Abstract: An earthquake is a type of natural disaster that is well-known for the devastation it causes to both naturally existing and artificial structures, including buildings, bungalows, and residential areas, to name a few. Seismometers, which pick up vibrations caused by seismic waves moving through the earth's crust, are used to measure earthquakes. The damage caused by an earthquake was categorised in this work into damage ratings, which have values ranging from one to five. The damage grade of a certain structure, which is linked to a Unique Identification String, was predicted using a previously gathered data set and a number of criteria. An analysis of current machine learning classifier techniques was used to make the forecast. Logistic Regression, Support Vector Machine (SVM), Random Forest Classifier, and K-Nearest Neighbors were the machine learning techniques employed in this study. The best algorithm was taken into consideration after a review of a number of attributes. The method used to predict the property underwent a thorough investigation, and the data analysis that followed revealed information that could help future earthquakes' effects be lessened.

Keywords: Machine learning, Support Vector Machine (SVM), Random Forest Classifier, Logistic Regression, K Nearest Neighbors, and predictive analysis.

I. INTRODUCTION

A catastrophic event such as an earthquake is harmful to human interests and has negative effects on the environment. Incalculable harm to buildings and other assets has always been done by earthquakes, which have also claimed millions of lives around the world. Numerous national, international, and transnational organizations implement various disaster warning and preventive strategies to lessen the effects of such an incident. Organization managers have a number of challenges when it comes to allocating the organization's resources because time and quantity are constraints. To estimate the extent of damage done to buildings after an earthquake, it is possible to use machine learning. This is accomplished by categorizing these buildings according to a degree of damage severity based on a number of elements, including their age, foundation, number of floors, kind of material used, and others. Then, ward-by-ward in a district, the number of families and the likely casualties are considered. This enables the proportionate distribution of relief forces by ward and their prioritizing according to the severity of the damage. Such models can contribute to the fastest possible lifesaving and prove to be a successful and affordable option.[1-3] It can be further enhanced by include the distribution of goods like food, clothing, medical care, and money in accordance with the number of fatalities among people and the degree of structural damage.

II. BACKGROUND

From its foundations in databases, statistics, applied science, theory, and algorithms, the discipline of machine learning has developed into a core set of approaches that are applied to a variety of issues. Over the past 20 years, significant advancements have been made in the scientific and technical fields of computational modelling and data gathering. Additional data repositories have been produced as a result of a combination of sophisticated algorithms, exponentially rising processing power, and precise sensing and measurement tools. Networks with cutting-edge technologies have made it possible to send enormous amounts of data around the globe. This leads to an extreme need for tools and technologies in order to analyse scientific data sets effectively with the aim of deciphering the underlying physical events. Machine Learning applications in geology and geophysics have achieved significant success within the areas as weather prediction, mineral prospecting, ecology, modelling etc and eventually predicting the earthquakes from satellite maps.[4] An interesting aspect of the numerous of these applications is that they combine both spatial and temporal aspects within the info and within the phenomena that's being mined. Investigations on earthquake predictions are supported the concept that each one amongst the regional factors is filtered out and general information about the earthquake precursory patterns is extracted. Feature extraction involves a pre-selection process of varied statistical properties of data and generation of a group of seismic parameters, which correspond to linearly independent coordinator within the feature space. The seismic within the sort of statistic are often analysed by using various pattern

III. MOTIVATION

Earthquakes are one amongst the foremost destructive natural disasters. They typically occur without notice and do not allow much time for people to react. Therefore, earthquakes can cause serious injuries and loss of life and destroy tremendous buildings and infrastructures, resulting in great economy loss. The prediction of earthquakes is clearly critical to the protection of our society, but it's proven to be a Challenging task to predict beforehand and yet we discover this as a motivating problem to be solved.[6]

IV. SOFTWARE REQUIREMENT SPECIFICATION

i. Anaconda Navigator 2.3.2
ii. Jupyter Notebook 6.0.3
iii. Tkinter
iv. OpenCV
v. Pycharm 2022.2.4
vi. Language: Python 3.7
vii. Environment: Keras and Tensorflow environment
viii. OS: Windows 7 or higher

V. METHODOLOGY

A. Importing Libraries

Figure 1 shows the Python code to import libraries. We have used four libraries

• Python has a library called Numpy that is used for scientific computing. This library is utilized throughout the project and is imported as np.

