Earthquake Perception Classification

Sayak Goswami and Santanu Biswas

November 21, 2024

Abstract

The dataset collects data from a survey conducted by **Antonio Cola and Rosario Urso**, who hold master's degrees in Statistical Sciences for **Decision Making** from the University of Naples Federico II. The study analyzes individuals' perceptions of seismic events, with particular reference to the seismic activity in the Campi Flegrei area, which occurred between October 31 and November 16, 2023.

The aim of the study was to examine how different groups of people perceive and react to seismic events, identifying key components of **"Earthquake Stress"** and assessing differences in perception and response based on variables such as gender, employment status, income, marital status, and education level.

The Dataset

The dimension of the dataset is:-

```
## [1] 472  43
```

The Columns of the dataset along with their data types are as follows :-

```
## 'data.frame': 472 obs. of  43 variables:
##  $ age                           : int  21 21 21 23 64 22 26 22 40 42 ...
##  $ sex                           : chr  "Male" "Female" "Female" "Male" ...
##  $ marital_status                : chr  "Unmarried" "Unmarried" "Unmarried" "Unmarried" ...
##  $ residence                     : chr  "Napoli" "Napoli" "Napoli" "Quarto" ...
##  $ education                     : chr  "Upper Secondary School" "Upper Secondary School" "Upper Seconda
##  $ occupation                    : chr  "Student" "Student" "Student Worker" "Student" ...
##  $ out_of_region_employment      : chr  "No" "No" "No" "No" ...
##  $ family_members                : chr  "4" "5" "5" "5" ...
##  $ family_disabilities           : chr  "No" "No" "No" "No" ...
##  $ house_floor                   : chr  "1" "4" "4" "2" ...
##  $ earthquake80                  : chr  "No" "No" "No" "No" ...
##  $ political_orientation         : chr  "Left" "Center" "Left" "None" ...
##  $ shocks                        : int  3 1 3 3 4 2 2 1 2 1 ...
##  $ fear                          : int  2 2 2 3 4 3 3 3 4 1 ...
##  $ anxiety                       : int  2 2 2 3 4 4 3 3 5 2 ...
##  $ physiological_symptoms        : int  2 1 1 3 3 1 1 1 5 1 ...
##  $ decision_timeliness           : int  3 3 4 1 2 3 2 4 2 2 ...
##  $ insomnia                      : int  1 2 2 3 3 3 3 1 3 1 ...
##  $ seismic_concern               : int  2 4 3 4 4 4 3 3 5 4 ...
##  $ abroad                        : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ out_of_region                 : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ out_of_region_earthquake      : chr  "No" "No" "Yes" "Yes" ...
##  $ change_of_residence           : chr  "No" "Yes" "Yes" "Yes" ...
##  $ change_of_residence_earthquake: chr  "No" "No" "Yes" "Yes" ...
##  $ red_zone_frequency            : chr  "No" "Yes" "Yes" "Yes" ...
##  $ radio_info                    : int  3 3 2 1 3 3 3 2 4 4 ...
##  $ TV_info                       : int  4 2 2 1 3 3 3 2 4 4 ...
##  $ social_media_info             : int  2 2 2 1 3 3 3 3 4 3 ...
##  $ newspaper_info                : int  4 2 2 1 3 3 3 2 3 3 ...
##  $ app_info                      : int  4 2 2 1 3 3 3 4 4 3 ...
##  $ municipal_institutions_trust  : int  3 2 2 1 2 3 1 1 2 2 ...
##  $ regional_institutions_trust   : int  3 2 2 1 3 3 1 1 2 1 ...
##  $ national_institutions_trust   : int  3 2 2 1 2 3 3 3 3 3 ...
##  $ INGV_trust                    : int  4 3 5 1 4 3 3 4 5 4 ...
##  $ security                      : int  2 1 3 1 2 3 1 2 1 1 ...
```

```
##  $ reception_centers             : chr  "No" "No" "Yes" "No" ...
##  $ property_house                : chr  "Yes" "Yes" "Yes" "No" ...
##  $ housing_type                  : chr  "Apartment" "Other" "Apartment" "Apartment" ...
##  $ elevator                      : chr  "Yes" "Yes" "Yes" "No" ...
##  $ n_vehicles                    : int  0 5 0 2 0 2 3 2 2 2 ...
##  $ vehicle_type                  : chr  "None" "Car and Motorcycle" "None" "Car" ...
##  $ end_of_month                  : int  1 1 1 5 3 2 1 1 1 2 ...
##  $ salary                        : chr  "Up to €15000" "Between €28000 and €50000" "Between €15000 and
```

Here, we observe that, many columns should have the data type "factor" that is categorical data. We shall convert them accordingly.

