ResearchQ-s-STAT515-SM.R

smann

2024-05-01

# Research Question - 1  
# Load required libraries  
library(randomForest)

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

library(caret)

## Loading required package: ggplot2

##   
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':  
##   
## margin

## Loading required package: lattice

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ lubridate 1.9.3 ✔ tibble 3.2.1  
## ✔ purrr 1.0.2 ✔ tidyr 1.3.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::combine() masks randomForest::combine()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ purrr::lift() masks caret::lift()  
## ✖ ggplot2::margin() masks randomForest::margin()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

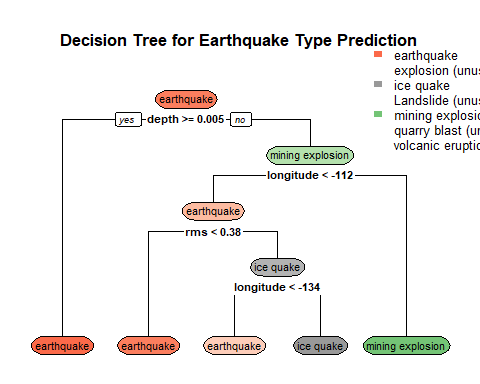
library(rpart)  
library(rpart.plot)  
  
# Read the earthquake data  
earthquakes <- read.csv("Global\_Seismic\_2023.csv")  
  
# Convert the 'type' column to a factor  
earthquakes$type <- as.factor(earthquakes$type)  
  
# Select relevant features  
features <- earthquakes %>%  
 select(latitude, longitude, depth, mag, magType, nst, gap, dmin, rms)  
  
# Split data into training and test sets  
set.seed(123)  
train\_index <- createDataPartition(earthquakes$type, p = 0.8, list = FALSE)  
train\_data <- earthquakes[train\_index, ]  
test\_data <- earthquakes[-train\_index, ]  
  
# Random Forest Model  
rf\_model <- randomForest(type ~ ., data = train\_data[, c("type", names(features))])  
rf\_predictions <- predict(rf\_model, newdata = test\_data[, names(features)])  
rf\_accuracy <- mean(rf\_predictions == test\_data$type)  
print(paste("Random Forest Accuracy:", rf\_accuracy))

## [1] "Random Forest Accuracy: 0.99939197405756"

# Decision Tree Model  
dt\_model <- rpart(type ~ ., data = train\_data[, c("type", names(features))])  
dt\_predictions <- predict(dt\_model, newdata = test\_data[, names(features)], type = "class")  
dt\_accuracy <- mean(dt\_predictions == test\_data$type)  
print(paste("Decision Tree Accuracy:", dt\_accuracy))

## [1] "Decision Tree Accuracy: 0.99939197405756"

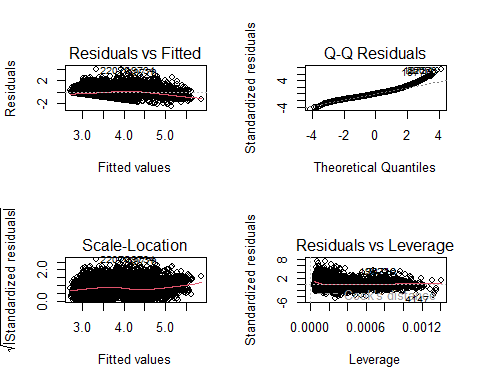
# Visualize the decision tree  
rpart.plot(dt\_model, main = "Decision Tree for Earthquake Type Prediction", extra = 0)



# Research Question - 2  
# Fit a multiple linear regression model  
lm\_model <- lm(mag ~ latitude + longitude + depth + rms, data = earthquakes)  
  
# Summary of the regression model  
summary(lm\_model)

##   
## Call:  
## lm(formula = mag ~ latitude + longitude + depth + rms, data = earthquakes)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.4250 -0.3630 -0.0397 0.3058 3.8648   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.641e+00 9.188e-03 396.276 < 2e-16 \*\*\*  
## latitude -1.134e-02 1.203e-04 -94.267 < 2e-16 \*\*\*  
## longitude 2.266e-03 2.727e-05 83.109 < 2e-16 \*\*\*  
## depth 1.222e-04 3.049e-05 4.008 6.15e-05 \*\*\*  
## rms 9.743e-01 1.351e-02 72.105 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5281 on 24677 degrees of freedom  
## Multiple R-squared: 0.5777, Adjusted R-squared: 0.5777   
## F-statistic: 8441 on 4 and 24677 DF, p-value: < 2.2e-16