• Pandas are used for data analysis and manipulation. An open source, BSD-licensed library called pandas offers simple data structures and tools for data analysis. It is imported as pd.

• matplotlib is a python library. The command-style utilities in pyplot enable matplotlib to behave similarly to MATLAB. It is imported as plt.

• Seaborn is a matplotlib-based Python data visualization package for aesthetically pleasing and educational statistical visuals. It is imported as sns.

B. Importing data

Figure 2 displays the Python code for importing data from the appropriate directory or file and allocating it to a DataFrame. It imports the data that is kept in CSV format.

C. Checking for NaN

Checking for NaN is a critical step in the pre-processing of data. We were only able to identify a few NaNs in this test. The Python code to check for NaN is displayed in Figure 3.

D. Manipulating NaN values

It is essential to remove the NaN values. This can be done by

- Removing the entire column containing many NaN values
- Forward fillna method
- Backward fillna method
- Mean method

Figure 4 shows the technique of forward fillna method.

E. Plotting a Heatmap

A heatmap is used to assess the correlation between the fields of the collected data. When developing various AI prediction models, the magnitude of the values along with the sign (which may be negative or positive) is crucial. Figure 5 displays a correlation model and heatmap.

F. Train/Test split

Creating train and test sets from the data is our next step towards developing a Machine Learning model. The Python code to divide the data set into train and test data is shown in Figure 6.

In [20]:

```
from sklearn.model_selection import train_test_split
```

In [77]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=12)
```

Figure 6 shows the python code to split the data set into train and test data.

In [82]:

```
from sklearn.linear_model import LogisticRegression
```

In [85]:

```
logmodel= LogisticRegression()
logmodel.fit(X_train,y_train)
```

Figure 7 shows logistic regression on given data set.

Out[85]:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='warn',
          tol=0.0001, verbose=0, warm_start=False)
```

Figure 8 shows the results of logistic regression model.

In [86]:

```
predictions= logmodel.predict(X_test)
predictions
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,predictions)
from sklearn.metrics import accuracy_score
accuracy_score(y_test,predictions)
```

Out[86]:

```
0.9833333333333333
```

Figure 9 shows the Accuracy score of the designed model.

G. Applying Classification algorithms

Classification algorithms such as SVM, Random Forest Classifier and KNN were applied to the dataset and accuracy of each model has been described in the figure 10 of our paper. Logistic Regression had a higher accuracy rate when the data was acquired at the initial stages and with a lot of columns dropped.[8-11] But, the accuracy decreased with the columns back into the train/ test split making it not suitable for consideration. As a result classification algorithms had to be considered. Based on the model a 0-5 rating is given for the earthquakes.

pandas.DataFrame.fillna

`DataFrame.fillna(value=None, method=None, axis=None, inplace=False, limit=None, downcast=None, **kwargs)`

Fill NA/NaN values using the specified method. [source]

Parameters:

value : *scalar, dict, Series, or DataFrame*
Value to use to fill holes (e.g. 0), alternately a dict/Series/DataFrame of values specifying which value to use for each index (for a Series) or column (for a DataFrame). (values not in the dict/Series/DataFrame will not be filled). This value cannot be a list.

method : *{'backfill', 'bfill', 'pad', 'ffill', None}, default None*
Method to use for filling holes in reindexed Series pad / ffill: propagate last valid observation forward to next valid backfill / bfill: use NEXT valid observation to fill gap

axis : *{0 or 'index', 1 or 'columns'}*

inplace : *boolean, default False*
If True, fill in place. Note: this will modify any other views on this object, (e.g. a no-copy slice for a column in a DataFrame).

limit : *int, default None*
If method is specified, this is the maximum number of consecutive NaN values to forward/backward fill. In other words, if there is a gap with more than this number of consecutive NaNs, it will only be partially filled. If method is not specified, this is the maximum number of entries along the entire axis where NaNs will be filled. Must be greater than 0 if not None.

downcast : *dict, default is None*
a dict of item->dtype of what to downcast if possible, or the string 'infer' which will try to downcast to an appropriate equal type (e.g. float64 to int64 if possible)

Returns:

filled : *DataFrame*

Figure 4 shows the technique of forward fillna method.

