Classification and Analysis of Real-World Earthquake Data Using Various Machine Learning Algorithms

Manka Vasti and Amita Dev

Abstract The research paper describes the implementation of different machine learning algorithms used for classification. The aim is to provide the interested learner about the implementation of the classification algorithms on the real-world earthquake dataset. This study has been done using support vector machine (SVM), K-nearest neighbor (KNN), random forest (RF), and Naïve Bayes (NB) algorithms in R programming language. The result of the study is the analysis and classification of each data value in the dataset and hence assigning it to the correct class label. The study is done on the dataset collected from the Web site of Indian Meteorological Department, Ministry of Earth Sciences, Government of India. The methodology includes data acquisition and description, feature selection, data normalization, data partitioning, model implementation and prediction, optimization and fine-tuning, classification and finding rate of accuracy, and misclassification error. The research work also compares the accuracy of the algorithms used by comparing the accuracy of classification of data and their misclassification error. The research paper also provides details of confusion matrix that lists statistics of accuracy, sensitivity, and specificity in order to make the correct decision about the selection of the most accurate algorithm on the provided dataset.

Keywords Machine learning · Classification algorithms · Naïve Bayes · Supervised learning · Support vector machine · Random forest · K-nearest neighbor · Earthquakes

M. Vasti (✉)
G D Goenka University, Gurgaon, Haryana 122103, India
e-mail: mankavasti@hotmail.com

A. Dev
Indira Gandhi Delhi Technical University for Women, Delhi 110006, India
e-mail: amita_dev@hotmail.com

V. Jain et al. (eds.), *Advances in Data Sciences, Security and Applications*,
Lecture Notes in Electrical Engineering 612,
https://doi.org/10.1007/978-981-15-0372-6_1

1 Introduction

Every year, there are many earthquakes witnessed all across the world. It is a natural phenomenon and affects all parts of earth. The magnitude of the earthquakes may vary which categorizes them into various classes ranging from nonsignificant ones to the disastrous ones [1, 2]. The event classification can be done based on long period (LP) earthquakes, shallow earthquakes, deep earthquakes [3]. Also, some online resources classified earthquakes based on the magnitude as given in Table 1 [1, 2].

A pool of earthquake dataset is collected using seismograms and other equipments worldwide [4]. Data mining is the process to extract meaningful patterns from a pool of raw dataset. Hence, it is also sometimes termed as knowledge mining. It analyzes large amount of data to extract meaningful information out from the dataset. There are many free software tools available for general data mining such as RapidMiner, R, Weka, KNIME, Orange, and scikit-learn [5]. Data mining techniques include association, classification, clustering [6], and genetic algorithms [5]. Machine learning is a subfield of data science that focuses on supervised and unsupervised learning methods to do the predictions. Classification is the process of assigning a class label to the data values in the dataset. A study shows that within a period of ten years, the use of support vector machine and random forest classification algorithms has increased significantly [7]. Indian Meteorological Department (IMD), Ministry of Earth Sciences, Government of India, collects the global observations in the parameterized format of climatological, marine meteorological, seismological data [8]. Indian Meteorological Department's Centre for Seismology is maintaining a countrywide National Seismological Network (NSN), consisting of a total of 82 seismological stations, spread over the entire length and breadth of the country [9]. The remaining part of this section describes the process of data acquisition and description, feature selection, data normalization, and data partitioning. Section 2 illustrates the model implementation, prediction, and optimization of the algorithms, namely K-nearest neighbor (KNN), random forest (RF), Naïve Bayes (NB), and support vector machine (SVM) and the comparison of their accuracy and misclassification error. Section 3 describes the conclusion of the study performed and the future scope of the work. This is followed by the list of references for the reader to refer for more details.

Table 1 Criteria to classify earthquakes based on their magnitude [1, 2]

Class	Magnitude
Great	8 or more
Major	7–7.9
Strong	6–6.9
Moderate	5–5.9
Light	4–4.9
Minor	3–3.9

1.1 Data Acquisition and Description

The seismological dataset available from Indian Meteorological Department (IMD) is parameterized dataset having relevant earthquake parameters such as date, time, longitude, latitude, depth (in kms), magnitude, and region [8]. Date and time indicate Indian Standard Time (IST) and Coordinated Universal Time (UTC) of the occurrence of the event. The longitude and the latitude give the exact location of the earthquake, whereas region provides with the name of the location of the occurrence of the earthquake worldwide. The parameters, namely depth and magnitude, indicate the intensity of the earthquake. The total dataset used in the research study is collected for duration of nearly 4.5 years, i.e., from January 1, 2013, till August 22, 2018, which comprises 1575 records. Table 2 shows a glimpse of the dataset used.

Nowadays, the latest available intensity observations for India provide a dataset for evaluating the location and magnitude of numerous earthquakes that have until now been amenable only to qualitative analysis and, in particular, permit us to assess attenuation throughout the subcontinent [10].

