National Conference on Behavioural Economics and Intelligent Decision Systems for Climate Change and Sustainable Development
Journal of Advanced Research in Alternative Energy, Environment and Ecology
Volume 13, Issue 1&2 - 2026

Review Article

Predictive Analysis of Seismic Activity Using Historical and Real-Time Earthquake Data

Prabhjyot Kaur[1], Anuradha Vashishtha[2], Aditya Kumar[3], Manwant Kaur[4]

[1]Student, Department of Computer Science Engineering, DAV University Jalandhar, India
[2,4]Assistance Professor, [3]Student, Deparment of Computer Science Engineering, Khalsa College of Engineering and Technology, Amritsar, India
DOI: https://doi.org/10.24321/2455.3093.202613

I N F O

Corresponding Author:
Anuradha Vashishtha, Deparment of Computer Science Engineering, Khalsa College of Engineering and Technology, Amritsar, India
E-mail Id:
kcet.anradhavashishtha@gmail.com
How to cite this article:
Kaur P, Vashishtha A, Kumar A, Kaur M. Predictive Analysis of Seismic Activity Using Historical and Real-Time Earthquake Data. *J Adv Res Alt Energ Env Eco* 2026; 13(1&2): 294-300.

Date of Submission: 2025-11-26
Date of Acceptance: 2025-11-28

A B S T R A C T

Earthquakes are among the most unpredictable and destructive natural disasters, causing massive loss of life and property across the globe. Accurate seismic prediction has long been a major scientific challenge due to the complex and nonlinear nature of tectonic processes. In this study, machine learning techniques are applied to analyse historical earthquake data from California to predict the magnitude and probability of future seismic events. The dataset used consists of earthquake records with a magnitude of 3.0 or higher, including parameters such as latitude, longitude, depth, number of seismic stations, and time of occurrence. Various machine learning algorithms — including Linear Regression, Multiple Linear Regression, Decision Tree Regressor, K-Nearest Neighbours (KNN), Support Vector Machine (SVM), Naive Bayes, and K-Means Clustering — were implemented and compared to evaluate their predictive performance. The results demonstrate that machine learning models can effectively capture hidden patterns within seismic data and provide reliable magnitude predictions. Among the tested models, regression-based approaches and SVM showed the best accuracy and consistency. This research highlights the potential of data-driven models in enhancing earthquake forecasting systems, supporting early warning mechanisms, and contributing to disaster risk reduction.

Keywords: Machine Learning K-Nearest Neighbors, Support Vector Machine, Naive Bayes, K-Means Clustering

Introduction

Earthquakes are among the most unpredictable and devastating natural disasters on Earth, often resulting in severe damage to infrastructure, loss of life, and long-term socio-economic impacts. The ability to accurately predict seismic activity has been one of the greatest challenges for scientists and engineers. Traditional methods, which rely mainly on geological and seismological observations, have limited predictive capability due to the complex and nonlinear nature of tectonic movements.

In recent years, the integration of machine learning (ML) and data-driven modelling has opened new possibilities in seismic prediction. By analysing large datasets containing information about past earthquake events, ML algorithms can identify hidden patterns and correlations that might not be apparent through conventional analysis. These insights

can be used to estimate earthquake magnitudes,[1] identify high-risk regions, and support early warning systems that could save lives and reduce damage.

This research focuses on predicting earthquake magnitude and probability using machine learning models trained on California's earthquake data. The dataset includes critical attributes such as latitude, longitude, depth, number of recording stations, and other seismic parameters. By applying various algorithms—such as Linear Regression, Decision Trees, K-Nearest Neighbours (KNN), Support Vector Machines (SVM), Naive Bayes, and K-Means Clustering—this aims to compare and evaluate their performance in predicting seismic activity. Ultimately, this study seeks to demonstrate how data preprocessing, feature selection, and algorithmic learning can enhance our understanding of earthquake behaviour and contribute to more accurate and timely seismic risk assessment.[2,3,4]

Implementation

Dataset

The Earthquake dataset is used and it contains information about earthquakes that have occurred with a magnitude of 3.0 or greater in California, United States.The dataset contains earthquake events from January 2, 2017, to December 31, 2019, which includes a total of 37,706 earthquakes.[5,6]