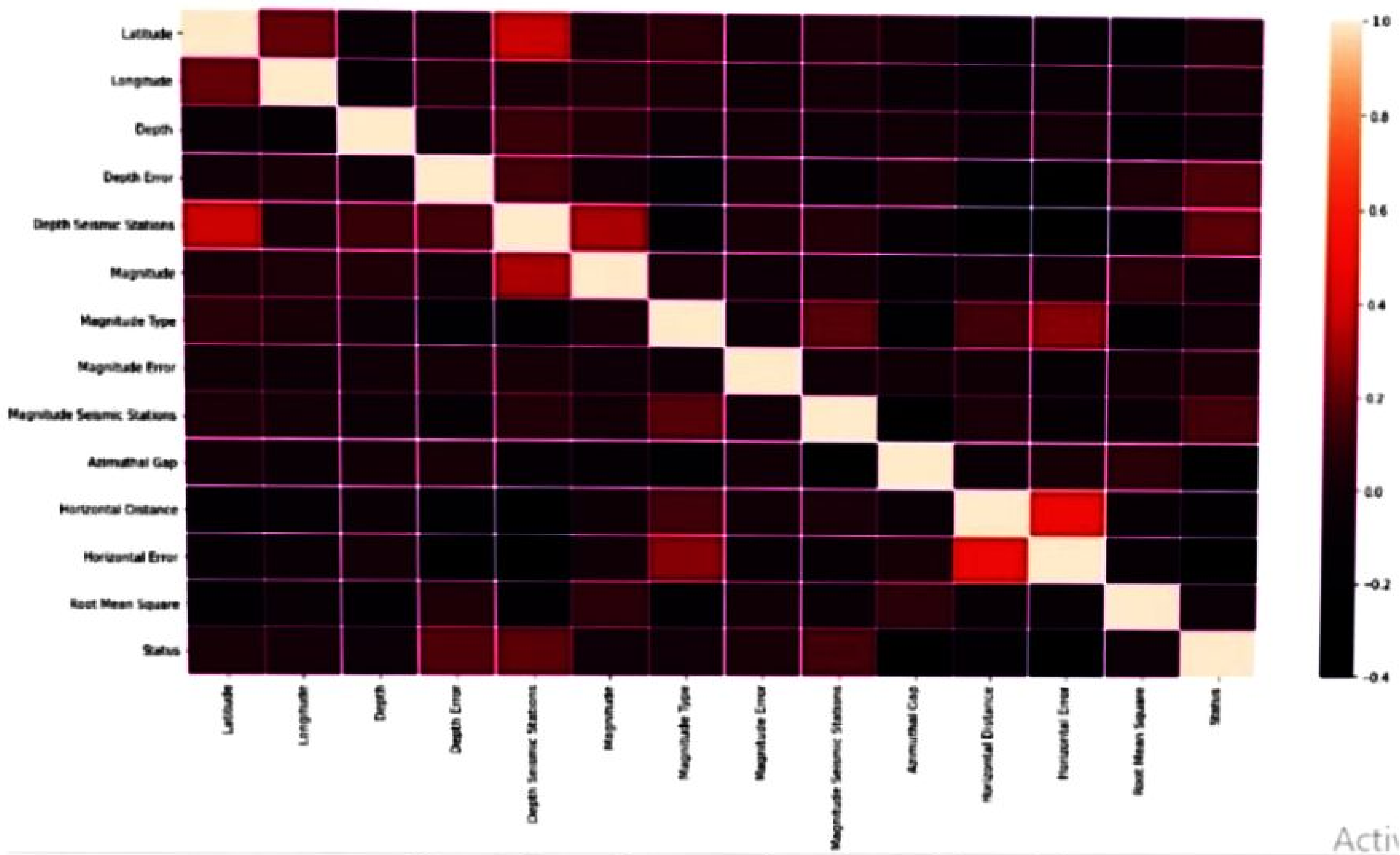

Figure 5 shows a heatmap and correlation of the model.