These are the columns that are to be converted to categorical :-

```
##  [1] "sex"                       "marital_status"
##  [3] "residence"                 "education"
##  [5] "occupation"                "out_of_region_employment"
##  [7] "family_members"            "house_floor"
##  [9] "insomnia"                  "decision_timeliness"
## [11] "seismic_concern"           "abroad"
## [13] "out_of_region"             "out_of_region_earthquake"
## [15] "change_of_residence"       "INGV_trust"
## [17] "security"                  "reception_centers"
## [19] "property_house"            "elevator"
## [21] "vehicle_type"
```

Handling NA values

The total number of NA values are :-

```
## age
##  16
```

We shall remove the rows containing NA values

After imputing the NA values with median, we also observe that there are "" in many column which implies missing values.

The number of empty values in columns are :-

```
##                salary political_orientation   family_disabilities
##                   160                   151                     8
##             residence                   sex
##                     4                     2
```

- The number of empty cells in "salary" and "political orientation" is very high, so the only way to handle those is to drop the columns.
- The "family_disabilities" column has 8 empty cells, as it is categorical we can just impute the mode of the data in the missing places or we can drop the rows. Dropping the rows might lead to data loss from other columns hence imputing the mode in those rows seems a better option.
- The "sex" and "residence" columns has very less rows with empty cells, so to avoid random generalization in such cases we shall drop the corresponding rows.

After performing the above mentioned steps, the dimension of the data is :-

```
## [1] 450  41
```

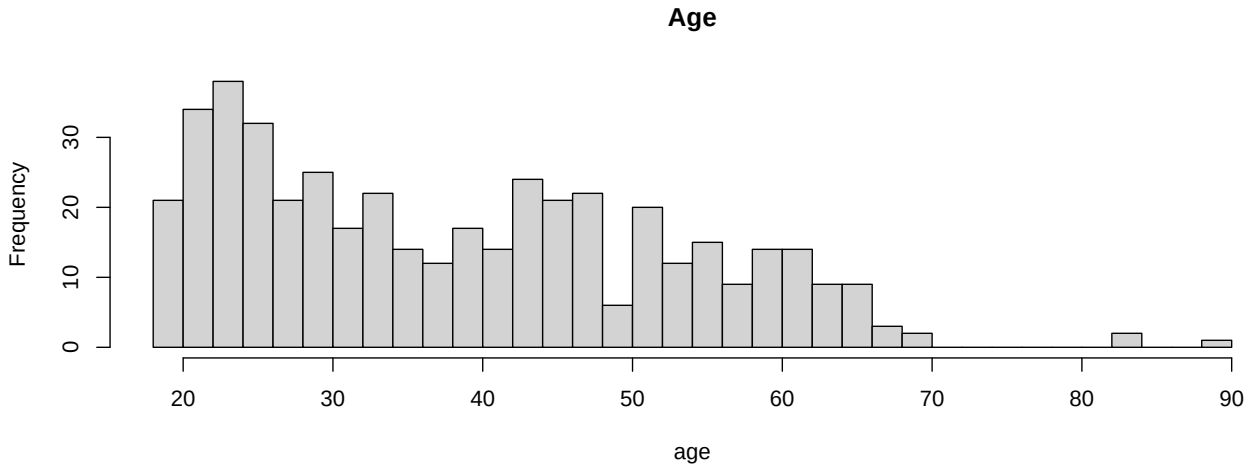
Exploratory Data Analysis

The columns of the dataset are mostly categorical, so we shall make some plots that might lead to some meaningful insights.