Table 2 Earthquake dataset collected from IMD, Delhi [4]

Date	Time (IST)	Lat.	Long.	Depth (km)	Magnitude	Region
22/8/2018	15:01:00	43.64° N	127.60° W	10	6.2	Oregon
22/8/2018	3:01:00	10.73° N	62.91° W	123	7.3	Near coast of Venezuela
22/8/2018	4:02:00	16.0° S	168.1° E	10	6.7	Vanuatu Islands
22/8/2018	3:01:00	10.8° N	62.9° W	10	7	Near coast of Venezuela
20/8/2018	7:49:04	30.1° N	79.8° E	10	3.6	Bageshwar, Uttarakhand
19/8/2018	20:26:25	8.4° S	116.6° E	10	6.7	Sumbawa region, Indonesia
19/8/2018	9:59:00	17.1° S	177.9° W	438	6.6	Fiji Islands region
19/8/2018	9:40:21	8.3° S	116.5° E	10	6.5	Sumbawa region, Indonesia
19/8/2018	7:48:53	18.3° S	178.3° W	630	6.2	Fiji Islands region
19/8/2018	5:49:38	18.2° S	178.0° W	586	8	Fiji Islands region
17/8/2018	21:05:02	7.5° S	119.8° E	547	6.2	Flores Sea

1.2 Feature Selection

This step includes selecting relevant features or attributes required for the research study. Feature selection is essential in data preprocessing as all the attributes may not be useful in the process of data mining. Dimensional reduction can be performed using various techniques. The most commonly used method includes removing the nonessential attributes from the dataset. The other method may be representation of thousands of attributes to their equivalent lesser number of attributes using methods such as principal component analysis (PCA). The present study simply selects two predictors, namely depth and magnitude, and computes the response variable in the name of a class label. The label classifies each data point or data value to the type of earthquake, i.e., great, major, strong, moderate, light, minor, and nonsignificant [1, 2].

1.3 Data Normalization

Data normalization is performed before the application of the algorithm on the dataset. This is required as the input parameters may have different types of ranges. Therefore, it is essential to get them within the range of 0–1. The equation used for data normalization is as follows:

$$\frac{(x - \min(x))}{(\max(x) - \min(x))} \tag{1}$$

1.4 Data Partitioning

Supervised learning is based on training the machine to learn from prior examples, i.e., the training dataset, and evaluate its performance by exposing it to a test dataset. Generally, a combination of two-thirds, i.e., 75% of the dataset, is used for training and one-third is used for the testing purpose. The research study has also followed the same standard and randomly distributed rows into a partition of two-thirds of the dataset for training and one-third of the dataset for testing the machine.

Henceforth, the research work will be shown in 'R' platform, which is an open source and used as an environment for the analysis of data [5].

2 Model Implementation, Prediction, and Optimization

2.1 Implementation of K-Nearest Neighbor Algorithm

K-nearest neighbor (KNN) algorithm is the simplest classification and regression predictive algorithm. KNN when used as classification algorithm classifies a new data point to a particular class. It is used as a regression algorithm, when a new data point gets labeled based on the average value of K-nearest neighbor. It is most widely used to solve the classification problems, is a robust and versatile classifier, and may act as a base classifier to many classification algorithms. *K* in the K-nearest neighbor algorithm signifies the number of data points to consider in a plane while classifying a new data point. The value of *K* may vary for various types of applications. However, it is recommended to initiate the process using the square root values of the total number of data points in the dataset. Thus, KNN finds and classifies two similar objects by measuring the distance between them and it uses the neighbor point's information to predict the target class.

KNN is a supervised learning algorithm used for machine learning based on prior examples. KNN is used for the classification of images, predicting weather, etc.

In *R*, KNN may be implemented using 'class' library. 'Caret' library is installed for data partitioning which is done using createDataPartition () function. Figure 1 shows results of data normalization after importing data into *R*. The attributes with varied ranges can be brought in the range of 0–1.

createDataPartition() function of caret library in *R* divides the dataset into train and test dataset. In this research paper, 1053 rows of the dataset are used for training data and 522 rows for the test.

Figure 2 shows the results of function knn() from the class library on the training and the test dataset with *K* value set to 40, i.e., nearly the square root of total records in the dataset. It shows the classification of test dataset to various labels.