In [2]:

```
import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
import matplotlib.pyplot as plt
```

Figure 1 shows the Python code to import libraries.

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
df=pd.read_csv(r"C:\Users\DELL\Desktop\earth.csv")
```

```
df
```

	Date	Time	Latitude	Longitude	Depth	Depth Error	Depth Seismic Stations	Magnitude	Magnitude Type	Magnitude Error	Magnitude Seismic Stations	Azimuti G
0	1/1/2000	5:58:20	-60.7220	153.6700	10.00	4.7	158	6.0	0	0.030	4	228.
1	1/2/2000	12:14:39	-17.9430	-178.4760	582.30	4.7	473	5.5	0	0.041	4	228.
2	1/2/2000	12:58:42	51.4470	-175.5580	33.00	4.7	149	5.8	0	0.071	4	228.
3	1/2/2000	15:16:32	-20.7710	-174.2360	33.00	4.7	169	5.8	1	0.000	4	228.
4	1/5/2000	7:32:19	-20.9640	-174.0970	33.00	4.7	116	5.6	0	0.045	4	228.

Figure 2 shows the Python code to import data and assigning it to DataFrame df.

Applying Regression

Figure 7 shows logistic regression on given data set. Figure 8 displays the findings from the logistic regression model. The accuracy rating of the created model is displayed in Figure 9.[8][9]

```
df.isnull()
```

	Date	Time	Latitude	Longitude	Depth	Depth Error	Depth Seismic Stations	Magnitude	Magnitude Type	Magnitude Error	Magnitude Seismic Stations	Azimuthal Gap	Horizontal Distance	Horizontal Error	Root Mean Square	Statu
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	Fals
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	Fals
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	Fals
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	Fals
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	Fals
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8739	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	Fals
8740	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	Fals
8741	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	Fals
8742	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	Fals
8743	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	Fals

Figure 3 shows the Python code to check for NaN.

In [5]:

```
df.drop(["year resalevalue"], axis=1, inplace= True)
```

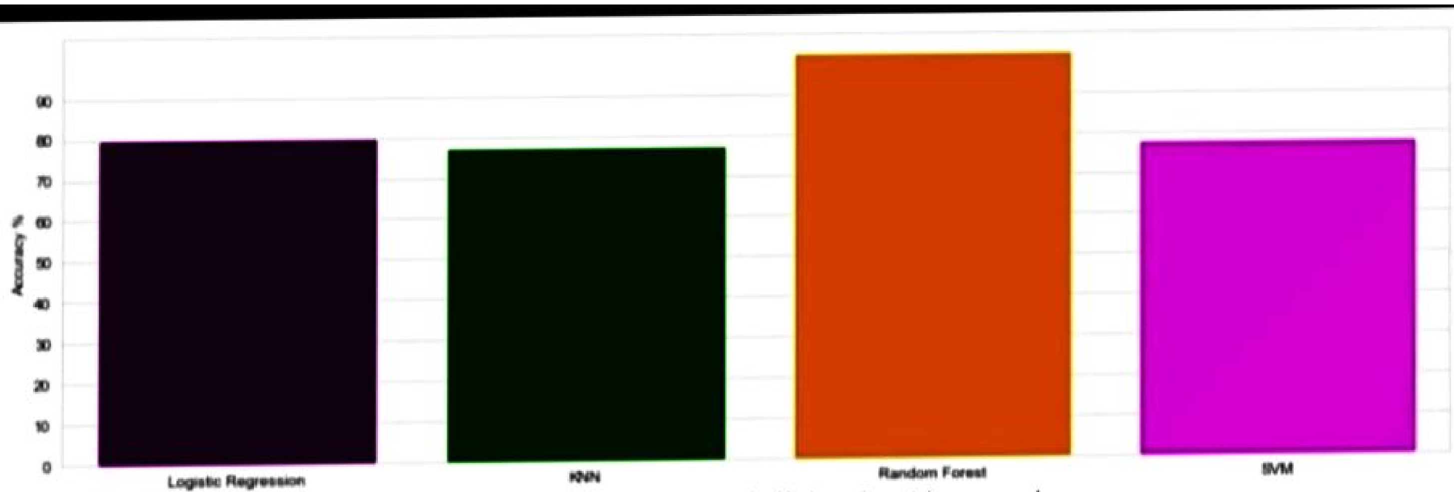

Figure 10 shows the accuracy plot of all the algorithms used.

VI. CONCLUSIONS AND FUTURE WORK

Based on the Accuracies and F1 scores determined for each of the four algorithms previously discussed in this study, this analysis demonstrates that the Random Forest Classifier method has the highest accuracy in forecasting the damage caused by earthquakes. It has been noted that Logistic Regression is the second most used method for predicting earthquake damage. The research finds that reinforced concrete is the material best suited to preventing damage to structures during an earthquake after analysing the materials that can do so. It is commonly known that earthquakes trigger electromagnetic pulses that induce tremors beneath the Earth's crust. Reinforced concrete adequately shields these electromagnetic pulses. Due to the low tensile strength of reinforced concrete, steel bars that are implanted in the concrete are used. The applications of this work can be further extended to predict damage caused by Earthquakes in areas for which a similar and relevant dataset can be obtained and crack analysis can be done using Neural Networks.[12]

REFERENCES

[1] C.P Shabariram and K.E Kannammal "Earthquake prediction using map reduce framework" 2017 International Conference on Computer Communication and Informatics (ICCCI), pp. 1–6, 2017.

[2] You-Po Su and Qing-Jie Zhu "Application of ANN to Prediction of Earthquake Influence" 2009 Second International Conference on Information and Computing Science, vol. 2 pp. 234-237, 2009.

[3] Cao Li and Xiaoyu Liu, "An improved PSO-BP neural network and its application to earthquake prediction," in 2016 Chinese Control and Decision Conference 2016, pp. 3434–3438.

[4] Xueli Wei, Xiaofeng Cui, Chun Jiang, Xinbo Zhou "The Earthquake Probability prediction based on weighted factor coefficients of principal components "2009 Fifth International Conference on Natural Computation, vol. 2, p. 608-612, 2009

[5]Dezhang Sun, Baitao Sun "Rapid prediction of earthquake damage to buildings based on fuzzy analysis"2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery, vol. 3, p. 1332-1335, 2010

[6] Jianwen Zhang, Changyong Wang "Sheer Capacity computation of steel reinforced lightweight concrete beams" International Conference on Mechanic Automation and Control Engineering, 2010. p 1502- 1506

[7] The Hackerearth platform official website https://www.hackerearth.com

[8] Scikit-Learn documentation- "https://scikitlearn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html"

[9] Long Wang, Xiaoqing Wang, Aixia Dou, Dongliang Wang "Study on construction seismic damage loss assessment using RS and GIS" International Symposium on Electromagnetic compatibility, 2014.

[10] Ramli Adnan. Abd Manan Samad, Zainazlan Md Zain, Fazlina Ahmat Ruslan "5 hours flood prediction modeling using improved NNARX structure: case study Kuala Lumpur", IEEE 4th International Conference on System Engineering and Technology, 2014.

[11] H Takata, H. Nakamura, T Hachino "On prediction of electric power damage by typhoons in each district in Kagoshima Prefecture via LRM and NN", SICE Annual Conference, 2004.

[12] Vishesh S, D S Pavan, Rishi Singh, Rakesh Gowda B, 2022. Prediction of Diabetic Retinopathy using Neural Networks.

[13] Vishesh S, Rachana S, Chethan K, Harshitha V Raj, 2022. AI to Predict Diabetic Retinopathy: Image Pre- Processing and Matrix Handling.

[12] Vishesh S, Sujay Singh, Rishi Singh, 2021. CNN Algorithm: H5 model for Accurate Prediction of COVID-19.