In the dataset age is the only column that contains **numerical data** and rest of them are basically categorical variables with different number of levels.
So summary statistics can only be provided for age variable.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   25.00   37.00   38.78   49.75   90.00
```

From the summary statistics value we can note that **MODE>MEDIAN>MEAN** hence the distribution of age is positively skewed.Plotting the histogram can provide a better idea regarding it.



Note that our objective is to infer about Earthquake stress so we consider seismic_concern as our variable of interest.Then we should try to establish some relationship between our response variable with remaining variables.For

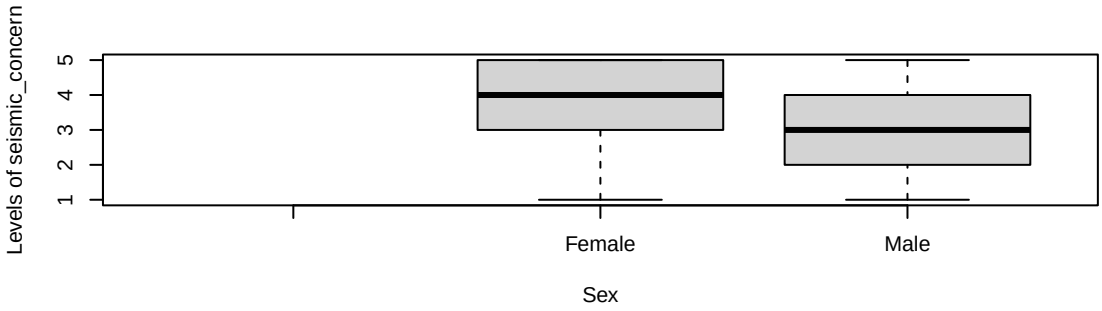
this task we construct tables of responses and use boxplots as well.

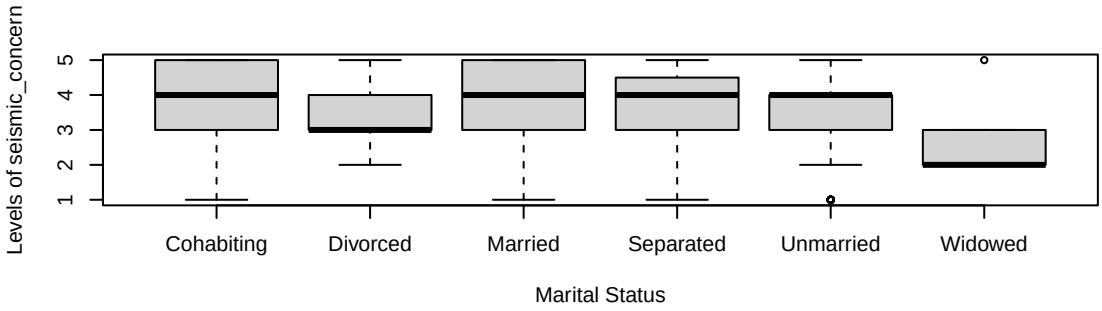
Shortcomings

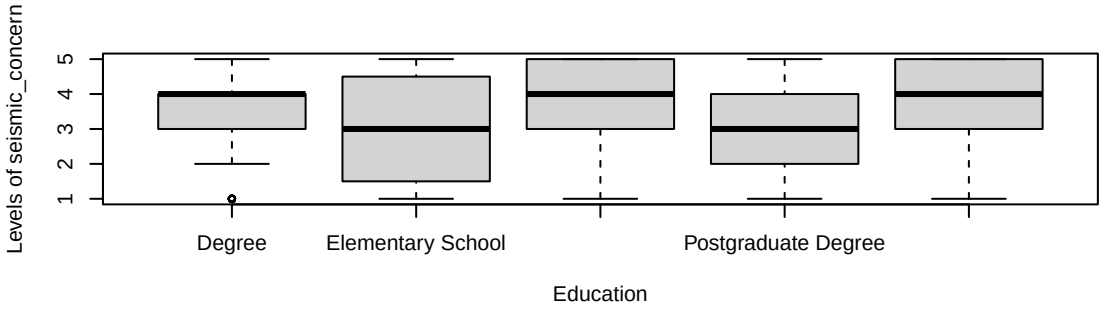
```
seismic_concern vs sex:

    Female Male
  1     19   13
  2     25   21
  3     99   19
  4    119   24
  5    101   10
seismic_concern vs marital_status:

    Cohabiting Divorced Married Separated Unmarried Widowed
  1          4        0       7         1        20       0
  2          3        1      15         0        24       3
  3         16        6      34         4        57       1
  4         14        3      52         3        71       0
  5         22        1      42         3        42       1
seismic_concern vs education:

    Degree Elementary School Lower Secondary School Postgraduate Degree
  1     17                 1                      2                   5
  2     12                 1                      1                  11
  3     59                 0                      6                  17
  4     67                 1                      6                  18
  5     39                 1                     12                   9

    Upper Secondary School
  1                      7
  2                     21
  3                     36
  4                     51
  5                     50
seismic_concern vs occupation:

    Employee Retired Self-Employed Student Student Worker Unemployed
  1        8       1             6      13              2          2
  2        8       3            12      13              4          6
  3       44       1            23      30              3         17
  4       64       4            17      28             11         19
  5       40       4            20      21              2         24
```

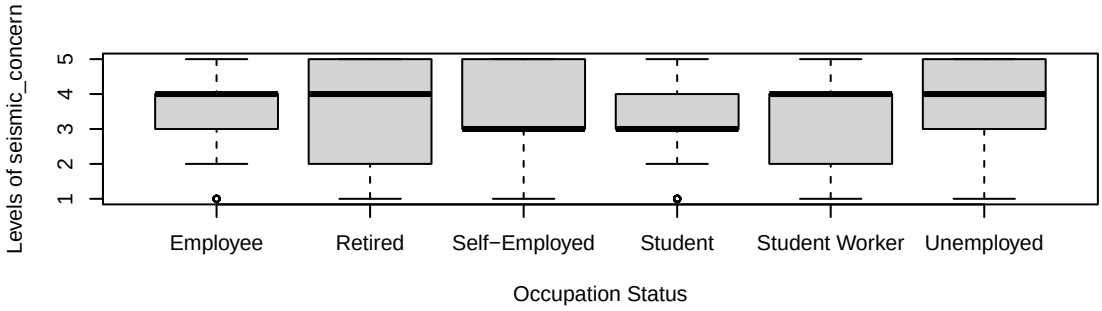
To visualize the outcomes, we shall take help of boxplots on each of the mentioned columns.

The corresponding boxplots are:-

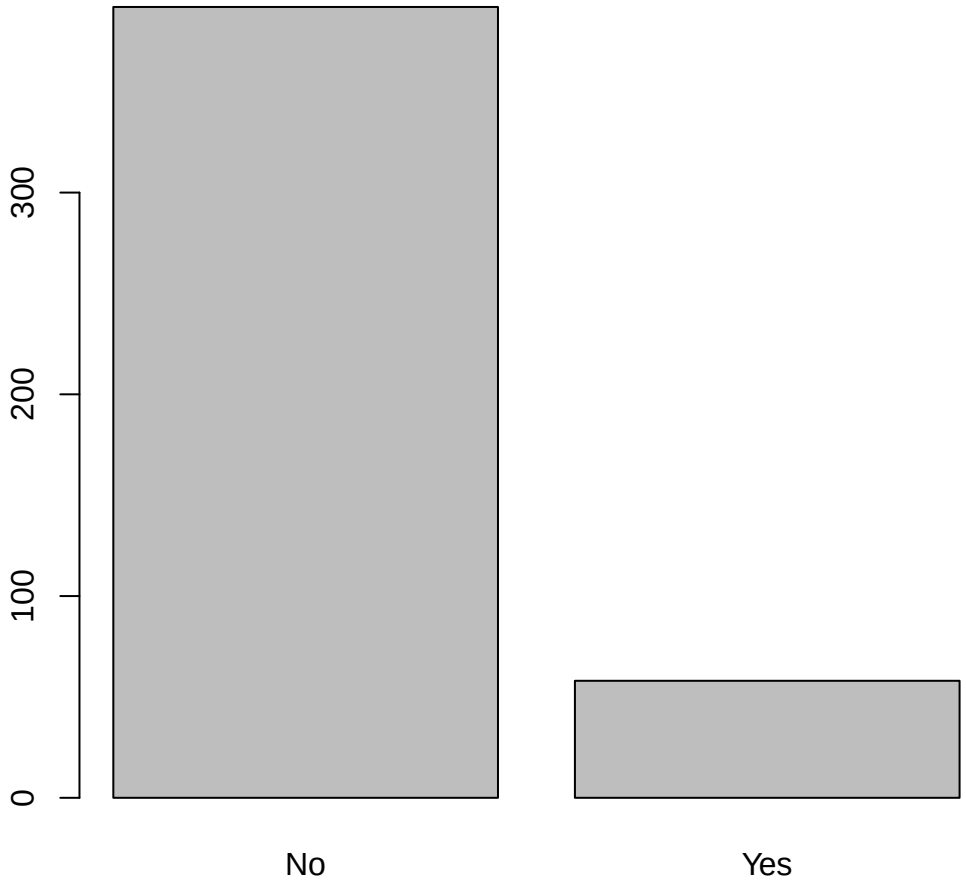