Table 3 shows a matrix that illustrates predicted versus the actual values upon the application of KNN. For example, out of a total of twelve 'major earthquakes' in the

```
          Depth          Magnitude
 Min.   :  1.00   Min.   :2.10
 1st Qu.: 10.00   1st Qu.:3.50
 Median : 10.00   Median :4.50
 Mean   : 42.54   Mean   :4.74
 3rd Qu.: 35.00   3rd Qu.:6.00
 Max.   : 660.00   Max.   :8.20

Normalized Data:

          Depth          Magnitude
 Min.   :0.00000   Min.   :0.0000
 1st Qu.:0.01366   1st Qu.:0.2295
 Median :0.01366   Median :0.3934
 Mean   :0.06303   Mean   :0.4327
 3rd Qu.:0.05159   3rd Qu.:0.6393
 Max.   :1.00000   Max.   :1.0000
```

Fig. 1 Importing dataset into R and data normalization

```
     GREAT      LIGHT      MAJOR      MINOR   MODERATE
         0        116         10        180         78
NOT_SIGNIFICANT     STRONG
         5        133
```

Fig. 2 KNN algorithm in R and classification of test data based on training dataset

Table 3 Predicted versus actual values using KNN

	Actual values							
Predicted	Great	Light	Major	Minor	Moderate	Nonsignificant	Strong	Total
Great	0	0	0	0	0	0	0	0
Light	0	117	0	0	0	0	0	117
Major	1	0	11	0	0	0	0	12
Minor	0	0	0	173	0	6	0	179
Moderate	0	2	0	0	71	0	0	73
Nonsignificant	0	0	0	0	0	6	0	6
Strong	0	0	8	0	6	0	121	135

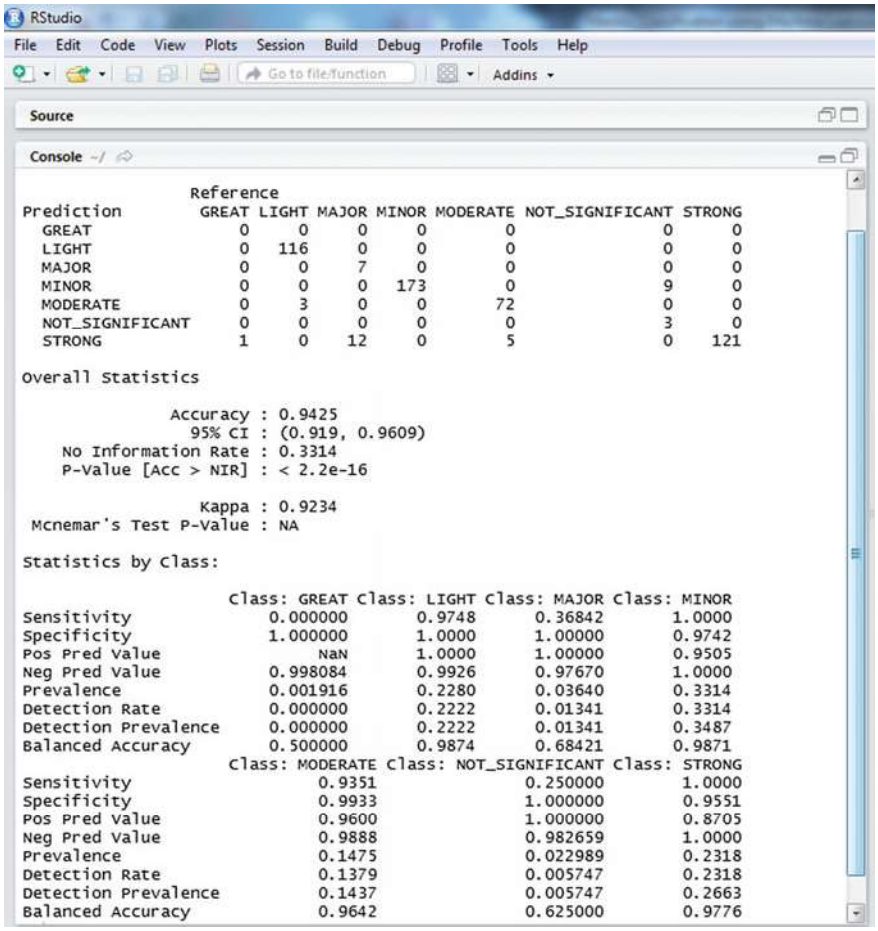
test dataset, eleven have been labeled as 'major' and one has been moved up in the category and labeled as a 'great' earthquake class label.

The diagonal values in Table 3 are the correctly classified data points, and others are incorrectly classified. Thus, the accuracy and the rate of misclassification are 94.25 and 5.74%, respectively. Therefore, the confusion matrix for knn represents the accuracy, specificity, and sensitivity for each class as shown in Fig. 3.

2.2 Implementation of Random Forest Algorithm in R

Random forest algorithm is based on the concept of 'cross-validation' which indicates how well model will work on an unknown dataset. Random forest model learns from predictors; therefore, it is necessary to find out the predictor variable(s) and target variable(s) in the dataset. Depending upon the partitions of the dataset, RF may work on threefold, fivefold, sevenfold, or tenfold cross-validation. In *R*, randomForest library provides support to work on the implementation of RF algorithm. After importing and partitioning the dataset, train () function was applied with the parameter methods = 'RF' and the results are shown in Table 4.