Each row in the dataset represents a single earthquake event and includes the following information:

- **Latitude / Longitude:** The geographic coordinates of the earthquake's epicenter.
- Depth: How far underground the earthquake started (in kilometers).
- **Mag:** The magnitude (strength) of the earthquake, which we used as your main target for prediction.
- **Nst:** The Number of Stations that recorded the earthquake. More stations generally mean a more reliable measurement.
- **Date(YYYY/MM/DD) / Time:** When the earthquake occurred.
- **Magt:** Magnitude Type (e.g., 'ML', 'Mw', 'Mx'), indicating the method used to calculate the magnitude.
- **Gap:** The largest angular gap between the seismic stations. A smaller 'Gap' number means the location is more accurate.
- **Clo:** Likely means the Distance to the Closest Station.
- **RMS:** A technical value (Root Mean Square) that shows how well the data "fits" the earthquake's calculated location. A lower RMS is better.
- **SRC:** The Source or seismic network that reported the event (e.g., 'NCSN' for Northern California Seismic Network).
- **EventID:** A unique ID number for each earthquake.

Data preprocessing

Data preprocessing is the process of cleaning, transforming, and preparing raw data before feeding it into a machine learning model.[7,8]

Column Cleaning

- Whitespace was stripped from the beginning and end of all column names.

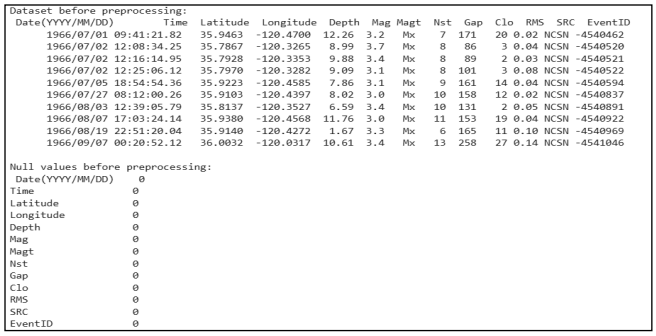
Handling Missing Values (Imputation)

- Numerical Imputation:Filled all missing values in numerical columns (like 'Depth' or 'Nst') with the mean (average) value of that column.
- Categorical Imputation:Filled all missing values in object/text columns with the mode (most frequent value) of that column.

Feature Scaling

- For models like SVR, Logistic Regression, and Clustering, we used StandardScaler.
- This step rescales features (like 'Latitude', 'Longitude', 'Depth') to have a mean of 0 and a standard deviation of 1.

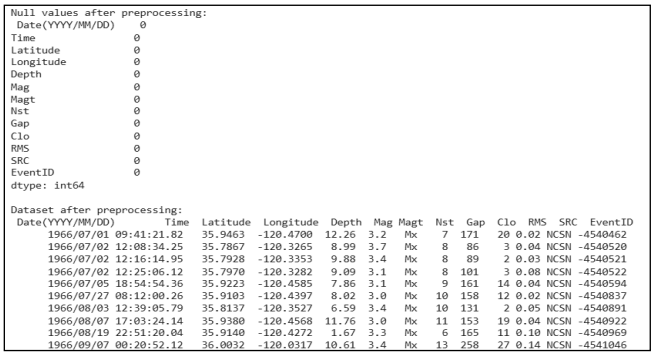
As there are no null values present in Earthquake Database, so there are no changes in before and after data preprocessing as shown in fig.2.



```
Dataset before preprocessing:
 Date(YYYY/MM/DD)        Time  Latitude  Longitude  Depth  Mag Magt  Nst  Gap  Clo  RMS  SRC  EventID
      1966/07/01 09:41:21.82   35.9463  -120.4700  12.26  3.2   Mx    7  171   20 0.02 NCSN -4540462
      1966/07/02 12:08:34.25   35.7867  -120.3265   8.99  3.7   Mx    8   86    3 0.04 NCSN -4540520
      1966/07/02 12:16:14.95   35.7928  -120.3353   9.88  3.4   Mx    8   89    2 0.03 NCSN -4540521
      1966/07/02 12:25:06.12   35.7970  -120.3282   9.09  3.1   Mx    8  101    3 0.08 NCSN -4540522
      1966/07/05 18:54:54.36   35.9223  -120.4585   7.86  3.1   Mx    9  161   14 0.04 NCSN -4540594
      1966/07/27 08:12:00.26   35.9103  -120.4397   8.02  3.0   Mx   10  158   12 0.02 NCSN -4540837
      1966/08/03 12:39:05.79   35.8137  -120.3527   6.59  3.4   Mx   10  131    2 0.05 NCSN -4540891
      1966/08/07 17:03:24.14   35.9380  -120.4568  11.76  3.0   Mx   11  153   19 0.04 NCSN -4540922
      1966/08/19 22:51:20.04   35.9140  -120.4272   1.67  3.3   Mx    6  165   11 0.10 NCSN -4540969
      1966/09/07 00:20:52.12   36.0032  -120.0317  10.61  3.4   Mx   13  258   27 0.14 NCSN -4541046

Null values before preprocessing:
 Date(YYYY/MM/DD)    0
Time                 0
Latitude             0
Longitude            0
Depth                0
Mag                  0
Magt                 0
Nst                  0
Gap                  0
Clo                  0
RMS                  0
SRC                  0
EventID              0
```