```
seismic_concern vs out_of_region_employment:

 No Yes
392  58
```



Model Fitting

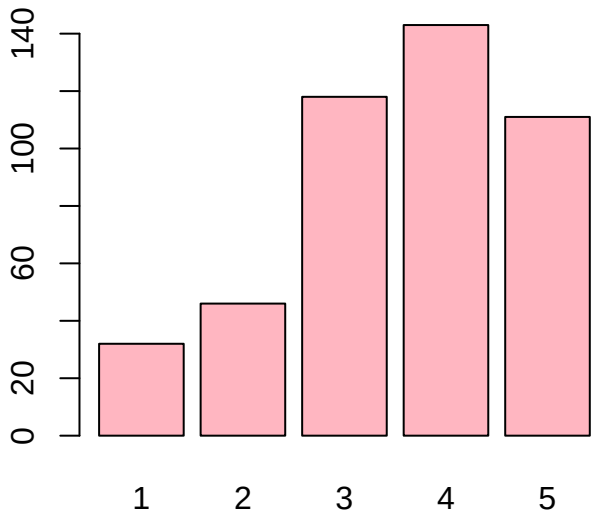
Here, our main objective is to classify the levels of "seismic concern"

The response variable is seismic_concern which has 5 levels (1,2,3,4,5).

Firstly we need to split the dataset into training and testing data for final Performance Metrics judgement.

Note: We shall observe that **"residence"** is a string type variable and hence not required for our model so we shall choose to drop it.

Before splitting the dataset we need to check whether the seismic concern column is balanced or not:-



Clearly the dataset is "**Imbalanced**".

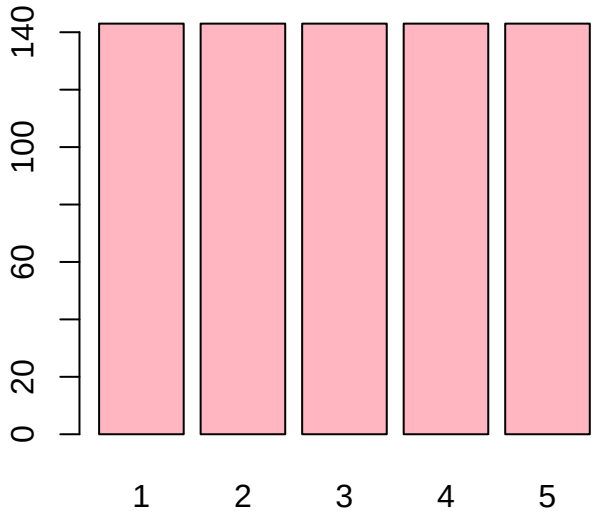
We shall use the **SMOTE** family of functions to make the column "seismic_concern" (target variable) balanced

```
## Warning:  package 'themis' was built under R version 4.4.1
## Loading required package:  recipes
## Loading required package:  dplyr
## Warning:  package 'dplyr' was built under R version 4.4.1
##
## Attaching package:  'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
##
## Attaching package:  'recipes'
## The following object is masked from 'package:stats':
##
##     step