The accuracy and kappa matrices evaluate to nearly 100%. Therefore as shown in Table 4, almost all values are predicted correctly giving an accuracy of 100% and a misclassification of 0% on the given dataset. The predicted function applies the test data on the developed RF model. The confusion matrix for the above algorithm is as shown in Fig. 4.




```
                 Reference
Prediction        GREAT LIGHT MAJOR MINOR MODERATE NOT_SIGNIFICANT STRONG
  GREAT               0     0     0     0        0                0      0
  LIGHT               0   116     0     0        0                0      0
  MAJOR               0     0     7     0        0                0      0
  MINOR               0     0     0   173        0                9      0
  MODERATE            0     3     0     0       72                0      0
  NOT_SIGNIFICANT     0     0     0     0        0                3      0
  STRONG              1     0    12     0        5                0    121

Overall Statistics

               Accuracy : 0.9425
                 95% CI : (0.919, 0.9609)
    No Information Rate : 0.3314
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9234
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: GREAT Class: LIGHT Class: MAJOR Class: MINOR
Sensitivity              0.000000       0.9748      0.36842       1.0000
Specificity              1.000000       1.0000      1.00000       0.9742
Pos Pred Value                NaN       1.0000      1.00000       0.9505
Neg Pred Value           0.998084       0.9926      0.97670       1.0000
Prevalence               0.001916       0.2280      0.03640       0.3314
Detection Rate           0.000000       0.2222      0.01341       0.3314
Detection Prevalence     0.000000       0.2222      0.01341       0.3487
Balanced Accuracy        0.500000       0.9874      0.68421       0.9871
                     Class: MODERATE Class: NOT_SIGNIFICANT Class: STRONG
Sensitivity                   0.9351                0.250000        1.0000
Specificity                   0.9933                1.000000        0.9551
Pos Pred Value                0.9600                1.000000        0.8705
Neg Pred Value                0.9888                0.982659        1.0000
Prevalence                    0.1475                0.022989        0.2318
Detection Rate                0.1379                0.005747        0.2318
Detection Prevalence          0.1437                0.005747        0.2663
Balanced Accuracy             0.9642                0.625000        0.9776
```

Fig. 3 Confusion matrix for KNN algorithm on a given dataset

2.3 Implementation of Naïve Bayes Algorithm in R

Naïve Bayes is a probabilistic model based on the Bayes theorem which is mathematically stated as follows:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2}$$

In *R*, naiveBayes function is used to train the model and the results are shown in Fig. 5.

Table 5 shows the predicted versus the actual values using Naïve Bayes classification.

Table 4 Result of random forest in *R*

Random forest								
1053 samples								
2 predictors								
7 classes: 'GREAT,' 'LIGHT,' 'MAJOR,' 'MINOR,' 'MODERATE,'								
'NOT_SIGNIFICANT,' and 'STRONG'								
Preprocessing: centered (2), scaled (2) Resampling: Cross-validated (fivefold) Summary of sample sizes: 840, 842, 845, 841, 844 Resampling results across tuning parameters:								
mtry accuracy kappa								
1 0.9943216 0.9925588								
2 1.0000000 1.0000000								
Accuracy was used to select the optimal model using the largest value. The final value used for the model was mtry = 2								
Predicted →	Great	Light	Major	Minor	Moderate	Nonsignificant	Strong	Total
Great	1	0	0	0	0	0	0	1
Light	0	119	0	0	0	0	0	119
Major	0	0	19	0	0	0	0	19
Minor	0	0	0	173	0	0	0	173
Moderate	0	0	0	0	77	0	0	77
Nonsignificant	0	0	0	0	0	12	0	12
Strong	0	0	0	0	0	0	121	121

Therefore, the accuracy of the algorithm is 97.89% with 2.10% of misclassification error. The confusion matrix for the Naïve Bayes algorithm is as shown in Fig. 6.