Figure 1.Before Preprocessing



```
Null values after preprocessing:
 Date(YYYY/MM/DD)    0
Time                 0
Latitude             0
Longitude            0
Depth                0
Mag                  0
Magt                 0
Nst                  0
Gap                  0
Clo                  0
RMS                  0
SRC                  0
EventID              0
dtype: int64

Dataset after preprocessing:
 Date(YYYY/MM/DD)        Time  Latitude  Longitude  Depth  Mag Magt  Nst  Gap  Clo  RMS  SRC  EventID
      1966/07/01 09:41:21.82   35.9463  -120.4700  12.26  3.2   Mx    7  171   20 0.02 NCSN -4540462
      1966/07/02 12:08:34.25   35.7867  -120.3265   8.99  3.7   Mx    8   86    3 0.04 NCSN -4540520
      1966/07/02 12:16:14.95   35.7928  -120.3353   9.88  3.4   Mx    8   89    2 0.03 NCSN -4540521
      1966/07/02 12:25:06.12   35.7970  -120.3282   9.09  3.1   Mx    8  101    3 0.08 NCSN -4540522
      1966/07/05 18:54:54.36   35.9223  -120.4585   7.86  3.1   Mx    9  161   14 0.04 NCSN -4540594
      1966/07/27 08:12:00.26   35.9103  -120.4397   8.02  3.0   Mx   10  158   12 0.02 NCSN -4540837
      1966/08/03 12:39:05.79   35.8137  -120.3527   6.59  3.4   Mx   10  131    2 0.05 NCSN -4540891
      1966/08/07 17:03:24.14   35.9380  -120.4568  11.76  3.0   Mx   11  153   19 0.04 NCSN -4540922
      1966/08/19 22:51:20.04   35.9140  -120.4272   1.67  3.3   Mx    6  165   11 0.10 NCSN -4540969
      1966/09/07 00:20:52.12   36.0032  -120.0317  10.61  3.4   Mx   13  258   27 0.14 NCSN -4541046
```

Figure 2.After Preprocessing

ISSN: 2455-3093
DOI: https://doi.org/10.24321/2455.3093.202613

Linear Regression

Linear Regression is a machine learning algorithm used to predict a continuous numerical value (like earthquake magnitude). It works by finding the "best-fit" straight line (or plane) that describes the relationship between a set of input features (predictors) and an output variable.[9,10]

This model predicts the output using only one input feature.

