 630, 629, 629, 631, ...
Resampling results across tuning parameters:

  usekernel  Accuracy   Kappa
  FALSE      0.7741128  0.6237505
   TRUE      0.7798484  0.6321326

Tuning parameter 'laplace' was held constant at a value of 0

Tuning parameter 'adjust' was held constant at a value of 1
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were laplace = 0, usekernel =
 TRUE and adjust = 1.
```

Warning: Numerical 0 probability for all classes with observation 4Warnin
g: Numerical 0 probability for all classes with observation 6Warning: Nume
rical 0 probability for all classes with observation 7Warning: Numerical 0
probability for all classes with observation 8Warning: Numerical 0 probabi
lity for all classes with observation 11Warning: Numerical 0 probability f
or all classes with observation 13Warning: Numerical 0 probability for all
classes with observation 14Warning: Numerical 0 probability for all classe
s with observation 15Warning: Numerical 0 probability for all classes with
observation 21Warning: Numerical 0 probability for all classes with observ
ation 25Warning: Numerical 0 probability for all classes with observation
27Warning: Numerical 0 probability for all classes with observation 28Warn
ing: Numerical 0 probability for all classes with observation 33Warning: N
umerical 0 probability for all classes with observation 34Warning: Numeric
al 0 probability for all classes with observation 36Warning: Numerical 0 p
robability for all classes with observation 40Warning: Numerical 0 probabi
lity for all classes with observation 41Warning: Numerical 0 probability f
or all classes with observation 45Warning: Numerical 0 probability for all
classes with observation 55Warning: Numerical 0 probability for all classe
s with observation 57Warning: Numerical 0 probability for all classes with
observation 59Warning: Numerical 0 probability for all classes with observ
ation 68Warning: Numerical 0 probability for all classes with observation
76Warning: Numerical 0 probability for all classes with observation 89Warn
ing: Numerical 0 probability for all classes with observation 90Warning: N
umerical 0 probability for all classes with observation 92Warning: Numeric
al 0 probability for all classes with observation 93Warning: Numerical 0 p
robability for all classes with observation 95Warning: Numerical 0 probabi
lity for all classes with observation 97Warning: Numerical 0 probability f
or all classes with observation 99Warning: Numerical 0 probability for all
classes with observation 102Warning: Numerical 0 probability for all class
es with observation 104Warning: Numerical 0 probability for all classes wi
th observation 111Warning: Numerical 0 probability for all classes with ob
servation 115Warning: Numerical 0 probability for all classes with observa
tion 117Warning: Numerical 0 probability for all classes with observation
118Warning: Numerical 0 probability for all classes with observation 120Wa
rning: Numerical 0 probability for all classes with observation 122Warnin
g: Numerical 0 probability for all classes with observation 127Warning: Nu

Discussions:
The classification models provided insight into which factors might influence the magnitude of an earthquake, although there were limitations due to the nature of the dataset and how rarer it was to see high magnitude earthquakes. The decision tree analysis showed that depth and the number of reporting stations were some of the most important features in predicting magnitude categories. Considering that there was an uneven distribution of classes—particularly the small proportion of High-magnitude earthquakes—the tree struggled to make accurate predictions for that category. This imbalance likely contributed to some overfitting in the original tree, which was addressed through pruning and 10-fold cross-validation. These actions led to a slight drop in accuracy, but not so much to cause concern. Similarly, the Naive Bayes model performed comparably to the decision tree, but also showed reduced accuracy when applied to the less frequent classes, hence the warning messages. Both models demonstrated how limited class variation and overlapping data distributions can affect classification performance. While the models were effective at recognizing general patterns, improvements could be made by incorporating features that can account for unevenness in data.

Acknowledgements:
Chat GPT was used in troubleshooting when we had problems with our code, and helping us figure out how each of the functions in R work (if we had trouble figuring it out). The example project files attached in Canvas were also referenced when creating our own report. The dataset we used exists in R and is called “quakes.”

Literature Cited:
Quakes dataset: https://www.rdocumentation.org/packages/datasets/versions/3.6.2/topics/quakes

Extra Information:
Refer to the code for the head(quakes) observations

CODE

Source Visual

```
---
title: "Final Project"
output: html_notebook
---

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

Import Libraries

```{r}
install.packages("klaR")
install.packages("naivebayes")
install.packages("caret")
```

```{r}
library(naivebayes)
library(klaR)
library(rpart)
library(rpart.plot)
library(caret)
```
```

Import Data Set

```{r}
set.seed(1234)
data.frame(quakes)
summary(quakes)
str(quakes)
head(quakes)
tail(quakes)
dim(quakes)
```

GRAPHS

```{r}
#BARPLOT
magnitude <- quakes$mag
table_of_mags <- table(magnitude)
barplot(table_of_mags,col = "skyblue", main = "Earthquake Magnitudes", xlab = "Magnitude", ylab = "Number of Earthquakes")

#PIECHART
proportion_table <- prop.table(table_of_mags)
pie(proportion_table, col = rainbow(length(proportion_table)),main="Proportions of earthquakes")

#DOTPLOT
plot(magnitude, xlab="Number of earthquakes", ylab="Magnitude", main="Number of earthquakes vs. magnitude")

#BOXPLOT
boxplot(quakes$mag,col="lightblue",main="Boxplot of Earthquake Magnitudes",ylab="Magnitude")

```

CONFUSION MATRIX, DECISION TREES

```{r}
set.seed(123)
data(quakes)

#target variable
quakes$mag_cat <- cut(quakes$mag, breaks = c(-Inf, 4.5, 5.0, Inf), labels = c("Low", "Medium", "High"))

train_idx <- sample(seq_len(nrow(quakes)), size = 0.7 * nrow(quakes))
train_data <- quakes[train_idx, ]
test_data <- quakes[-train_idx, ]
#classification tree
quake_tree <- rpart(mag_cat ~ lat + long + depth + stations, data = train_data, method = "class")
rpart.plot(quake_tree)
pred <- predict(quake_tree, test_data, type = "class")
# Confusion matrix
confusionMatrix(pred, test_data$mag_cat)
```

PRUNING TREE

```{r}
printcp(quake_tree)
best_cp <- quake_tree$cptable[which.min(quake_tree$cptable[,"xerror"]), "CP"]

pruned_quake_tree <- prune(quake_tree, cp = best_cp)
rpart.plot(pruned_quake_tree)
pruned_pred <- predict(pruned_quake_tree, test_data, type = "class")

# Confusion matrix for pruned tree
confusionMatrix(pruned_pred, test_data$mag_cat)
```

Ten fold cross validation

```{r}
cv_ctrl <- trainControl(method = "cv", number = 10)

cv_tree <- train(mag_cat ~ lat + long + depth + stations,
                 data = train_data,
                 method = "rpart",
                 trControl = cv_ctrl)

print(cv_tree)
rpart.plot(cv_tree$finalModel)

cv_pred <- predict(cv_tree, test_data)

#Confusion matrix
confusionMatrix(cv_pred, test_data$mag_cat)
```

NAIVE BAYES

```{r}
library(klaR)
library(naivebayes)
library(caret)

nb_model <- NaiveBayes(mag_cat ~ lat + long + depth + stations, data =
train_data)

nb_pred <- predict(nb_model, test_data)
confusionMatrix(nb_pred$class, test_data$mag_cat)

nb_cv <- train(mag_cat ~ lat + long + depth + stations, data = train_data,
method = "naive_bayes", trControl = cv_ctrl)

print(nb_cv)
```