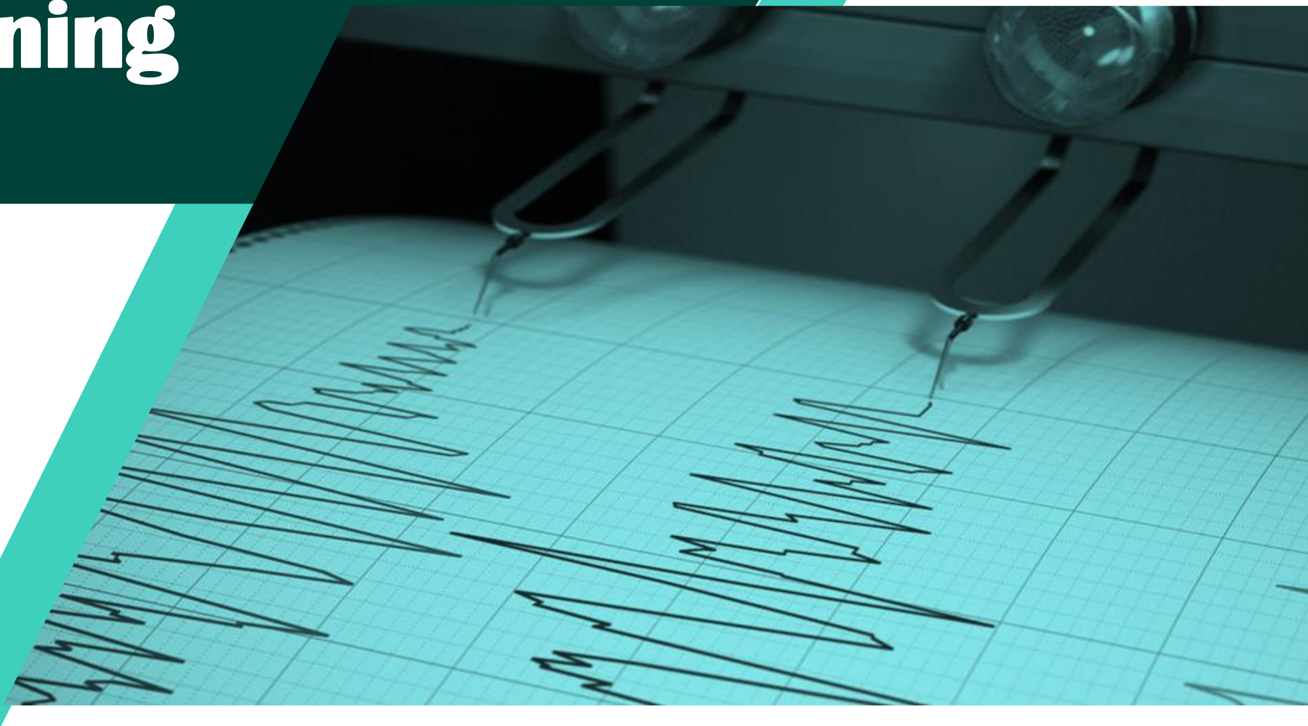
ITSOLERA
PVT LTD

Earthquake Prediction Using Machine Learning



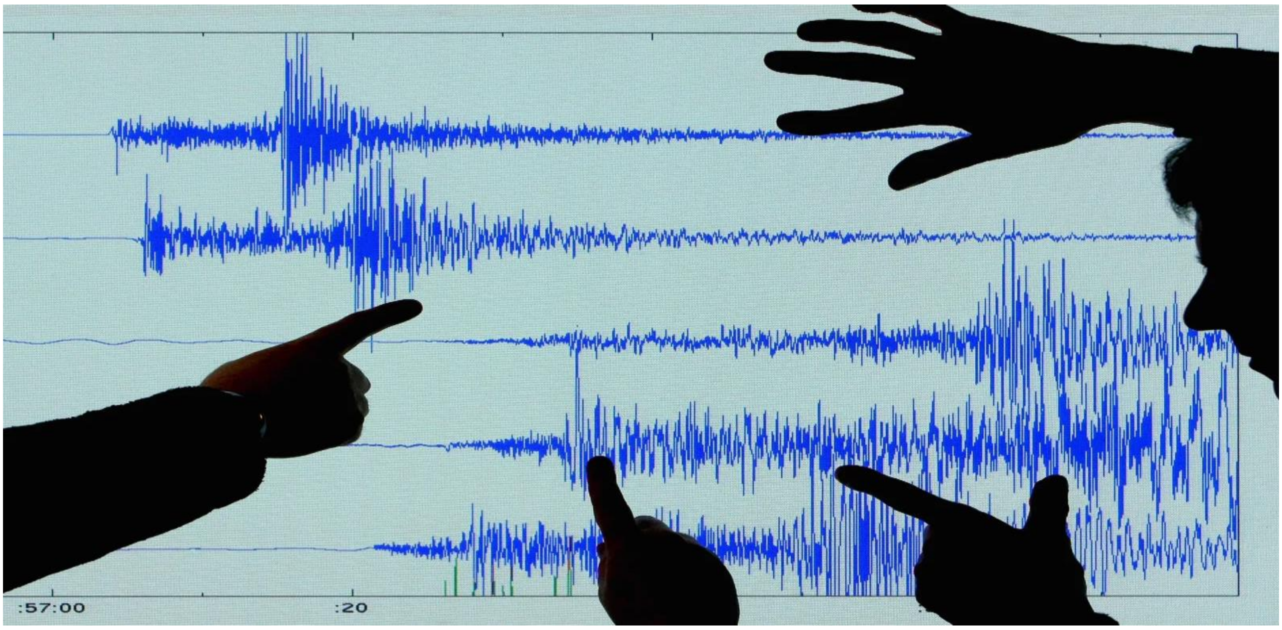
TEAM GAMMA

PROJECT: 01

TABLE OF CONTENTS

INTRODUCTION

Earthquakes are catastrophic natural events that can lead to severe damage and loss of life. Accurate prediction of earthquakes is challenging due to the complexity and variability of involved factors, but advancements in machine learning offer new opportunities for enhancing predictive capabilities. This project aims to develop a machine learning model to predict earthquakes by utilizing various data sources, including seismic activity records, geological data, and environmental factors, significantly increasing prediction accuracy. Machine learning excels at handling large datasets, recognizing patterns, processing real-time data, integrating diverse data sources, improving models through continuous learning, detecting anomalies, assessing risks, and providing scalable solutions. By leveraging these capabilities, the project seeks to develop a robust earthquake prediction model, potentially saving lives and reducing the devastating impacts of earthquakes.