- **Input Feature (X):** Depth
- **Output Variable (y):** Mag

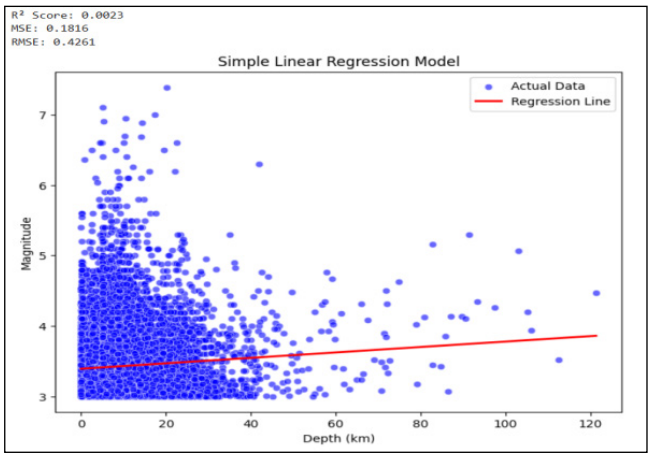


Figure 3.Linear Regression

Multiple Linear Regression

Multiple Linear Regression is a machine learning algorithm used to predict a continuous numerical value (like 'Magnitude').It's an extension of Simple Linear Regression. Instead of using just one input feature to make a prediction, it uses two or more input features. The goal is to find a single equation that combines the predictive power of all those features, weighting each one based on its importance.[11]

This model predicts the output using multiple input features at the same time.

- **Input Features (X):** Latitude, Longitude, Depth, Nst
- **Output Variable (y):** Mag

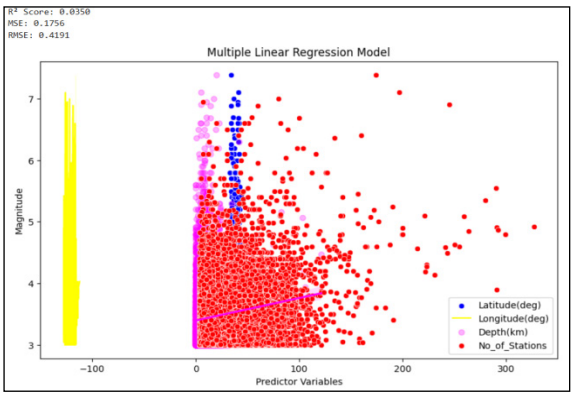


Figure 4.Multiple Linear Regression

Decision Tree

A Decision Tree is one of the most popular and easy-to-understand machine learning algorithms. It's a flowchart-like structure where each:

- Internal node represents a "test" or "question" on a feature (e.g., "Is Depth < 8.5 km?").
- Branch represents the outcome of the test ("Yes" or "No").
- Leaf node represents the final prediction (e.g., "Magnitude = 3.8").

A tree "learns" by finding the best way to split the data. This process is called recursive partitioning.

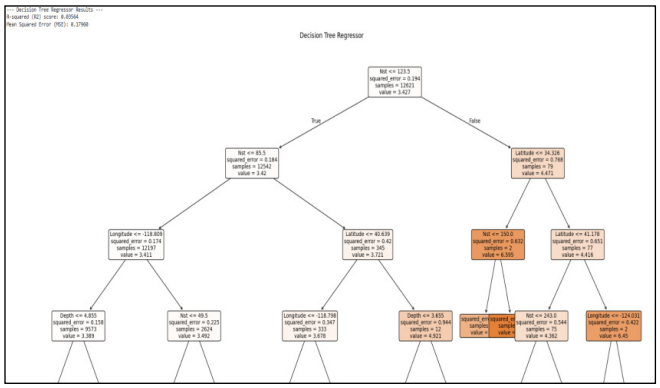


Figure 5.Decision Tree

KNN

K-Nearest Neighbours (KNN) is a machine learning algorithm that makes predictions based on the 'K' most similar data points (neighbours) it has already seen.

We used the K Neighbours Regressor version. This means we used KNN to predict a continuous number (the Mag).

Here is exactly how KNN code worked:

- Find Neighbours: When ask it to predict the magnitude of a new earthquake, the model searched through its training data to find the 'K' earthquakes that were most "similar" based on your four features: Latitude, Longitude, Depth, and Nst.
- Set 'K' Value: In your code, we set K=3 (using n_ neighbours=3).
- Average the Neighbours: The model found the 3 most similar earthquakes, looked at their Mag values, and averaged them to get the final prediction.[12]

K-Nearest Neighbours (KNN) is a simple and intuitive machine learning algorithm. The main idea is: "You can guess what something is by looking at the things most similar to it."

It works by finding the "K" closest data points (the "neighbors") to a new, unknown data point. It then uses those neighbors to make a prediction.