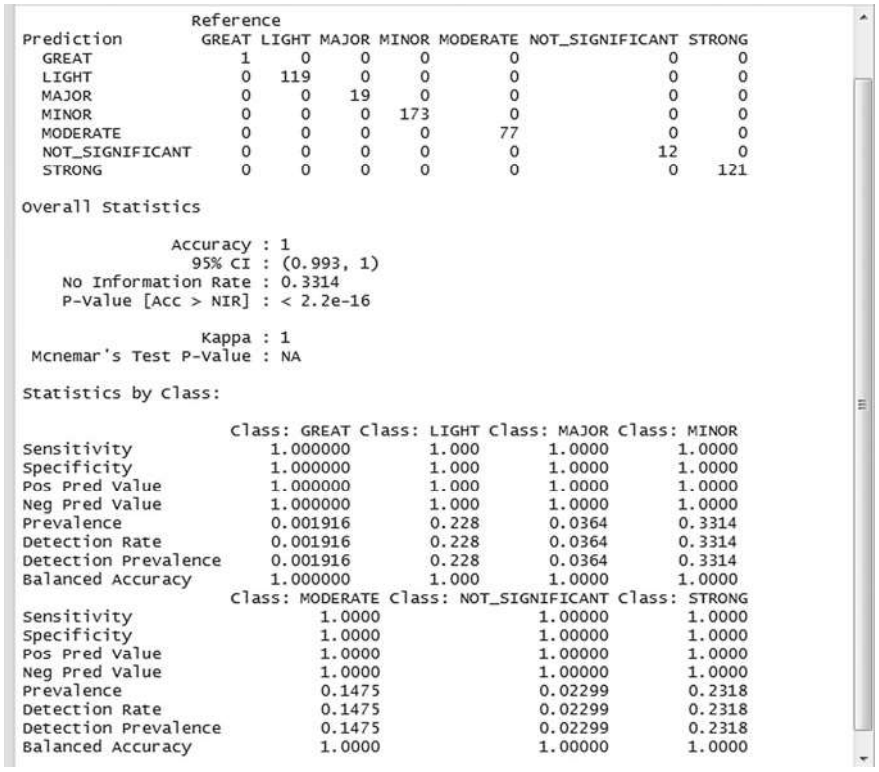
2.4 Implementation of Support Vector Machine Algorithm in R

It is a classification algorithm used to find extremes in the same dataset as above. Support vectors are the points close to opposite class. SVM function was applied on the dataset as shown, and the results are shown in Fig. 7.

Figure 8 shows the plot of the support vectors after the model is applied on the dataset.

Table 6 applies the model on the test dataset and shows predicted versus the actual values.

Therefore, the accuracy of the algorithm is 95.74% with 4.25% of misclassification error. The confusion matrix for the SVM algorithm is as shown in Fig. 9.



```
                Reference
Prediction       GREAT LIGHT MAJOR MINOR MODERATE NOT_SIGNIFICANT STRONG
  GREAT              1     0     0     0        0                0      0
  LIGHT              0   119     0     0        0                0      0
  MAJOR              0     0    19     0        0                0      0
  MINOR              0     0     0   173        0                0      0
  MODERATE           0     0     0     0       77                0      0
  NOT_SIGNIFICANT    0     0     0     0        0               12      0
  STRONG             0     0     0     0        0                0    121

Overall Statistics

               Accuracy : 1
                 95% CI : (0.993, 1)
    No Information Rate : 0.3314
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: GREAT Class: LIGHT Class: MAJOR Class: MINOR
Sensitivity              1.000000        1.000       1.0000       1.0000
Specificity              1.000000        1.000       1.0000       1.0000
Pos Pred Value           1.000000        1.000       1.0000       1.0000
Neg Pred Value           1.000000        1.000       1.0000       1.0000
Prevalence               0.001916        0.228       0.0364       0.3314
Detection Rate           0.001916        0.228       0.0364       0.3314
Detection Prevalence     0.001916        0.228       0.0364       0.3314
Balanced Accuracy        1.000000        1.000       1.0000       1.0000
                     Class: MODERATE Class: NOT_SIGNIFICANT Class: STRONG
Sensitivity                1.0000                1.00000        1.0000
Specificity                1.0000                1.00000        1.0000
Pos Pred Value             1.0000                1.00000        1.0000
Neg Pred Value             1.0000                1.00000        1.0000
Prevalence                 0.1475                0.02299        0.2318
Detection Rate             0.1475                0.02299        0.2318
Detection Prevalence       0.1475                0.02299        0.2318
Balanced Accuracy          1.0000                1.00000        1.0000
```

Fig. 4 Confusion matrix for RF algorithm on a given dataset

In addition to linear classification, support vector machine (SVM) with efficiency performs a nonlinear classification using different types of kernel functions, implicitly by mapping their inputs into high-dimensional feature areas [11]. The algorithm was applied on the dataset and using the following four kernel functions, and the result obtained is shown in Table 7.

The comparison between the various algorithms used on the dataset is as shown in Table 8.

3 Conclusion and Future Work

Classification is the process of assigning a class label to each data item in the dataset. This research paper successfully classified each data value by using various machine learning algorithms, namely K-nearest neighbor, support vector machine, Naïve Bayes, and random forest. All the said algorithms are applied on the same dataset primarily using the two main attributes of the earthquake dataset, namely