# Diagnostic plots  
par(mfrow = c(2, 2))   
plot(lm\_model)



#Research Question - 3  
  
# Filter for only earthquake events  
earthquakes\_only <- earthquakes %>%  
 filter(type == "earthquake")  
  
# Select relevant variables for clustering  
clustering\_vars <- earthquakes %>%  
 select(latitude, longitude, depth, mag)  
  
# Check for missing values  
summary(clustering\_vars)

## latitude longitude depth mag   
## Min. :-65.850 Min. :-180.00 Min. : -3.37 Min. :2.600   
## 1st Qu.: -5.919 1st Qu.:-150.99 1st Qu.: 10.00 1st Qu.:3.180   
## Median : 19.173 Median : -66.17 Median : 22.00 Median :4.200   
## Mean : 18.001 Mean : -15.67 Mean : 66.88 Mean :3.968   
## 3rd Qu.: 43.811 3rd Qu.: 126.82 3rd Qu.: 66.34 3rd Qu.:4.500   
## Max. : 86.594 Max. : 180.00 Max. :681.24 Max. :7.800

# Apply k-means clustering to the selected variables  
set.seed(123) # Set a seed for reproducibility  
K <- kmeans(clustering\_vars, centers = 20, nstart = 25)  
  
# Print the cluster assignments for the first 5 earthquakes  
head(K$cluster, n = 5)

## [1] 14 12 11 12 20

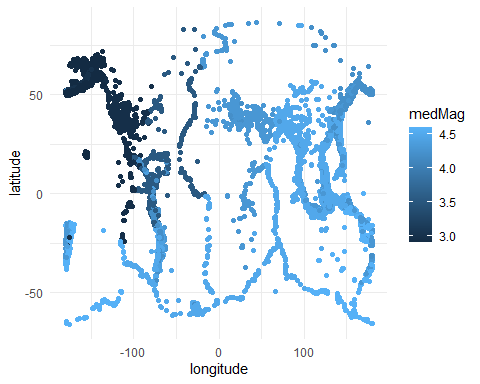
# Print the number of earthquakes in each cluster  
K$size

## [1] 2373 2336 179 275 1230 921 444 1149 841 798 2440 1646 962 977 1060  
## [16] 1476 649 157 536 4233

# Print the cluster centers (mean values of each variable for each cluster)  
K$centers

## latitude longitude depth mag  
## 1 34.946368 -111.86282 7.870429 3.178070  
## 2 12.845242 132.67965 15.512655 4.572175  
## 3 15.427472 138.39782 381.023821 4.308939  
## 4 -9.770567 156.24677 544.308840 4.473091  
## 5 -20.485684 162.63908 14.921506 4.640650  
## 6 -28.110834 -172.77426 21.197208 4.678176  
## 7 -17.834266 -98.18502 205.195635 4.410045  
## 8 12.327894 138.17273 128.830642 4.311662  
## 9 46.788105 -164.61452 126.869313 3.252081  
## 10 -41.660859 -48.37372 24.974477 4.592732  
## 11 19.397657 -65.62455 21.001639 3.666721  
## 12 5.045346 129.13308 63.463708 4.557412  
## 13 -4.076842 -68.18922 107.713308 4.178399  
## 14 48.272523 161.58100 39.294028 3.948209  
## 15 13.793635 83.69157 14.615772 4.518585  
## 16 34.457498 34.61538 12.194370 4.397900  
## 17 -20.205088 -175.98997 555.734590 4.361171  
## 18 -20.956380 -177.24759 384.672236 4.359236  
## 19 9.110910 130.32453 212.341890 4.320709  
## 20 51.309052 -163.09795 25.546071 3.060638

# Create a data frame with the original data and the assigned cluster  
df <- data.frame(earthquakes, cluster = K$cluster)  
  
# Add the median magnitude for each cluster  
df <- group\_by(df, cluster) %>%  
 mutate(medMag = median(mag)) %>%  
 ungroup()  
  
# Visualize the clusters on a map  
ggplot(df) +  
 geom\_point(aes(x = longitude, y = latitude, color = medMag)) +  
 theme\_minimal()



# Write data frame "df" to a csv file to use in Tableau  
write.csv(df, "NewGlobalSeismic2023.csv")