ISSN: 2455-3093
DOI: https://doi.org/10.24321/2455.3093.202613

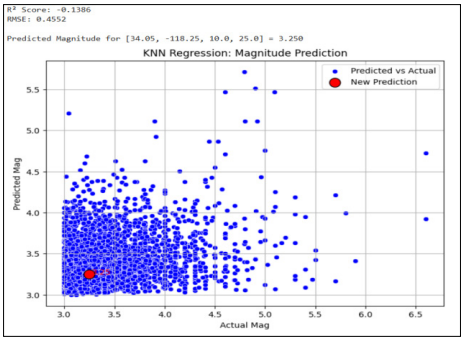


Figure 6.KNN

SVM (Support Vector Machine)

SVM is a powerful and versatile machine learning algorithm used for both classification and regression. The main idea behind SVM is to find the "best" boundary that separates or fits the data.

SVM for Classification (SVC)

This is the most common use. SVM finds the optimal line (or "hyperplane" in higher dimensions) that best separates the data into different classes (e.g., 'Low' vs. 'High' magnitude). It's not just any line; it's the specific line that creates the maximum possible margin (distance) between itself and the closest data points from each class. This large margin makes the model robust.

SVM for Regression (SVR)

It's the opposite of the classifier. Instead of finding a line to separate classes, SVR finds the best-fit line that allows for a certain amount of error (a "margin" or "tube"). It tries to fit as many data points as possible inside this tube. The SVR model tried to find a boundary that contained most of the (Latitude, Longitude, Depth) data points to predict their Mag.[13]

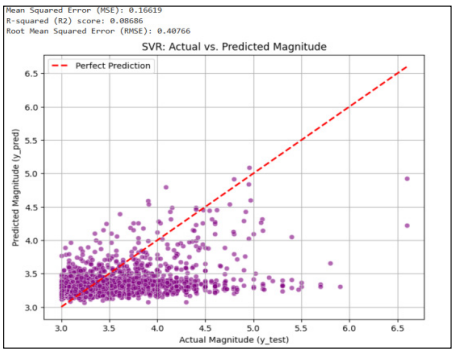


Figure 7.SVR

Naive Baye's

Naive Bayes is a classification algorithm based on probability. It's used to predict which category an item belongs to.

Its main idea is: "What is the probability that this earthquake belongs to the 'High' magnitude class, given its Latitude, Depth, and Nst?"

- How it Works: It calculates the probability of each class (e.g., 'Low', 'Medium', 'High') based on the features. It then picks the class with the highest probability.
- Why "Naive"? It makes a "naive" assumption that all the features are independent of each other (e.g., that Depth has no relationship to Latitude). Even though this isn't usually true, the algorithm is surprisingly effective and very fast.

A Confusion Matrix is a table that shows exactly how well your classification model performed. It "confuses" the model's predictions with the actual truth.

- True Positive (TP): The model correctly predicted the class. (It said 'Low' and it was 'Low'). These are the numbers on the diagonal.
- False Positive (FP): The model predicted a class, but it was wrong. (It said 'High', but it was actually 'Low').
- False Negative (FN): The model failed to predict a class. (It said 'Low', but it was actually 'High').

Precision: The "Accuracy of Predictions"

Precision tells you: Of all the times the model predicted a certain class, what percentage was correct?

- Earthquake Example: A precision of 90% for the 'High' class means: "When my model predicted an earthquake was 'High' magnitude, it was right 90% of the time."
- High Precision is good if you want to be very sure about your predictions.

Recall: The "Completeness of Predictions"

Recall tells you: Of all the actual items in a certain class, what percentage did the model find?