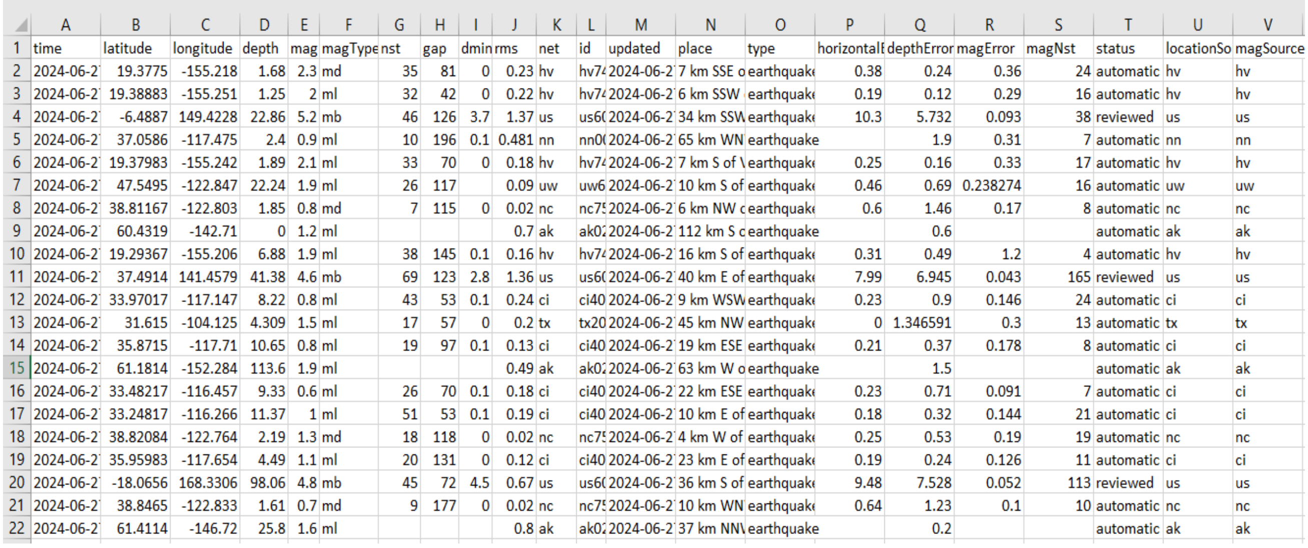
OBJECTIVES

- **Data Collection:** Acquire and pre-process data related to seismic activities and relevant geological and environmental factors.
- **Exploratory Data Analysis (EDA):** Identify patterns and correlations in the data through visualizations and statistical analysis.
- **Model Development:** Develop and train machine learning models to predict earthquake occurrences and magnitudes.
- **Model Evaluation:** Assess the models' performance using appropriate metrics and validate their predictive accuracy.
- **Deployment:** Create a user-friendly interface or dashboard to visualize and interact with the model’s predictions.

DATA COLLECTION

To collect data on recent earthquakes, the process involved utilizing the United States Geological Survey (USGS) Earthquake Hazards Program, which provides a comprehensive dataset of global earthquakes. Specifically, the earthquake data feed in CSV format was accessed via the URL `https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_month.csv`.