## [1] 450  40
```

The dimension of the dataset after bootstrapping new samples is :-

```
## [1] 715  40
```

Lets confirm again whether the dataset is balanced or not :-



Now we are going to split the data set into training data and testing data

The dimension of the training data :-

```
## [1] 500  40
```

The dimension of the testing data :-

```
## [1] 215  40
```

In the given data, we observe that there are 41 columns that implies 40 predictor variables

Classification Model

Here the response variable is "seismic_concern" which has 5 levels hence we have several choices of classification algorithms to be considered :-

- Multinomial Logistic Regression
- Decision tree
- Random Forest
- Naive Bayes

General Steps for model fitting and prediction :-

- we shall consider each model separately
- If possible we will try to reduce the dimension of the predictors
- Check the accuracy score when we use this model to predict the test data
- Suggest the subset of covariates which has a better accuracy vs complexity trade off

Multinomial Logistic Regression:

```
## Loading required package: ggplot2
## Warning: package 'ggplot2' was built under R version 4.4.1
## Loading required package: lattice
## Warning: package 'nnet' was built under R version 4.4.1
## # weights: 385 (304 variable)
## initial value 804.718956
## iter 10 value 555.954045
## iter 20 value 328.841864
## iter 30 value 269.999576
## iter 40 value 242.422058
## iter 50 value 216.920499
## iter 60 value 203.038110
## iter 70 value 197.210347
## iter 80 value 193.499606
## iter 90 value 188.659059
## iter 100 value 183.960059
## final value 183.960059
## stopped after 100 iterations
## Time difference of 0.1608689 secs
```

The accuracy score of this model is :-

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2   3   4   5
##          1 102   0   1   1   0
##          2   1  87   5   3   0
##          3   0   5  85  11   2
##          4   0   0  10  77  15
##          5   0   0   3   9  83
```

```
##
## Overall Statistics
##
##                Accuracy : 0.868
##                  95% CI : (0.8351, 0.8964)
##     No Information Rate : 0.208
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.835
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity            0.9903   0.9457   0.8173   0.7624   0.8300
## Specificity            0.9950   0.9779   0.9545   0.9373   0.9700
## Pos Pred Value         0.9808   0.9062   0.8252   0.7549   0.8737
## Neg Pred Value         0.9975   0.9876   0.9521   0.9397   0.9580
## Prevalence             0.2060   0.1840   0.2080   0.2020   0.2000
## Detection Rate         0.2040   0.1740   0.1700   0.1540   0.1660
## Detection Prevalence   0.2080   0.1920   0.2060   0.2040   0.1900
## Balanced Accuracy      0.9926   0.9618   0.8859   0.8499   0.9000
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3  4  5
##          1 33  3  4  3  1
##          2  1 40 11  3  0
##          3  6  5 14  8  1
##          4  0  2  5 19 18
##          5  0  1  5  9 23
##
## Overall Statistics
##
##                Accuracy : 0.6
##                  95% CI : (0.5312, 0.666)
##     No Information Rate : 0.2372
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.4983
##
##  Mcnemar's Test P-Value : 0.1246
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity            0.8250   0.7843  0.35897  0.45238   0.5349
## Specificity            0.9371   0.9085  0.88636  0.85549   0.9128
## Pos Pred Value         0.7500   0.7273  0.41176  0.43182   0.6053
## Neg Pred Value         0.9591   0.9313  0.86188  0.86550   0.8870
## Prevalence             0.1860   0.2372  0.18140  0.19535   0.2000
## Detection Rate         0.1535   0.1860  0.06512  0.08837   0.1070
## Detection Prevalence   0.2047   0.2558  0.15814  0.20465   0.1767
## Balanced Accuracy      0.8811   0.8464  0.62267  0.65394   0.7238
```

```
## Time difference of 0.1608689 secs
```

Training Accuracy: 1
Testing Accuracy: 0.6419
This is a case of **Overfitting** of the model