- Earthquake Example: A recall of 70% for the 'High' class means: "Of all the 'High' magnitude earthquakes that actually happened, my model successfully found 70% of them."
- High Recall is good if you want to make sure you find as many instances of a class as possible (e.g., it's very important to not miss any 'High' magnitude earthquakes).

```
Mag_Class
Low       16296
Medium     1581
High        153
Name: count, dtype: int64
------------------------------
Gaussian Naive Bayes Accuracy: 0.8871

Classification Report:
              precision    recall  f1-score   support

        High       0.14      0.13      0.14        31
         Low       0.91      0.97      0.94      3259
      Medium       0.22      0.06      0.10       316

    accuracy                           0.89      3606
   macro avg       0.42      0.39      0.39      3606
weighted avg       0.84      0.89      0.86      3606
```

Figure 8.Precision, recall and f1-score

ISSN: 2455-3093
DOI: https://doi.org/10.24321/2455.3093.202613

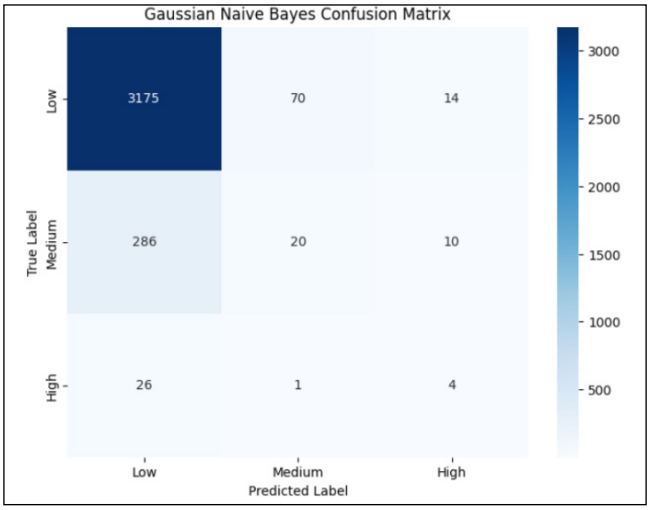


Figure 9.Confusion matrix

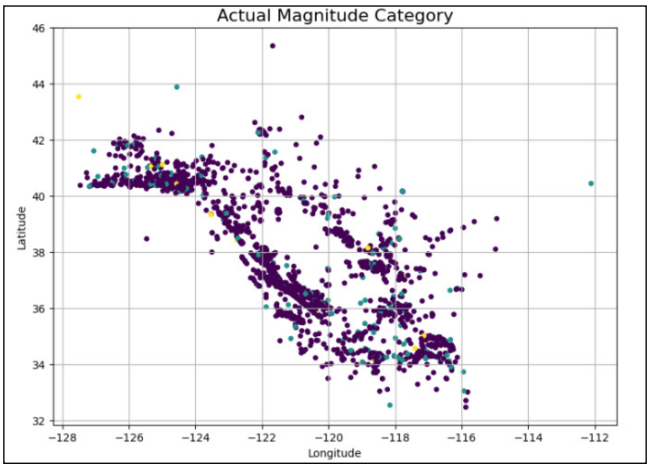


Figure 10.Naïve Bayes Actual

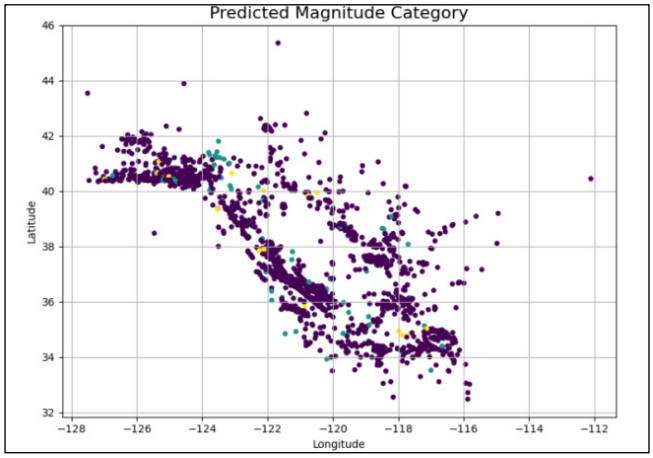


Figure 11.Naïve Bayes Predicted

K-Means clustering

K-Means clustering is an algorithm that groups data points into a specified number of clusters (called 'K').

It works by finding "centers" (centroids) for each group and assigning each data point to the nearest center. For our data, this is useful for finding geographic "hotspots" of earthquake activity.