Using Python's panda's library, the CSV file was read into a Data Frame. This dataset includes information about all recorded earthquakes from the past month, such as their magnitude, location, depth, and time of occurrence. After loading the data, the first few rows were displayed to get an overview of its structure, and the `info () ` method was used to verify the data types and completeness. Finally, the dataset was saved to a local file named `earthquakes_all_month.csv` (Figure 1.1) for further analysis or reference.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	net	id	updated	place	type	horizontalE	depthError	magError	magNst	status	locationSo	magSource
2	2024-06-2	19.3775	-155.218	1.68	2.3	md	35	81	0	0.23	hv	hv74	2024-06-2	7 km SSE o	earthquake	0.38	0.24	0.36	24	automatic	hv	hv
3	2024-06-2	19.38883	-155.251	1.25	2	ml	32	42	0	0.22	hv	hv74	2024-06-2	6 km SSW	earthquake	0.19	0.12	0.29	16	automatic	hv	hv
4	2024-06-2	-6.4887	149.4228	22.86	5.2	mb	46	126	3.7	1.37	us	us60	2024-06-2	34 km SSW	earthquake	10.3	5.732	0.093	38	reviewed	us	us
5	2024-06-2	37.0586	-117.475	2.4	0.9	ml	10	196	0.1	0.481	nn	nn00	2024-06-2	65 km WN	earthquake		1.9	0.31	7	automatic	nn	nn
6	2024-06-2	19.37983	-155.242	1.89	2.1	ml	33	70	0	0.18	hv	hv74	2024-06-2	7 km S of \	earthquake	0.25	0.16	0.33	17	automatic	hv	hv
7	2024-06-2	47.5495	-122.847	22.24	1.9	ml	26	117		0.09	uw	uw6	2024-06-2	10 km S of	earthquake	0.46	0.69	0.238274	16	automatic	uw	uw
8	2024-06-2	38.81167	-122.803	1.85	0.8	md	7	115	0	0.02	nc	nc75	2024-06-2	6 km NW c	earthquake	0.6	1.46	0.17	8	automatic	nc	nc
9	2024-06-2	60.4319	-142.71	0	1.2	ml				0.7	ak	ak02	2024-06-2	112 km S c	earthquake		0.6			automatic	ak	ak
10	2024-06-2	19.29367	-155.206	6.88	1.9	ml	38	145	0.1	0.16	hv	hv74	2024-06-2	16 km S of	earthquake	0.31	0.49	1.2	4	automatic	hv	hv
11	2024-06-2	37.4914	141.4579	41.38	4.6	mb	69	123	2.8	1.36	us	us60	2024-06-2	40 km E of	earthquake	7.99	6.945	0.043	165	reviewed	us	us
12	2024-06-2	33.97017	-117.147	8.22	0.8	ml	43	53	0.1	0.24	ci	ci40	2024-06-2	9 km WSW	earthquake	0.23	0.9	0.146	24	automatic	ci	ci
13	2024-06-2	31.615	-104.125	4.309	1.5	ml	17	57	0	0.2	tx	tx20	2024-06-2	45 km NW	earthquake	0	1.346591	0.3	13	automatic	tx	tx
14	2024-06-2	35.8715	-117.71	10.65	0.8	ml	19	97	0.1	0.13	ci	ci40	2024-06-2	19 km ESE	earthquake	0.21	0.37	0.178	8	automatic	ci	ci
15	2024-06-2	61.1814	-152.284	113.6	1.9	ml				0.49	ak	ak02	2024-06-2	63 km W o	earthquake		1.5			automatic	ak	ak
16	2024-06-2	33.48217	-116.457	9.33	0.6	ml	26	70	0.1	0.18	ci	ci40	2024-06-2	22 km ESE	earthquake	0.23	0.71	0.091	7	automatic	ci	ci
17	2024-06-2	33.24817	-116.266	11.37	1	ml	51	53	0.1	0.19	ci	ci40	2024-06-2	10 km E of	earthquake	0.18	0.32	0.144	21	automatic	ci	ci
18	2024-06-2	38.82084	-122.764	2.19	1.3	md	18	118	0	0.02	nc	nc75	2024-06-2	4 km W of	earthquake	0.25	0.53	0.19	19	automatic	nc	nc
19	2024-06-2	35.95983	-117.654	4.49	1.1	ml	20	131	0	0.12	ci	ci40	2024-06-2	23 km E of	earthquake	0.19	0.24	0.126	11	automatic	ci	ci
20	2024-06-2	-18.0656	168.3306	98.06	4.8	mb	45	72	4.5	0.67	us	us60	2024-06-2	36 km S of	earthquake	9.48	7.528	0.052	113	reviewed	us	us
21	2024-06-2	38.8465	-122.833	1.61	0.7	md	9	177	0	0.02	nc	nc75	2024-06-2	10 km WN	earthquake	0.64	1.23	0.1	10	automatic	nc	nc
22	2024-06-2	61.4114	-146.72	25.8	1.6	ml				0.8	ak	ak02	2024-06-2	37 km NN\	earthquake		0.2			automatic	ak	ak

[Figure 1.1]

DATA PRE-PROCESSING

Data pre-processing was a crucial step in preparing the data for machine learning models. The raw data collected from USGS was meticulously processed through several steps to ensure its quality and suitability for analysis. The pre-processing steps included:

Filtering Relevant Columns:

- Only the necessary columns such as latitude, longitude, depth, and magnitude were retained from the raw dataset.
- This ensured that the dataset was focused and only included the features relevant to the machine learning task.

Handling Missing Values:

- Rows with missing values were identified and removed from the dataset.
- This step was essential to maintain data integrity and prevent any bias or errors that missing values could introduce in the analysis.

Standardizing Numerical Features:

- The numerical features were standardized to have a mean of zero and a standard deviation of one.
- Standardization helped in improving the performance of many machine learning algorithms by ensuring that all features contributed equally to the analysis and were on a comparable scale.

Splitting the Data:

- The dataset was divided into two subsets: training and testing sets.
- 80% of the data was allocated for training the model, while the remaining 20% is reserved for testing.
- This split allowed for the evaluation of the model's performance on unseen data, ensuring its generalizability.

EDA

EDA is a crucial step in any machine learning project as it helps in understanding the underlying patterns, relationships, and anomalies within the data. For this earthquake prediction project, EDA involved several steps, including visualizing the data distributions, identifying correlations, and detecting missing values.

