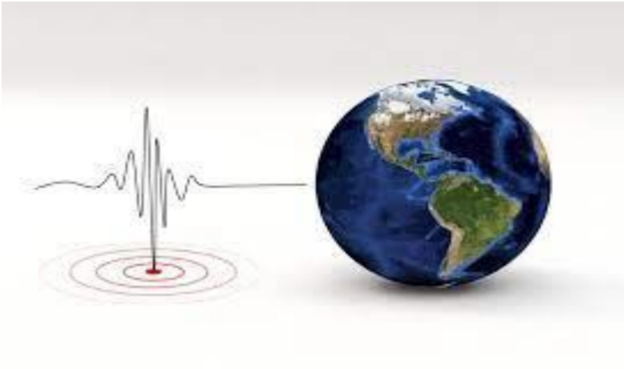
Earthquake prediction model using python



Problem Statement:

Earthquakes are one of the most destructive and unpredictable natural disasters in the world, causing significant damage to infrastructure, property, and human life. While scientists have made significant progress in predicting earthquakes, there is still much that is unknown about these events, including their magnitude, location, and timing.

The goal of this project is to develop an accurate and reliable machine learning model for predicting the severity of earthquakes based on a range of input variables, including seismic activity, geographical location, and historical earthquake data. To achieve this, the project will employ a range of ensemble techniques and perform cross-validation to evaluate the performance of the model.

By using ensemble techniques and cross-validation, the project aims to develop a highly accurate and reliable model that can be used to improve earthquake preparedness and response efforts in areas prone to seismic activity. Overall, this project aims to use advanced machine learning techniques to develop a highly accurate and reliable model for earthquake prediction. By doing so, it hopes to contribute to our understanding of these complex natural events and to ultimately help save lives and minimize the damage caused by earthquakes

Dataset Link: **https://www.kaggle.com/datasets/usgs/earthquake-database**.

Dataset:

	title	magnitude	date_time	cdi	mmi	alert	tsunami	sig	net	nst	dmin	gap	magType	depth	latitu
0	M 7.0 - 18 km SW of Malango, Solomon Islands	7.0	22-11-2022 02:03	8	7	green	1	768	us	117	0.509	17.0	mww	14.000	-9.7
1	M 6.9 - 204 km SW of Bengkulu, Indonesia	6.9	18-11-2022 13:37	4	4	green	0	735	us	99	2.229	34.0	mww	25.000	-4.9
2	M 7.0 -	7.0	12-11-2022 07:09	3	3	green	1	755	us	147	3.125	18.0	mww	579.000	-20.
3	M 7.3 - 205 km ESE of Neiafu, Tonga	7.3	11-11-2022 10:48	5	5	green	1	833	us	149	1.865	21.0	mww	37.000	-19.
4	M 6.6 -	6.6	09-11-2022 10:14	0	2	green	1	670	us	131	4.998	27.0	mww	624.464	-25.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
777	M 7.7 - 28 km SSW of Puerto El Triunfo, El Sal...	7.7	13-01-2001 17:33	0	8	NaN	0	912	us	427	0.000	0.0	mwc	60.000	13.0
778	M 6.9 - 47 km S of Old Harbor, Alaska	6.9	10-01-2001 16:02	5	7	NaN	0	745	ak	0	0.000	0.0	mw	36.400	56.7

<u>Creating models:</u>

1ST model-Using Logistic Regression

from sklearn.linear_model import LogisticRegression

l1=LogisticRegression()

l1.fit(x_train,y_train)

```
▾ LogisticRegression
LogisticRegression()
```

y_pred=l1.predict(x_test)

y_pred

```
array([1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0,
       1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0,
       1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1,
       1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1,
       1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1,
       0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0,
       1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,
       1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0])
```

from sklearn.metrics import accuracy_score

ac=accuracy_score(y_test,y_pred)*100

```
81.25
```

<u>2ND Model -using SVM</u>

from sklearn.svm import SVC

SVM=SVC(kernel="linear",random_state=2)

SVM.fit(x_train,y_train)

```
▾ SVC
SVC(kernel='linear', random_state=2)
```

y_pred1=SVM.predict(x_test)

```
array([1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1,
       0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1,
       1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0,
       1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1,
       1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1,
       1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1,
       0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0,
       1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0,
       1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0])
```

ac1=accuracy_score(y_test,y_pred1)*100

```
82.29166666666666
```

3RD Model -using Gaussian Naive Bayes

from sklearn.naive_bayes import GaussianNB

nb=GaussianNB()

nb.fit(x_train,y_train)

```
▾ GaussianNB
GaussianNB()
```

y_pred2=nb.predict(x_test)

```
array([1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0,
       0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0,
       1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0,
       1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1,
       1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1,
       1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1,
       0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0,
       1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,
       1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0])
```

ac2=accuracy_score(y_test,y_pred2)*100

```
79.6875
```

4th Model - using Decision Tree Classifier

from sklearn.tree import DecisionTreeClassifier

dt = DecisionTreeClassifier()

dt.fit(x_train, y_train)

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

y_pred3=dt.predict(x_test)

```
array([1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1,
       0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1,
       1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0,
       1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1,
       1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1,
       1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1,
       0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0,
       0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0,
       1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0])
```

ac3 = accuracy_score(y_test, y_pred3)*100

```
93.75
```

Ensemble Technique

from sklearn.ensemble import VotingClassifier

bc=VotingClassifier(estimators=[("logisticRegression",l1),("svm",SVM),("naivebayes",nb),("Decision Tree Classification", dt)])

bc.fit(x_train,y_train)

```
                      VotingClassifier
 logisticRegression    svm    naivebayes  Decision Tree Classification
 LogisticRegression    SVC    GaussianNB   DecisionTreeClassifier
```

y_pred4=bc.predict(x_test)

```
array([1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1,
       0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0,
       1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0,
       1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1,
       1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1,
       1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1,
       0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0,
       1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,
       1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0])
```

from sklearn.metrics import accuracy_score

ac4=accuracy_score(y_test,y_pred4)*100

```
81.77083333333334
```

Conclusion:

the development of an earthquake prediction model is a multifaceted undertaking that demands a systematic approach. It's important to acknowledge the inherent unpredictability of short-term earthquakes, but such models still hold immense potential for offering valuable insights and enhancing earthquake risk assessment and early warning systems. Starting with data collection and preprocessing, the journey involves feature engineering to extract relevant information from seismic data. Model selection plays a pivotal role, with traditional machine learning algorithms and advanced deep learning architectures offering different avenues for exploration. Rigorous training and evaluation, accompanied by hyperparameter tuning, are crucial for refining the model's performance. Moreover, ensemble methods can further bolster accuracy by combining multiple models. Ultimately, the deployment of a well-constructed earthquake prediction model can contribute to proactive earthquake risk mitigation and preparedness efforts, even if the certainty of predicting individual earthquakes remains elusive.