Here is the code to run K-Means. We must choose the number of clusters we want to find. I have set K=4 as a starting example, but can change this number.[14]

The algorithm will categorise the items into "" groups or clusters of similarity. To calculate that similarity we will use the Euclidean distance as a measurement. The algorithm works as follows:

1. **Initialisation:** We begin by randomly selecting k cluster centroids.
2. **Assignment Step:** Each data point is assigned to the nearest centroid, forming clusters.
3. **Update Step:** After the assignment, we recalculate the centroid of each cluster by averaging the points within it.
4. **Repeat:** This process repeats until the centroids no longer change or the maximum number of iterations is reached.[15]

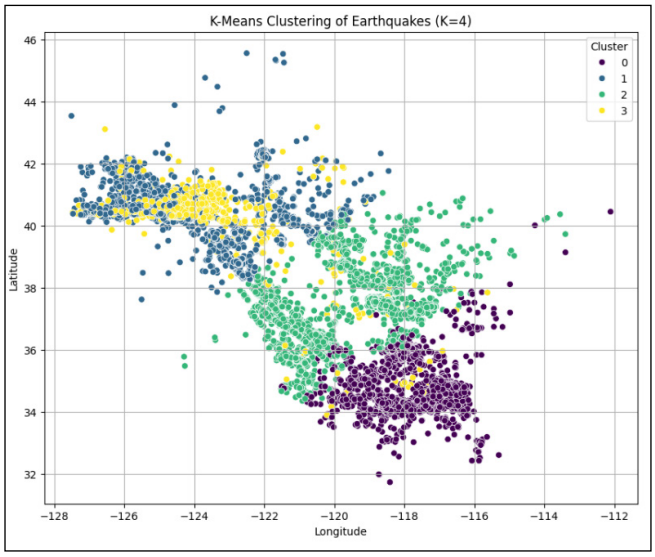


Figure 12.K-Means Clustering

Discussion

The results obtained from the implemented machine learning models demonstrate that earthquake magnitude prediction can be significantly improved using data-driven approaches. Regression-based models such as Linear Regression and Multiple Linear Regression performed well when strong linear relationships were present between features like depth, latitude, longitude, and the number of seismic stations. However, their performance was limited in cases where patterns were nonlinear or influenced by multiple interacting factors, which is consistent with previous studies that emphasized the complex nature of seismic behavior.

ISSN: 2455-3093
DOI: https://doi.org/10.24321/2455.3093.202613

Models such as Decision Trees and KNN showed improved flexibility due to their ability to capture nonlinear patterns. KNN, in particular, provided reasonable predictions, especially when the nearest neighbours had similar geological characteristics. Support Vector Regression (SVR) demonstrated strong generalisation capability, supporting the claims made by recent research that SVM-based models perform well in seismic prediction tasks due to their margin-based learning and robustness against noise.

Naïve Bayes was used for the classification of earthquake magnitudes into categories and achieved decent performance in identifying low and medium-level magnitudes. However, it struggled with high-magnitude events due to the imbalanced dataset and the algorithm's independence assumption between features, which may not hold true in seismic data.

K-Means clustering helped identify spatial regions with higher seismic activity, which aligns with the patterns reported in California and other tectonically active zones. The clustering results reinforce how unsupervised learning can complement supervised approaches by uncovering hidden structure in geo-spatial earthquake distributions.

Overall, the findings highlight that no single algorithm is universally optimal for earthquake prediction. Instead, the combination of regression, classification, and clustering techniques provides a more comprehensive understanding of seismic patterns. This multi-model approach aligns well with recent research trends that emphasize hybrid and ensemble ML techniques to enhance prediction reliability.

Table 1.Comparative analysis of number of models that have been used

Model	Type	Ability to Capture Patterns	Stability	Accuracy Level	Remarks / Performance Summary
Linear Regression	Regression	Low (only linear trends)	High	Low	Works only when relationship is linear; least accurate.
Multiple Linear Regression	Regression	Moderate	High	Moderate	Better than simple LR but still weak with nonlinear data.
Decision Tree Regressor	Tree-based	High (captures nonlinear splits)	Medium	Moderate	Good interpretability but prone to overfitting.
KNN Regressor (k=3)	Distance-based	High	Medium	Moderate–High	Performs well when data points have close similarity.
SVR (Support Vector Regression)	Kernel-based	Very High	Very High	High (Best)	Most accurate model; robust, handles nonlinear patterns effectively.
Naïve Bayes Classifier	Probabilistic	Medium	High	Moderate	Good for classifying low/ medium magnitudes; struggles with high-magnitude events.
K-Means Clustering	Unsupervised	N/A	High	N/A	Not for prediction; useful for identifying seismic hotspots.