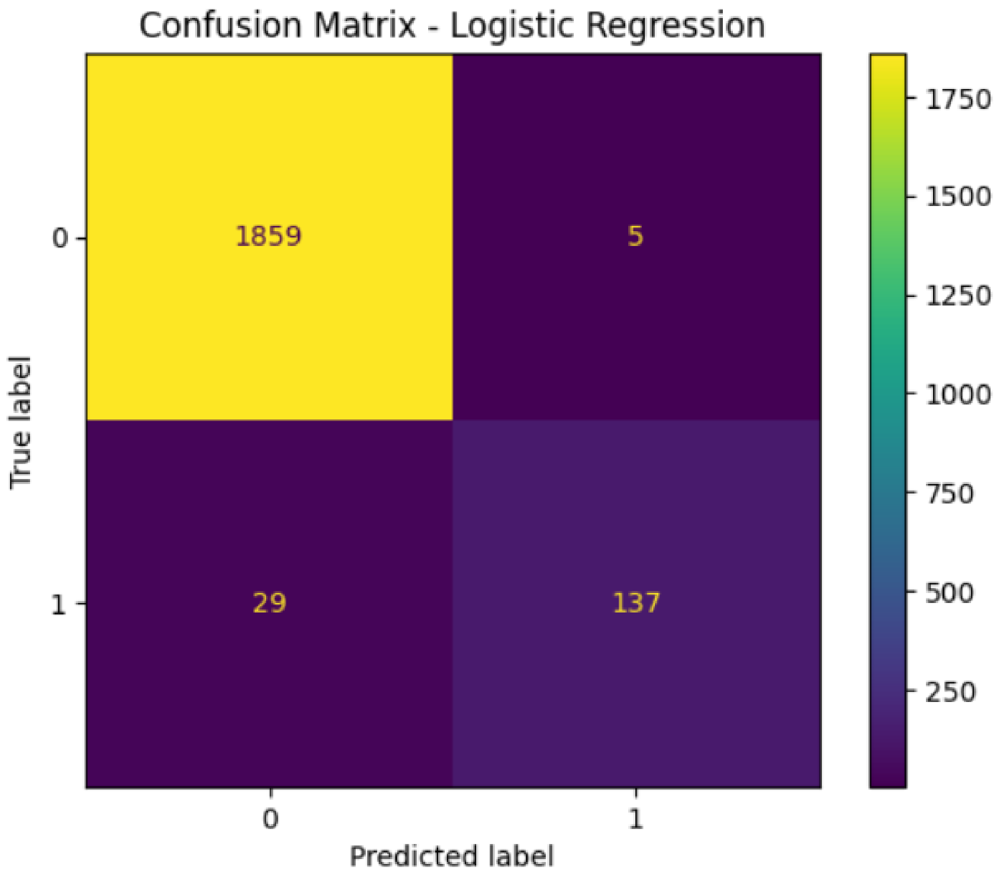
Data Visualization

Data visualization was carried out using libraries such as **Matplotlib** and **Seaborn**. These libraries provide a range of tools to create informative and aesthetically pleasing charts and graphs. Seaborn, in particular, is built on top of Matplotlib and offers a high-level interface for drawing attractive statistical graphics.

- **Confusion Matrix:** The confusion matrix provided a detailed breakdown of the logistic regression model's performance, showing the number of instances correctly and incorrectly classified. It visualized true positives, true negatives, false positives, and false negatives, effectively evaluating the model's accuracy in classification tasks.
- **Scatter Plot:** Scatter plots were used to visualize predictions from various models (Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, SVM, Neural Network) against the true values of the target variable. They provided a visual comparison of predicted values versus actual values, crucial for assessing the accuracy of regression models.
- **Bar Plot with Line Plot:** This combined plot allows for a comprehensive comparison of performance metrics (Mean Squared Error and R2 Score) across different models, helping in selecting the best-performing model.

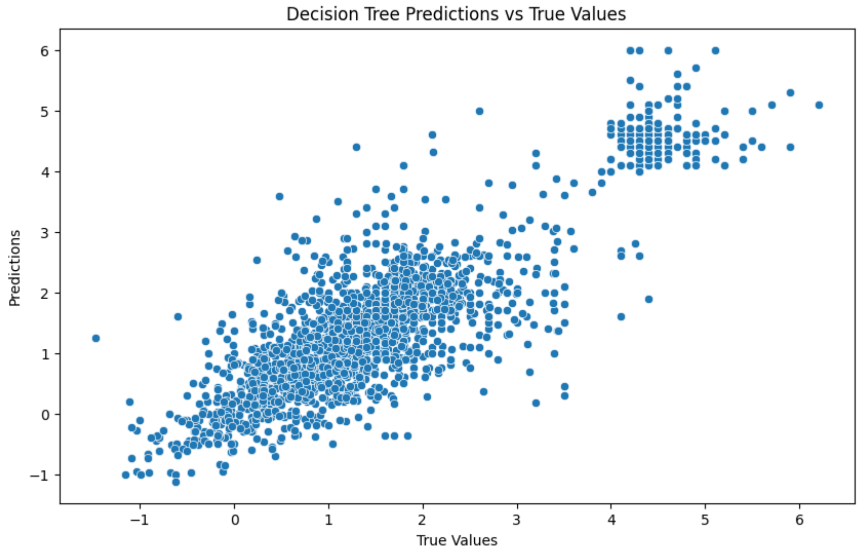
Logistic Regression

Logistic Regression was employed to classify earthquakes into two categories based on their magnitude: less than 4.0 and 4.0 or greater. This binary classification approach simplified the problem and allowed for the evaluation of the model's ability to distinguish between minor and significant earthquakes. Logistic Regression was chosen for its simplicity, interpretability, and efficiency in binary classification tasks. The model was trained on the standardized training data, and its performance was evaluated using Mean Squared Error (MSE) and R² score on the test data. Additionally, a confusion matrix was plotted to visualize the classification results.

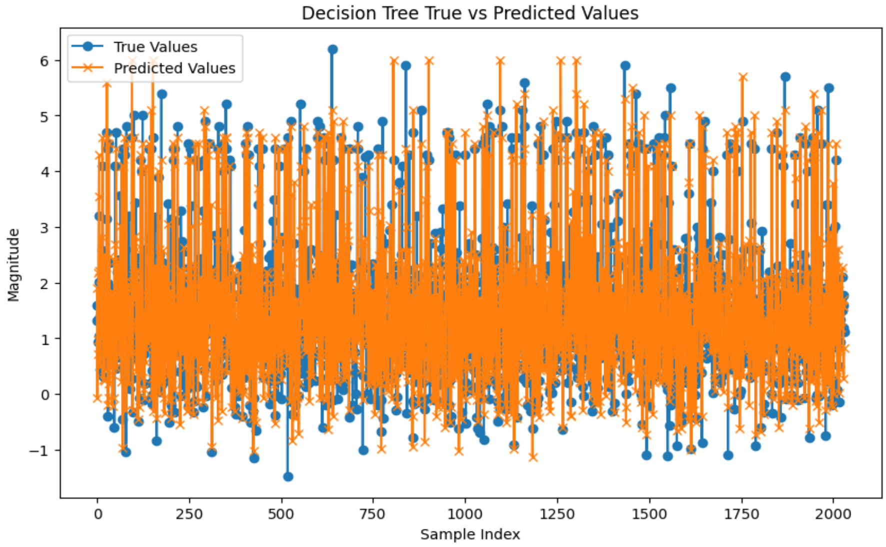


Decision Tree

A Decision Tree Regressor was trained to predict the continuous magnitude values of earthquakes. This model was selected for its straightforward nature and interpretability, making it a good baseline for comparison. Decision Trees are capable of capturing non-linear relationships in the data and can handle both numerical and categorical features. During training, the model learned from the standardized data to make predictions on earthquake magnitudes.