Decision Tree:

```
## Warning:  package 'rpart' was built under R version 4.4.1
## Warning:  package 'tree' was built under R version 4.4.1
## Warning:  package 'randomForest' was built under R version 4.4.1
## randomForest 4.7-1.2
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package:  'randomForest'
## The following object is masked from 'package:ggplot2':
##
##     margin
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## Warning in tree(seismic_concern ~ ., data = train):  NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by coercion
## Warning in pred1.tree(tree, tree.matrix(nd)):  NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by coercion
## Warning in pred1.tree(tree, tree.matrix(nd)):  NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by coercion
## Warning in pred1.tree(tree, tree.matrix(nd)):  NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by coercion
## Warning in pred1.tree(tree, tree.matrix(nd)):  NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by coercion
## Warning in pred1.tree(tree, tree.matrix(nd)):  NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by coercion
## Warning in pred1.tree(tree, tree.matrix(nd)):  NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by coercion
## Warning in pred1.tree(tree, tree.matrix(nd)):  NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by coercion
## Warning in pred1.tree(tree, tree.matrix(nd)):  NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by coercion
## Warning in pred1.tree(tree, tree.matrix(nd)):  NAs introduced by coercion
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by coercion
## Warning in pred1.tree(tree, tree.matrix(nd)):  NAs introduced by coercion
## Time difference of 0.001743078 secs

## Warning in pred1.tree(object, tree.matrix(newdata)):  NAs introduced by coercion

## [1] "Training Accuracy"
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  1  2  3  4  5
##          1 91  8 10  2  0
##          2  3 63 17  2  0
##          3  8 19 40 14  6
```

```
##          4  1  1 36 70 33
##          5  0  1  1 13 61
##
## Overall Statistics
##
##                Accuracy : 0.65
##                  95% CI : (0.6064, 0.6918)
##     No Information Rate : 0.208
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5622
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity            0.8835   0.6848   0.3846   0.6931   0.6100
## Specificity            0.9496   0.9461   0.8813   0.8221   0.9625
## Pos Pred Value         0.8198   0.7412   0.4598   0.4965   0.8026
## Neg Pred Value         0.9692   0.9301   0.8450   0.9136   0.9080
## Prevalence             0.2060   0.1840   0.2080   0.2020   0.2000
## Detection Rate         0.1820   0.1260   0.0800   0.1400   0.1220
## Detection Prevalence   0.2220   0.1700   0.1740   0.2820   0.1520
## Balanced Accuracy      0.9166   0.8154   0.6330   0.7576   0.7863
## [1] "Testing Accuracy"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3  4  5
##          1 36  3  4  4  1
##          2  2 38  8  2  1
##          3  2 10 10 10  2
##          4  0  0 14 20 12
##          5  0  0  3  6 27
##
## Overall Statistics
##
##                Accuracy : 0.6093
##                  95% CI : (0.5406, 0.6749)
##     No Information Rate : 0.2372
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.5105
##
##  Mcnemar's Test P-Value : 0.288
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity            0.9000   0.7451  0.25641  0.47619   0.6279
## Specificity            0.9314   0.9207  0.86364  0.84971   0.9477
## Pos Pred Value         0.7500   0.7451  0.29412  0.43478   0.7500
## Neg Pred Value         0.9760   0.9207  0.83978  0.86982   0.9106
## Prevalence             0.1860   0.2372  0.18140  0.19535   0.2000
## Detection Rate         0.1674   0.1767  0.04651  0.09302   0.1256
```

```
## Detection Prevalence   0.2233   0.2372  0.15814  0.21395   0.1674
## Balanced Accuracy      0.9157   0.8329  0.56002  0.66295   0.7878
```

```
## Time difference of 0.001743078 secs
```