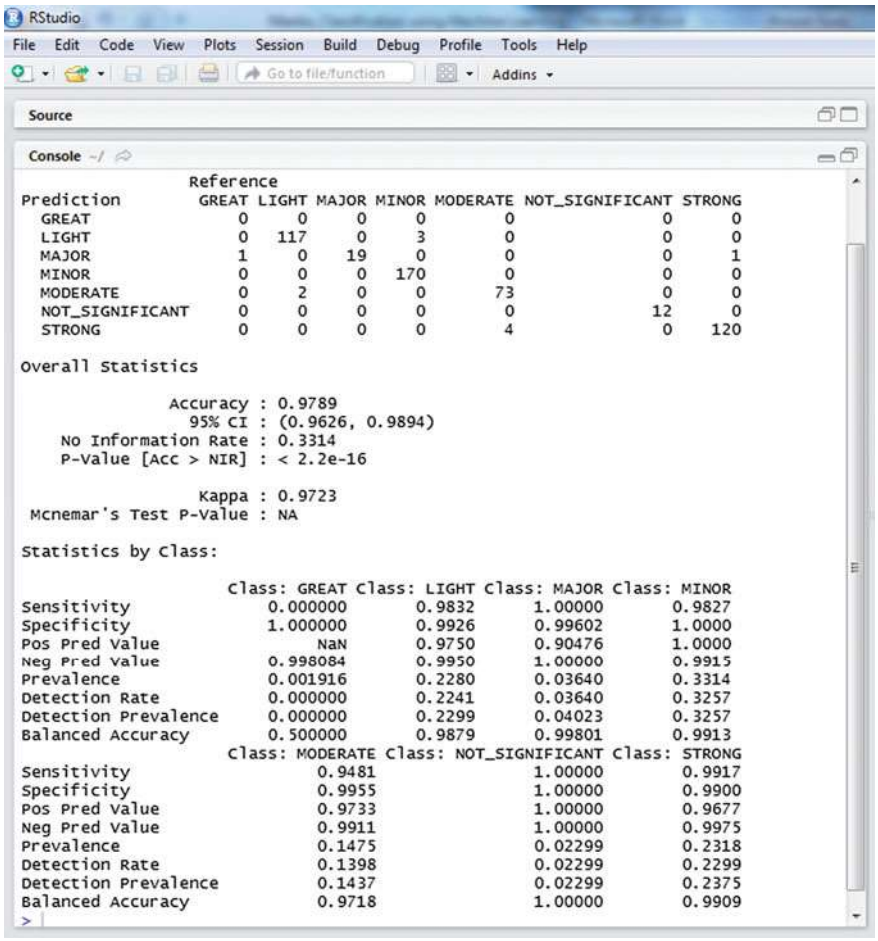
```
Naive Bayes Classifier for Discrete Predictors

          GREAT           LIGHT           MAJOR           MINOR        MODERATE
    0.002849003     0.226970560     0.037037037     0.329534663     0.148148148
NOT_SIGNIFICANT          STRONG
    0.024691358     0.230769231

Conditional probabilities:
                 Depth
train_data$Output      [,1]       [,2]
  GREAT           405.33333 342.790509
  LIGHT            26.66946  32.626154
  MAJOR            99.02564 169.710045
  MINOR            19.54467  21.578333
  MODERATE         49.14744  58.912379
  NOT_SIGNIFICANT   9.50000   7.273239
  STRONG           82.20576 145.028719

                 Magnitude
train_data$Output     [,1]      [,2]
  GREAT           8.100000 0.1000000
  LIGHT           4.423013 0.2947968
  MAJOR           7.271795 0.2937496
  MINOR           3.416427 0.2609253
  MODERATE        5.277564 0.2443091
  NOT_SIGNIFICANT 2.715385 0.2274947
  STRONG          6.373251 0.2691902
```

Fig. 5 Naive Bayes in *R*

Table 5 Predicted versus actual values using NB algorithm

	Actual values							
Predicted →	Great	Light	Major	Minor	Moderate	Nonsignificant	Strong	Total
Great	0	0	0	0	0	0	0	0
Light	0	117	0	3	0	0	0	120
Major	1	0	19	0	0	0	1	21
Minor	0	0	0	170	0	0	0	170
Moderate	0	2	0	0	73	0	0	75
Nonsignificant	0	0	0	0	0	12	0	12
Strong	0	0	0	0	4	0	120	124