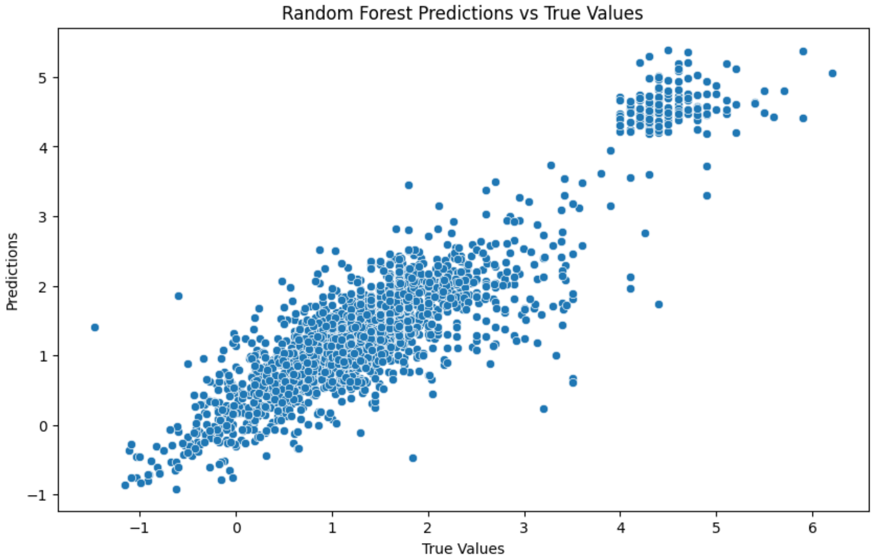


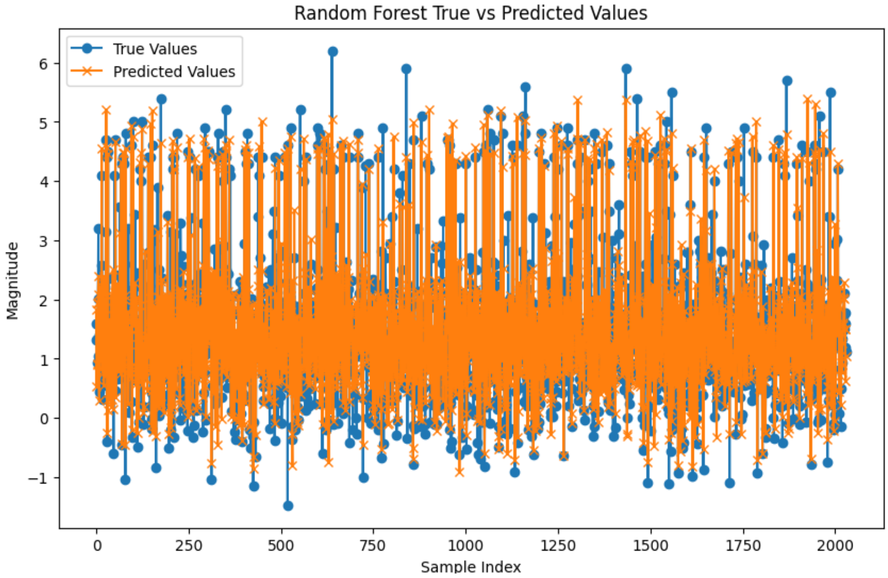
The performance of the Decision Tree model was evaluated using MSE and R² score, which provided a measure of the model’s prediction error and goodness of fit, respectively. This evaluation helped in understanding the basic capability of the model to predict earthquake magnitudes.



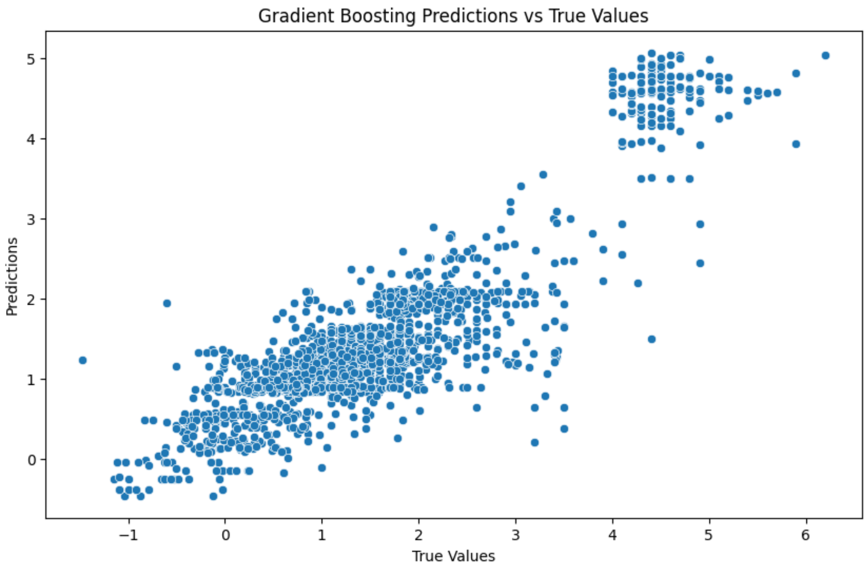
Random Forest

The Random Forest Regressor, an ensemble learning method, was used to build multiple decision trees and merge their predictions to improve accuracy and control overfitting. By training multiple trees on random subsets of the data and combining their outputs, the model aimed to enhance prediction accuracy and robustness. This model was trained on the standardized data, and its performance metrics were recorded, including MSE and R² score. Random Forests are particularly robust to overfitting, making them suitable for handling large datasets and complex interactions between features. The ensemble approach also helped in reducing variance and improving the reliability of the predictions.