ISSN: 2455-3093
DOI: https://doi.org/10.24321/2455.3093.202613

Best model identified is Support Vector Regression (SVR) which achieved the highest accuracy and overall best performance among all models.

It effectively captured nonlinear seismic patterns and produced stable magnitude predictions.

Conclusion

This study provides a comparative analysis of various machine learning approaches for predicting earthquake magnitude using historical seismic data from California. By evaluating models such as Linear Regression, Multiple Linear Regression, Decision Tree, KNN, SVM, Naïve Bayes, and K-Means Clustering, we found that ML-based methods can effectively capture patterns in earthquake data and offer improved prediction accuracy compared to traditional observational techniques.

Regression models and SVR emerged as strong predictors of earthquake magnitude, while classification-based approaches such as Naïve Bayes helped categorise seismic events, though with limitations for high-magnitude prediction. Clustering analysis further revealed meaningful patterns in the spatial distribution of earthquakes, demonstrating the value of unsupervised learning in seismic risk assessment.

The study confirms that machine learning can serve as a powerful tool for enhancing seismic monitoring and early warning systems. Although accurate short-term earthquake prediction remains a major scientific challenge, ML-driven models significantly contribute to risk reduction by identifying trends, hotspots, and probability estimates of seismic events. Future research can be directed toward hybrid models, deep learning architectures, integration of real-time sensor data, and ensemble techniques to further increase the reliability of seismic forecasts.

References

1. Mousavi, S. M., & Beroza, G. C. “A Machine-Learning Approach for Earthquake Magnitude Estimation.” arXiv preprint (2019).
2. Zhou, Z., Lin, Y., Zhang, Z., Wu, Y., & Johnson, P. “Earthquake Detection in 1-D Time Series Data with Feature Selection and Dictionary Learning.” arXiv (2018).
3. Wang, Y., Wang, Z., Cao, Z., & Lan, J. “Deep Learning for Magnitude Prediction in Earthquake Early Warning.” arXiv (2019).
4. Baveja, G. S., & Singh, J. “Earthquake Magnitude and b value Prediction Model Using Extreme Learning Machine.” arXiv (2023).
5. Salam, M. A., Ibrahim, L., &Abdelminaam, D. S. “Earthquake Prediction using Hybrid Machine Learning Techniques.” IJACSA, Vol.12(5), 2021.
6. Mousavi, S. M. “Machine Learning in Earthquake Seismology.” Annual Review of Earth and Planetary Sciences (2023).
7. Abebe, E. “Earthquakes Magnitude Prediction Using Deep Learning for the Horn of Africa.” ScienceDirect (2023).
8. Kaftan, I. “Machine Learning Applications for Earthquake Magnitude Prediction.” Applied Sciences, 2025.
9. Yavas, C. E., Chen, L., Kadlec, C., Ji, Y., et al. “Improving Earthquake Prediction Accuracy in Los Angeles with Machine Learning.” Scientific Reports (2024).
10. Mukherjee, B. “Earthquake Prediction Using Machine Learning: A Comparative Analysis of Six ML Classifier Methods.” ScienceDirect (2025).
11. Xu, “Prediction of Earthquake by Machine Learning Models and Feature Engineering.” ITM Conferences (2025).
12. “Earthquake Magnitude Prediction — A Machine Learning Study.” IRJMETS (May 2024).
13. Nurtas, M. “Predicting the Likelihood of an Earthquake Using Machine Learning Models.” Engineered Science (Date unspecified).
14. Mallouhy, R., et al. “Major Earthquake Event Prediction Using Various Machine Learning Algorithms.” ResearchGate (2019).
15. “Earthquake Prediction Using Machine Learning.” IRJMETS Volume 05 Issue 05 (2023), Yogaprakash MG et al.

ISSN: 2455-3093
DOI: https://doi.org/10.24321/2455.3093.202613