Training Accuracy: 0.704
Testing Accuracy: 0.6419

Random Forest

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3  4  5
##          1 94  1  9  3  1
##          2  4 80 10  2  0
##          3  4  9 60 24  6
##          4  1  1 23 52 21
##          5  0  1  2 20 72
##
## Overall Statistics
##
##                Accuracy : 0.716
##                  95% CI : (0.6743, 0.7551)
##     No Information Rate : 0.208
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6449
##
##  Mcnemar's Test P-Value : 0.5175
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity            0.9126   0.8696   0.5769   0.5149   0.7200
## Specificity            0.9647   0.9608   0.8914   0.8847   0.9425
## Pos Pred Value         0.8704   0.8333   0.5825   0.5306   0.7579
## Neg Pred Value         0.9770   0.9703   0.8892   0.8781   0.9309
## Prevalence             0.2060   0.1840   0.2080   0.2020   0.2000
## Detection Rate         0.1880   0.1600   0.1200   0.1040   0.1440
## Detection Prevalence   0.2160   0.1920   0.2060   0.1960   0.1900
## Balanced Accuracy      0.9387   0.9152   0.7342   0.6998   0.8313
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3  4  5
##          1 37  2  2  3  0
##          2  0 45  3  2  1
##          3  2  3 26  9  1
##          4  1  1  7 22 12
##          5  0  0  1  6 29
##
## Overall Statistics
##
```

```
##                Accuracy : 0.7395
##                  95% CI : (0.6755, 0.7969)
##     No Information Rate : 0.2372
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6738
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity            0.9250   0.8824   0.6667   0.5238   0.6744
## Specificity            0.9600   0.9634   0.9148   0.8786   0.9593
## Pos Pred Value         0.8409   0.8824   0.6341   0.5116   0.8056
## Neg Pred Value         0.9825   0.9634   0.9253   0.8837   0.9218
## Prevalence             0.1860   0.2372   0.1814   0.1953   0.2000
## Detection Rate         0.1721   0.2093   0.1209   0.1023   0.1349
## Detection Prevalence   0.2047   0.2372   0.1907   0.2000   0.1674
## Balanced Accuracy      0.9425   0.9229   0.7907   0.7012   0.8169
```

```
## Time difference of 1.162388 secs
```

Training Accuracy: 0.708
Testing Accuracy: 0.7488

Naive Bayes

```
## Time difference of 0.008970976 secs
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3  4  5
##          1 90 11 10  2  2
##          2  4 62 17  4  1
##          3  8 17 54 17  6
##          4  1  0 20 60 11
##          5  0  2  3 18 80
##
## Overall Statistics
##
##                Accuracy : 0.692
##                  95% CI : (0.6495, 0.7322)
##     No Information Rate : 0.208
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6147
##
##  Mcnemar's Test P-Value : 0.2188
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity            0.8738   0.6739   0.5192   0.5941   0.8000
```

```
## Specificity            0.9370   0.9363   0.8788   0.9198   0.9425
## Pos Pred Value         0.7826   0.7045   0.5294   0.6522   0.7767
## Neg Pred Value         0.9662   0.9272   0.8744   0.8995   0.9496
## Prevalence             0.2060   0.1840   0.2080   0.2020   0.2000
## Detection Rate         0.1800   0.1240   0.1080   0.1200   0.1600
## Detection Prevalence   0.2300   0.1760   0.2040   0.1840   0.2060
## Balanced Accuracy      0.9054   0.8051   0.6990   0.7569   0.8713
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  3  4  5
##          1 37  2  3  4  0
##          2  1 36  4  1  0
##          3  1 13 21  6  1
##          4  1  0  9 21 13
##          5  0  0  2 10 29
##
## Overall Statistics
##
##                Accuracy : 0.6698
##                  95% CI : (0.6026, 0.7322)
##     No Information Rate : 0.2372
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5874
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity            0.9250   0.7059  0.53846  0.50000   0.6744
## Specificity            0.9486   0.9634  0.88068  0.86705   0.9302
## Pos Pred Value         0.8043   0.8571  0.50000  0.47727   0.7073
## Neg Pred Value         0.9822   0.9133  0.89595  0.87719   0.9195
## Prevalence             0.1860   0.2372  0.18140  0.19535   0.2000
## Detection Rate         0.1721   0.1674  0.09767  0.09767   0.1349
## Detection Prevalence   0.2140   0.1953  0.19535  0.20465   0.1907
## Balanced Accuracy      0.9368   0.8346  0.70957  0.68353   0.8023
```

```
## Time difference of 0.008970976 secs
```

Training Accuracy: 0.698
Testing Accuracy: 0.6512