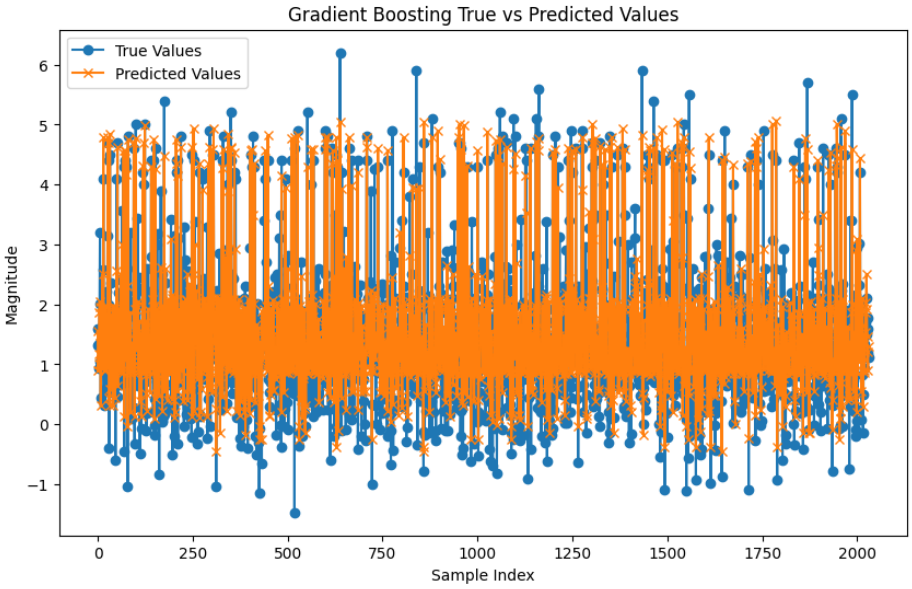




Gradient Boosting

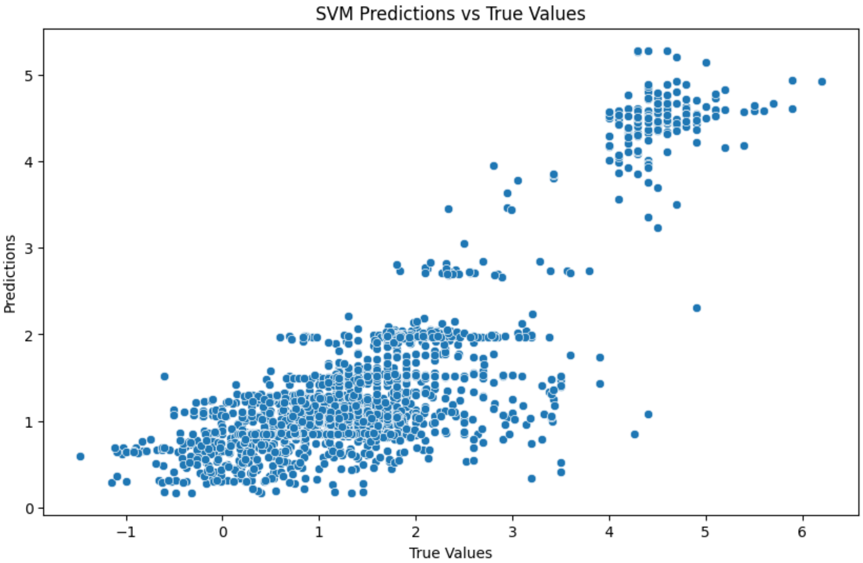


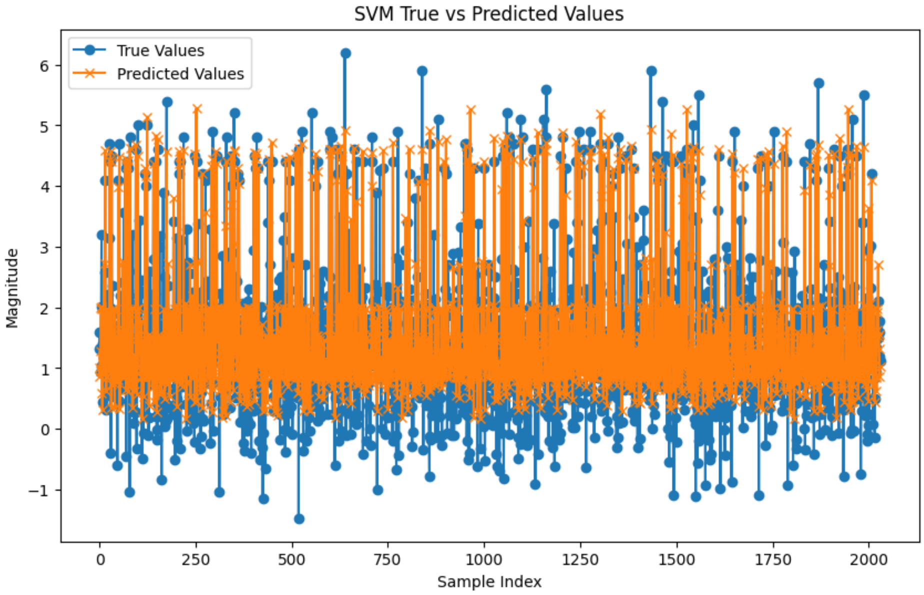
Gradient Boosting, another powerful ensemble method, was employed to build models sequentially, each trying to correct the errors of the previous one. The Gradient Boosting Regressor was trained and evaluated on the dataset, with performance metrics compared to other models. Gradient Boosting is particularly effective in handling imbalanced datasets and improving model accuracy through boosting weak learners. This iterative process allowed the model to progressively minimize prediction errors and enhance its performance. The evaluation using MSE and R² score provided a comprehensive understanding of the model's efficiency and accuracy in predicting earthquake magnitudes.