the depth and the magnitude. The algorithms were implemented on the dataset, and their accuracy is compared. It is observed that random forest algorithm has been the most accurate algorithm among others. Support vector machine with radial and linear kernel method has also shown the great accuracy and low misclassification error percentage. However, in the future, many other classification algorithms may be applied and the results can be compared. Further, more attributes of the dataset may be included in the study in order to add more accuracy during the process of




```
                Reference
Prediction       GREAT LIGHT MAJOR MINOR MODERATE NOT_SIGNIFICANT STRONG
  GREAT              0     0     0     0        0               0      0
  LIGHT              0   117     0     3        0               0      0
  MAJOR              1     0    19     0        0               0      1
  MINOR              0     0     0   170        0               0      0
  MODERATE           0     2     0     0       73               0      0
  NOT_SIGNIFICANT    0     0     0     0        0              12      0
  STRONG             0     0     0     0        4               0    120

Overall Statistics

               Accuracy : 0.9789
                 95% CI : (0.9626, 0.9894)
    No Information Rate : 0.3314
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9723
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: GREAT Class: LIGHT Class: MAJOR Class: MINOR
Sensitivity              0.000000       0.9832      1.00000       0.9827
Specificity              1.000000       0.9926      0.99602       1.0000
Pos Pred Value                NaN       0.9750      0.90476       1.0000
Neg Pred Value           0.998084       0.9950      1.00000       0.9915
Prevalence               0.001916       0.2280      0.03640       0.3314
Detection Rate           0.000000       0.2241      0.03640       0.3257
Detection Prevalence     0.000000       0.2299      0.04023       0.3257
Balanced Accuracy        0.500000       0.9879      0.99801       0.9913
                     Class: MODERATE Class: NOT_SIGNIFICANT Class: STRONG
Sensitivity                   0.9481                1.00000        0.9917
Specificity                   0.9955                1.00000        0.9900
Pos Pred Value                0.9733                1.00000        0.9677
Neg Pred Value                0.9911                1.00000        0.9975
Prevalence                    0.1475                0.02299        0.2318
Detection Rate                0.1398                0.02299        0.2299
Detection Prevalence          0.1437                0.02299        0.2375
Balanced Accuracy             0.9718                1.00000        0.9909
>
```

Fig. 6 Confusion matrix for NB algorithm

machine learning. Also, the dataset used may be expanded to include large volume of records and the study may then be done based on real-time big data. Thus, a system is recommended where the moment of an earthquake event, the data is added in the dataset and is automatically assigned a class label based on the classification algorithm applied.

```
Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.5
Number of Support Vectors:  511
( 82  41  110  4  138  102  34 )
Number of Classes:  7
Levels:
 GREAT LIGHT MAJOR MINOR MODERATE NOT_SIGNIFICANT
STRONG
```

Fig. 7 SVM function on the dataset

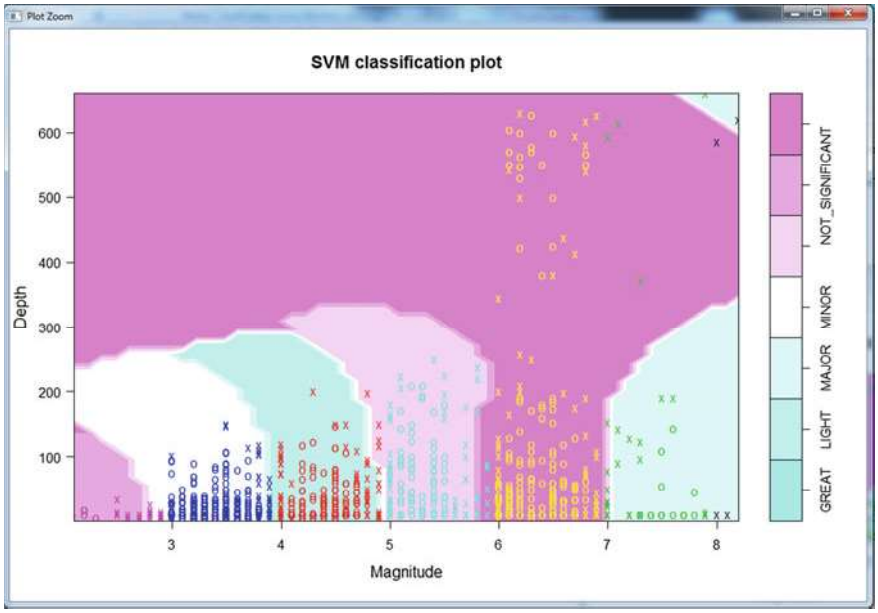


Fig. 8 Plot of support vectors

Table 6 Predicted versus actual values using SVM

	Actual values						
Predicted →	Great	Light	Major	Minor	Moderate	Nonsignificant	Strong
Great	0	0	0	0	0	0	0
Light	0	350	0	0	0	0	0
Major	3	0	35	0	0	0	0
Minor	0	0	0	520	0	22	0
Moderate	0	2	0	0	223	0	0
Nonsignificant	0	0	0	0	0	16	0
Strong	1	0	23	0	10	0	364