Support Vector Machine (SVM)

The SVM model, specifically the SVR (Support Vector Regressor), was used for predicting the continuous magnitude values. SVMs are effective in high-dimensional spaces and are known for their robustness, making them suitable for complex prediction tasks like earthquake magnitudes. The model's performance was evaluated in terms of

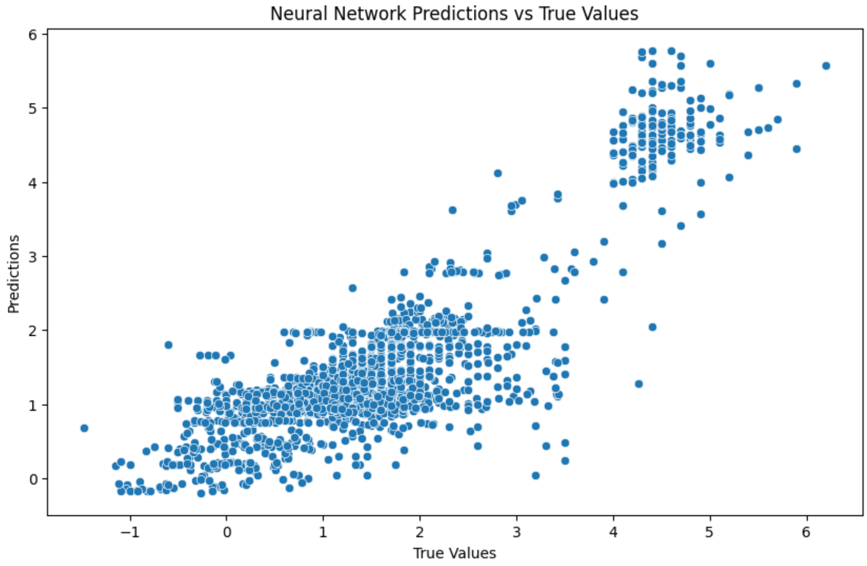


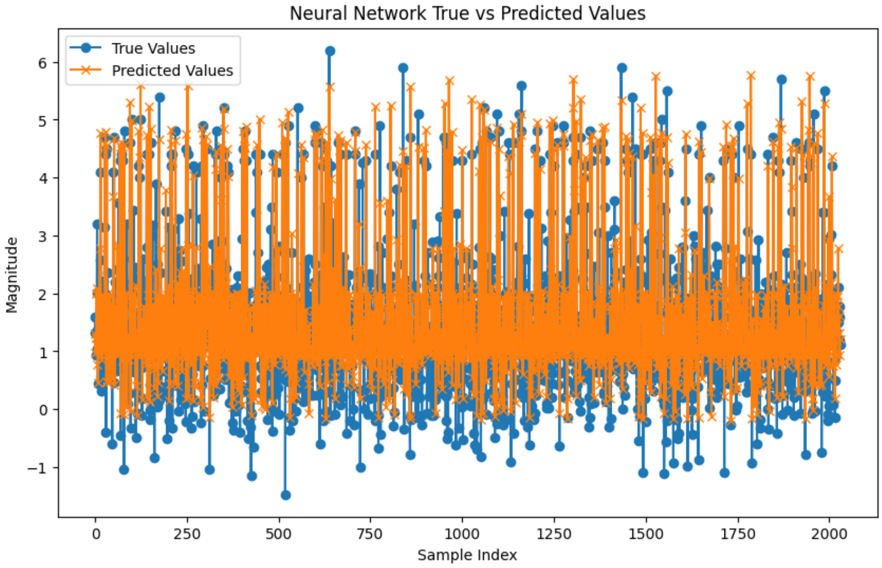


MSE and R² score, providing insights into its prediction accuracy and fit to the data. SVMs were chosen for their ability to handle non-linear relationships through the use of kernel functions, which allowed the model to capture complex patterns in the data and make accurate predictions.

Neural Network

A Neural Network model was built using TensorFlow and Keras. The network comprised multiple dense layers with ReLU activation functions, optimized using the Adam optimizer. The model was trained over several epochs, allowing it to learn and model complex, non-linear relationships in the data. Its predictions were compared against the true values using MSE and R² score, providing a measure of the model's accuracy and fit. Neural Networks were selected for their ability to capture intricate patterns and interactions within the data, making them powerful tools for predicting earthquake magnitudes. Visualization of the predictions against true values was provided to illustrate the model's performance and highlight areas for potential improvement.





MODEL COMPARISON

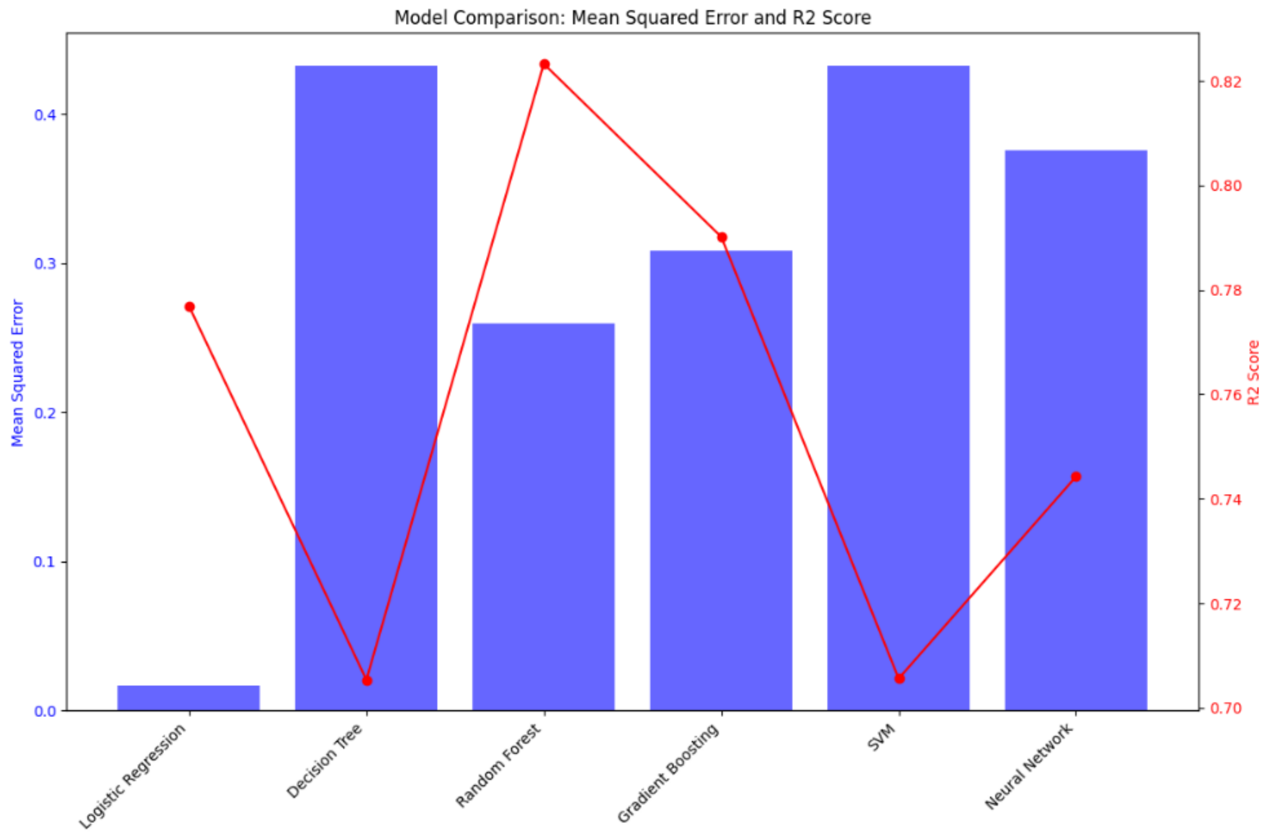
The performance of different machine learning models in predicting earthquake magnitudes was evaluated using Mean Squared Error (MSE) and R2 Score. The results are summarized in the table and visualized in the bar graph below:

Model	Mean Squared Error	R2 Score
Logistic Regression	0.016749	0.776940
Decision Tree	0.432765	0.705375
Random Forest	0.259584	0.823276
Gradient Boosting	0.308219	0.790166
SVM	0.432421	0.705609
Neural Network	0.375607	0.744288

- **Logistic Regression** demonstrated the lowest MSE of 0.016749, indicating high accuracy, but had a moderate R2 Score of 0.776940.
- **Decision Tree** had the highest MSE of 0.432765 and a lower R2 Score of 0.705375, suggesting it struggled to capture the complexity of the data.
- **Random Forest** showed a good balance with an MSE of 0.259584 and the highest R2 Score of 0.823276, making it one of the best-performing models.
- **Gradient Boosting** had a slightly higher MSE of 0.308219 compared to Random Forest but maintained a strong R2 Score of 0.790166.
- **SVM** had a high MSE of 0.432421 and a lower R2 Score of 0.705609, indicating it might not be the best choice for this problem.
- **Neural Network** performed decently with an MSE of 0.375607 and an R2 Score of 0.744288, showing potential with further tuning.

BEST MODEL

The Random Forest model stood out as the most robust predictor of earthquake magnitudes, balancing both low MSE and high R2 Score.

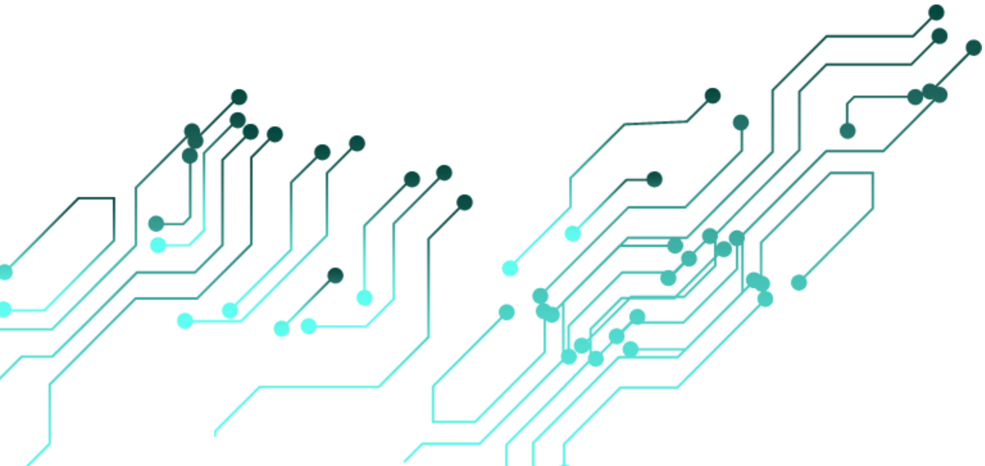


The bar graph above visualizes these results, highlighting the trade-offs between Mean Squared Error and R2 Score for each model. The blue bars represent the Mean Squared Error, and the red line represents the R2 Score.

CONCLUSION

This project demonstrates the application of various machine learning models to predict earthquake magnitudes using real-world data from the USGS. The results indicate that complex models such as Neural Networks, Random Forest, and Gradient Boosting outperform simpler models like Logistic Regression and Decision Tree in this context. The project highlights the importance of data pre-processing, model selection, and thorough evaluation in building robust predictive models.

Future work could involve exploring additional features, more sophisticated neural network architectures, and advanced techniques such as hyperparameter tuning to further improve prediction accuracy. Additionally, incorporating domain-specific knowledge and real-time data streaming could enhance the model's applicability in practical earthquake monitoring and early warning systems. The team also plans to add deployment strategies to ensure the models are effectively utilized in real-world applications.



WORK DISTRIBUTION

- Data Collection - Toseeq Haider Bajwa
- Data Preprocessing - Rehan Sarfraz
- Exploratory Data Analysis (EDA) – Luqman Ali Imran
- Model Development - Abdul Aziz & Muhammad Awais Ahmed
- Model Evaluation – Ayesha Iqbal & Hadiya Asif
- Deployment & Revisions – Muhammad Rizwan Khan (Team Lead)
- Presentation and Report – Ayesha Iqbal & Hadiya Asif