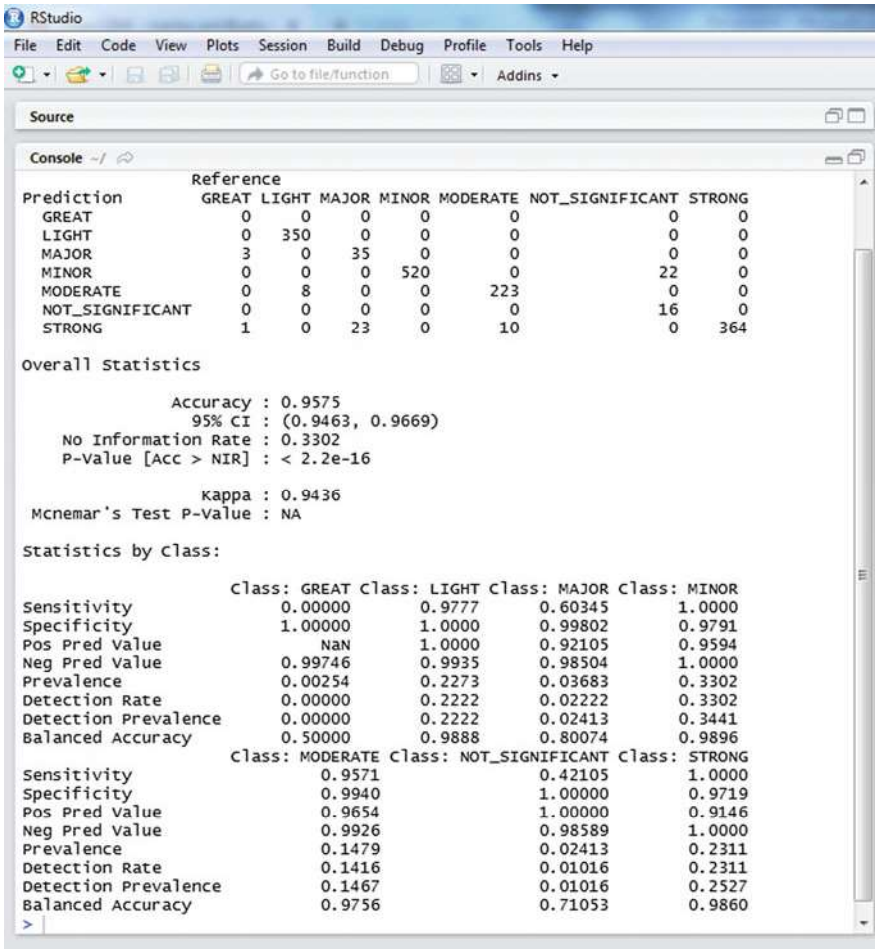


Fig. 9 Confusion matrix for SVM algorithm

Table 7 Comparison of various kernel functions in SVM

S. no.	Kernel function	No. of support vectors	Accuracy (in %)	Misclassification error (in %)
1	Radial	511 (82, 41,110,4,138, 102, 34)	95.74	4.25
2	Linear	463 (66, 36, 107, 4, 128, 90, 32)	95.74	4.25
3	Polynomial	775 (62,2,137,3,315,215,22)	85.77	14.22
4	Sigmoid	657 (91, 58, 136, 4, 162,168,38)	82.03	17.96

Table 8 Result comparison of different algorithms

	K-nearest neighbor	Random forest	Naïve Bayes	Support vector machine
Accuracy (%)	94.25	100	97.89	95.74
Misclassification error (%)	5.74	0	2.10	4.25

References

1. Murty CVR (2005) Earthquake tip 3, learning earthquake design and construction. National Information Center of Earthquake Engineering Indian Institute of Technology, Kanpur
2. Classification of Earthquakes. https://civildigital.com/classification-earthquakes/. Last accessed 14 June 2018
3. Lahr JC, Chouet BA, Stephens CD, Power JA, Page RA (1994) Earthquake classification, location, and error analysis in a volcanic environment: implications for the magmatic system of the 1989–1990 Eruptions at Redoubt Volcano, Alaska. J Volcanol Geotherm Res 64(1–4)
4. Preliminary locations of Earthquakes. http://www.imd.gov.in/pages/earthquake_prelim.php. Last accessed 22 Aug 2018
5. Jović A, Brkić K, Bogunović N (2014) An overview of free software tools for general data mining. MIPRO 2014, Opatija, Croatia
6. Ertöz L, Steinbach M, Kumar V (2001) Finding topics in collections of document—a shared nearest neighbor approach. In: Proceedings of text mine 01, first SIAM international conference on data mining, Chicago
7. Noi PT, Kappas M (2018) Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using Sentinel-2 imagery sensors 2018, 18, 18; https://doi.org/10.3390/s18010018
8. K-Nearest neighbor. https://i.ytimg.com/vi/Jtmh21_FAoU/maxresdefault.jpg. Last accessed 01 Aug 2018
9. Information about different centres at Indian Meteorological Department (IMD), Delhi, http://www.imd.gov.in/pages/services_seismo.php. Last accessed 20 Feb 2018
10. Szeliga W, Hough S, Martin S, Bilham R (2010) Intensity, magnitude, location, and attenuation in india for felt earthquakes since 1762. Bull Seismol Soc Am 100(2):570–584. https://doi.org/10.1785/0120080329
11. Kaur R, Bansal M (2016) Multidimensional attacks classification based on genetic algorithm and SVM. In: 2nd International conference on next generation computing technologies (NGCT-2016), Dehradun. 978-1-5090-3257-0/16/2016, IEEE